Politecnico di Torino

Politecnico di Torino

Corso di Laurea in Ingegneria Informatica
A.a. 2024/2025
Sessione di Laurea Marzo/Aprile 2025

Tassonomia degli Attacchi di Rete e Mitigazione utilizzando React-VEREFOO

Relatori:
SISTO RICCARDO
VALENZA FULVIO
BRINGHENTI DANIELE

Candidato:
MAFFEO FEDERICO

Abstract

La crescente minaccia rappresentata dagli attacchi informatici è una delle sfide più critiche per la sicurezza delle reti e dei sistemi informativi. Ogni organizzazione, a prescindere dalla sua dimensione, è esposta a minacce di livello sempre più sofisticato, in grado di compromettere dati sensibili, distruggere i servizi essenziali e infliggere danni economici e alla reputazione gravi, rendendo quindi necessario uno strumento per l'analisi degli attacchi informatici. Il modello proposto in questa tesi si basa su quattro dimensioni principali: la tecnica usata, i prerequisiti, le fasi coinvolte e le contromisure. Inoltre, lo studio condotto in questa tesi valuta l'efficacia di React-VEREFOO nell'applicare misure preventive e di mitigazione degli attacchi di rete, sfruttando la sua capacità di riallocare e riconfigurare automaticamente i firewall all'interno dele rete stessa, e proponendone delle ipotetiche estensioni, al fine di mitigare alcuni attacchi di rete. Infine, la tesi si propone di entrare a far parte del numeroso contributo scientifico e professionale sulla sicurezza informatica, con applicabilità di risultati nella ricerca e nello sviluppo di sistemi di difesa più efficienti.

Contents

Chapter 1

Introduzione

L'avvento dell'era digitale ha trasformato radicalmente la società contemporanea, infiltrando ogni aspetto della vita quotidiana, dalle interazioni sociali alle transazioni commerciali, dalla gestione delle informazioni alla comunicazione globale. Tale progressivo processo di digitalizzazione, pur comportando innumerevoli vantaggi a livello di efficienza, connettività e sviluppo degli orizzonti di accessibilità alla conoscenza, ha allo stesso tempo ampliato la superficie di attacco esposta ai rischi informatici. In particolare, la crescente dipendenza da parte delle organizzazioni e degli individui dalle reti informatiche ha reso queste parti in causa bersagli sempre più facili di un'ampia gamma di attacchi compromissori, i cui danni potrebbero rivelarsi estremamente nefasti. Negli ultimi tempi, nel contesto del progressivo sviluppo delle potenzialità delle reti informatiche e della digitalizzazione elettronica di tutti i processi, si è verificata un'importante escalation degli attacchi informatici per quanto riguarda la loro frequenza e complessità. Le minacce nell'era dell'informazione si evolvono costantemente, sfruttando le più recenti vulnerabilità e utilizzando tecniche di infiltrazione sempre più articolate. Di conseguenza, diventa fondamentale indagare sui principali attacchi di rete, sulle **vulnerabilità sfruttate** e sulle misure di prevenzione contro di esse. Come riportato nell'ultimo rapporto CLUSIT 2021 [1] $Figura1$, il numero di attacchi avente un impatto ritenuto "Critico" e "Alto" è notevolmente cresciuto nel corso dell'ultimo anno. Tale interpretazione risulta ulteriormente confermata dalle note di Check Point [2], le quali sostengono che il crescente dipendere da parte delle aziende delle proprie infrastrutture possa aumentarne il grado di esposizione agli attacchi informatici.

Chapter 2

Concetti fondamentali per la sicurezza delle reti

Prima di definire specificamente un attacco di rete, è utile contestualizzarlo all'interno del concetto più ampio di attacco informatico.

2.1 Definizione di Attacco Informatico

Un attacco informatico è qualsiasi azione ostile e intenzionale che ha lo scopo di compromettere la confidenzialità, l'integrità e la disponibilità di un sistema informatico, nonché dei suoi componenti, delle informazioni elaborati, memorizzati o trasmessi tramite esso o dei servizi offerti. Gli attacchi possono essere rappresentati da molti tipi diversi di attività, tra cui accessi non autorizzati, furto o distruzione delle informazioni, interruzione dei servizi, alterazioni del software e hardware [3]. Gli attacchi possono essere eseguiti da singoli individui, criminali organizzati, nazioni-stato e altri gruppi con varie intenzioni e motivazioni, come vandalismo, furto di informazioni commerciali, spionaggio industriale, sabotaggio dell'infrastruttura critica, ecc.. [4]

2.2 Definizione di Attacco di Rete

Un **attacco di rete**, in particolare, è un sottoinsieme degli attacchi informatici che si concentra specificamente sulla compromissione di una **rete informatica**, dei suoi componenti (hardware e software di rete) o dei dati che vi transitano. Si caratterizza per lo sfruttamento di vulnerabilità nei protocolli di rete, nei sistemi di comunicazione o nelle configurazioni di sicurezza per raggiungere i suoi obiettivi, che consistono nel compromettere la **confidenzialità**, l'**integrità** o la **disponibilità** (C.I.A.) della rete e delle

sue risorse [3, 5]. Questa definizione si basa sui principi fondamentali della sicurezza informatica [3].

2.2.1 Caratteristiche Distintive di un Attacco di Rete

Un attacco di rete si distingue da altri tipi di attacco informatico per le seguenti caratteristiche principali:

- **Focus sull'infrastruttura di rete:** L'obiettivo primario è la compromissione della rete e dei suoi componenti, come router, switch, firewall e server [6].
- **Sfruttamento di protocolli di rete:** Gli attacchi di rete spesso sfruttano vulnerabilità o debolezze nei protocolli di comunicazione, come TCP/IP, ARP, DNS e BGP [5].
- **Manipolazione del traffico di rete:** Molti attacchi di rete comportano la manipolazione del traffico di rete, come l'intercettazione, la modifica, il reindirizzamento o il blocco del flusso di dati [3].
- **Impatto su più sistemi:** Gli attacchi di rete si distinguono per la loro capacità di propagarsi attraverso l'infrastruttura, sfruttando le dipendenze tra i sistemi. Un singolo punto debole può essere sfruttato per compromettere più sistemi, causando un impatto a cascata sull'intera rete.

Le conseguenze di un attacco di rete possono essere molteplici e di vasta portata, tra cui:

- **Perdita di dati sensibili:** Furto di informazioni personali, dati finanziari, segreti industriali, con gravi danni per la privacy e la competitività.
- **Interruzione dei servizi:** Blocco di sistemi informatici critici, con impatti negativi sulla produttività, sulle operazioni aziendali e sui servizi pubblici.
- **Danni economici:** Perdite finanziarie dirette a causa di furti, frodi, interruzioni di attività, costi di ripristino dei sistemi e sanzioni legali.
- **Danni alla reputazione:** Perdita di fiducia da parte di clienti, partner e stakeholders, con conseguenze negative sull'immagine e sulla credibilità dell'organizzazione.

- **Implicazioni per la sicurezza nazionale:** Attacchi a infrastrutture critiche, come reti energetiche, sistemi di trasporto o istituzioni governative, possono compromettere la sicurezza e la stabilità di un paese.

Di fronte a questa crescente minaccia, la necessità di una solida comprensione delle dinamiche degli attacchi di rete è diventata imperativa. Questa tesi si propone di affrontare questa sfida attraverso un'analisi sistematica e strutturata delle diverse categorie di attacchi di rete.

Ai fini di questa tesi, considereremo come **attacchi di rete** le azioni che soddisfano almeno uno dei seguenti criteri, in quanto direttamente correlate alla compromissione della rete e dei suoi protocolli:

- **Coinvolgimento diretto di protocolli di rete:** L'attacco sfrutta direttamente vulnerabilità o debolezze in questi protocolli, compromettendone il corretto funzionamento e potenzialmente la disponibilità della rete. Esempi includono SYN flood (TCP) [5], DNS spoofing (DNS) [7] e ARP poisoning (ARP) [6].
- **Manipolazione del traffico di rete:** L'attacco intercetta, modifica, reindirizza, blocca o inonda il traffico di rete, influenzando la confidenzialità, l'integrità o la disponibilità dei dati e dei servizi. Esempi includono attacchi Man-in-the-Middle (MitM) [3], BGP hijacking [8] e attacchi DDoS [9].
- **Bersaglio primario l'infrastruttura di rete:** L'attacco mira direttamente a compromettere dispositivi di rete come router, switch, firewall, server DNS, influenzando direttamente la disponibilità e la stabilità della rete stessa.

Questa definizione ci permette di delimitare l'ambito di studio della tesi, concentrandoci sulle minacce che agiscono direttamente a livello di rete e che possono essere efficacemente classificate secondo la tassonomia presentata successivamente.

2.3 Principi Fondamentali della Sicurezza Informatica

La sicurezza informatica mira a proteggere le informazioni e i sistemi informatici da accessi non autorizzati, alterazioni e interruzioni del servizio [3].

Questo obiettivo si concretizza attraverso la garanzia di tre proprietà fondamentali, note come la Triade C.I.A.:

- **Confidenzialità:** Proteggere le informazioni da accessi non autorizzati, garantendo che solo le persone autorizzate possano accedere ai dati [3]. Esempi di meccanismi per garantire la confidenzialità includono la crittografia [10], il controllo degli accessi e l'autenticazione [3].

- **Integrità:** Assicurare che i dati non vengano modificati o alterati in modo non autorizzato, garantendone l'accuratezza e l'affidabilità [3]. Esempi di meccanismi per garantire l'integrità includono gli hash crittografici [10], le firme digitali [3] e i controlli di ridondanza.

- **Disponibilità:** Garantire che i sistemi e i dati siano accessibili agli utenti autorizzati quando necessario, prevenendo interruzioni di servizio [3]. Esempi di meccanismi per garantire la disponibilità includono i sistemi di backup e ripristino, i sistemi di ridondanza e i sistemi di rilevamento e prevenzione delle intrusioni [3].

2.4 Vulnerabilità, Minacce e Rischi

Oltre alla Triade C.I.A., è fondamentale comprendere i concetti di **vulnerabilità**, **minaccia** e **rischio** nel contesto della sicurezza informatica [11].

- **Vulnerabilità:** Una **vulnerabilità** è una debolezza o un difetto in un sistema informatico, in un software, in un protocollo o in una configurazione di sicurezza che può essere sfruttata da una minaccia per compromettere la sicurezza del sistema [11]. Le vulnerabilità possono essere di diversa natura, come bug nel codice software, errori di configurazione, debolezze nei protocolli di comunicazione o mancanza di aggiornamenti di sicurezza.

- **Minaccia:** Una **minaccia** è qualsiasi evento o circostanza che ha il potenziale di causare danni a un sistema informatico o alle informazioni che contiene [11]. Le minacce possono essere di origine naturale (es. terremoti, incendi) o umana (es. attacchi informatici, errori umani). Le minacce sfruttano le vulnerabilità per concretizzarsi in un attacco.

- **Rischio:** Il **rischio** è la probabilità che una minaccia sfrutti una vulnerabilità e causi un impatto negativo sul sistema. Il rischio è una funzione sia della probabilità che la minaccia si verifichi sia dell'impatto potenziale che essa può avere [11]. La gestione del rischio consiste nell'identificare, valutare e mitigare i rischi per la sicurezza informatica [12].

La relazione tra questi concetti può essere riassunta come segue: una **vulnerabilità** (debolezza) può essere sfruttata da una **minaccia** (pericolo) per concretizzare un **rischio** (danno potenziale). La sicurezza informatica si occupa di ridurre i rischi attraverso la mitigazione delle vulnerabilità e la prevenzione delle minacce.

Esempio: Un bug in un software di un router (vulnerabilità) può essere sfruttato da un attaccante (minaccia) per eseguire codice dannoso sul router e interrompere il servizio di rete (rischio).

2.5 Modello OSI e suite di protocolli TCP/IP

Il modello OSI (Open Systems Interconnection) fornisce un framework concettuale per descrivere le funzioni di una rete, mentre la suite di protocolli TCP/IP definisce l'architettura di Internet e le modalità di comunicazione tra i dispositivi [5, 6]. Il modello OSI, sviluppato dall'ISO (International Organization for Standardization), è un modello a sette livelli che descrive in dettaglio le diverse funzioni di rete, dalla trasmissione fisica dei dati all'interazione con le applicazioni.

I sette livelli del modello OSI sono:

1. **Livello 1 (Physical - Fisico):** Si occupa della trasmissione fisica dei dati attraverso il mezzo trasmissivo (cavi, onde radio, ecc.). Definisce le caratteristiche elettriche, meccaniche e funzionali del mezzo trasmissivo e dei segnali elettrici.

2. **Livello 2 (Data Link - Collegamento Dati):** Si occupa della trasmissione di dati tra nodi adiacenti nella stessa rete fisica. Definisce il formato dei dati (frame) e gestisce l'accesso al mezzo trasmissivo. Include protocolli come Ethernet e ARP.

3. **Livello 3 (Network - Rete):** Si occupa dell'instradamento dei pacchetti (datagrammi) tra reti diverse. Definisce l'indirizzamento logico (indirizzi IP) e il routing. Include protocolli come IP e ICMP.

4. **Livello 4 (Transport - Trasporto):** Fornisce servizi di trasporto end-to-end tra applicazioni. Gestisce la segmentazione dei dati, il controllo di flusso e il controllo di errore. Include protocolli come TCP e UDP.

5. **Livello 5 (Session - Sessione):** Gestisce le sessioni di comunicazione tra applicazioni, stabilendo, mantenendo e terminando le connessioni.

6. **Livello 6 (Presentation - Presentazione):** Si occupa della formattazione e della codifica dei dati, garantendo che i dati siano comprensibili per le applicazioni che comunicano.

7. **Livello 7 (Application - Applicazione):** Fornisce servizi specifici per le applicazioni, come la posta elettronica, il trasferimento di file e la navigazione web. Include protocolli come HTTP, DNS e SMTP.

La suite di protocolli TCP/IP, sviluppata per la rete Internet, è un'implementazione concreta di un'architettura di rete. TCP/IP prevede una struttura a quattro livelli, che riprende alcuni concetti del modello OSI, ma li semplifica e li adatta alle esigenze di Internet:

1. **Livello 1 (Link - Collegamento):** Combina le funzionalità dei livelli fisico e data link del modello OSI, occupandosi della trasmissione fisica dei dati e dell'accesso al mezzo trasmissivo.

2. **Livello 2 (Internet - Rete):** Corrisponde al livello network del modello OSI, occupandosi dell'instradamento dei pacchetti tra reti diverse.

3. **Livello 3 (Transport - Trasporto):** Corrisponde al livello transport del modello OSI, fornendo servizi di trasporto end-to-end tra applicazioni.

4. **Livello 4 (Application - Applicazione):** Combina le funzionalità dei livelli application, presentation e sessione del modello OSI, fornendo servizi specifici per le applicazioni.

Per questa tesi, faremo riferimento principalmente alla terminologia del modello OSI, poiché la stessa fornisce una granularità maggiore nel descrivere le funzionalità della rete, il che è essenziale nel contesto di una tesi che cerca di studiare le tipologie di attacco informatico. Ci riferiremo all'equivalente nella suite protocollare dei protocolli TCP/IP quando necessario per contestualizzare il nostro esame della rete al mondo reale di Internet.

Per una descrizione più dettagliata dei modelli OSI e TCP/IP e delle loro corrispondenze, si rimanda a [5, 6].

2.6 Protocolli di Rete Fondamentali

Di seguito una breve descrizione dei protocolli di rete più rilevanti per questa tesi, evidenziandone gli aspetti di sicurezza e le relative vulnerabilità:

- **IP (Internet Protocol):** Protocollo di livello 3 che si occupa dell'indirizzamento e dell'instradamento dei pacchetti. *Non fornisce meccanismi di autenticazione o integrità, rendendolo vulnerabile ad attacchi di spoofing* [3].
- **TCP (Transmission Control Protocol):** Protocollo di livello 4 orientato alla connessione che fornisce un trasporto affidabile e ordinato dei dati. *Soggetto ad attacchi di SYN flooding che ne sfruttano la procedura di handshake a tre vie* [5].
- **UDP (User Datagram Protocol):** Protocollo di livello 4 non orientato alla connessione che fornisce un trasporto veloce ma non affidabile. *Soggetto ad attacchi di UDP flooding a causa della sua natura senza connessione* [5].
- **ARP (Address Resolution Protocol):** Protocollo di livello 2 utilizzato per risolvere gli indirizzi IP in indirizzi MAC. *Vulnerabile ad attacchi di ARP spoofing a causa della mancanza di autenticazione* [6].
- **DNS (Domain Name System):** Protocollo di livello 7 utilizzato per la risoluzione dei nomi di dominio in indirizzi IP. *Soggetto ad attacchi di DNS spoofing e DNS hijacking* [7].
- **ICMP (Internet Control Message Protocol):** Protocollo di livello 3 utilizzato per la diagnostica e il controllo della rete. *Soggetto ad attacchi di ICMP flooding (Ping Flood)* [9].
- **BGP (Border Gateway Protocol):** Protocollo di routing dinamico utilizzato tra sistemi autonomi su Internet. *Soggetto ad attacchi di BGP hijacking che possono reindirizzare il traffico su Internet* [8].

Chapter 3

Obiettivi della Tesi

Con riferimento alle recenti ricerche e ai rapporti di settore che mostrano un aumento della frequentazione e del livello di sofisticazione degli attacchi informatici da parte di aggressori di ogni tipologia [1,2], c'è, effettivamente, la necessità di analizzare le minacce informatiche esistenti in modo più strutturato. L'andamento attuale della sicurezza informatica rileva come le tecniche di attacco cambino rapidamente, prendendo di mira nuove vulnerabilità e diventando sempre più sofisticate. Pertanto, per difendersi efficacemente da minacce sempre più complesse, è fondamentale dotarsi di strumenti in grado di classificare e analizzare i vari tipi di attacchi di rete in modo avanzato. Attraverso questa tesi si intende sviluppare un modello avanzato per la classificazione degli attacchi di rete, superando le limitazioni delle tassonomie esistenti e combinando i punti di forza di framework consolidati come CAPEC, ATT&CK e Kill Chain [13,14]. Tale modello non solo classificherà gli attacchi in base alle tecniche usate, ma includerà anche i prerequisiti per l'attaccante, le fasi lavorative e le contromisure. Questo permetterà una visione più dettagliata e sistematica della minaccia, risultando utile non solo per la ricerca accademica, ma anche per il settore operativo della cybersecurity. Inoltre, questo modello strutturato non solo gioverà alla comprensione delle minacce informatiche, ma renderà anche più facili le strategie di difesa proattiva. Infine, si valuterà l'utilità di React-VEREFOO [91] in vari scenari di attacco, sfruttando la sua capacità di riallocazione e riconfigurzione automatica dei firewall all'interno della rete, proponendo eventualmente delle estensioni che renderebbero possibile la mitigazione di alcuni attacchi, che non sono attualmente mitigabili nella sua versione attuale.

Chapter 4

Un Modello Integrato per l'Analisi degli Attacchi Informatici

In questo capitolo viene presentato un modello integrato per l'analisi degli attacchi informatici, concepito per superare le limitazioni dei sistemi di classificazione tradizionali. Il modello proposto è finalizzato a fornire uno strumento pratico e dettagliato che consenta di rappresentare gli attacchi in maniera completa, mettendo in luce non solo la tecnica utilizzata per raggrupparli, ma anche i prerequisiti che l'attaccante deve soddisfare, le fasi operative che caratterizzano il flusso dell'attacco e le contromisure applicabili per rilevarli e mitigarli.

4.1 Modelli di Analisi degli Attacchi Esistenti

Nel panorama attuale sono stati sviluppati diversi modelli per l'analisi degli attacchi informatici, ognuno dei quali offre importanti contributi al panorama della sicurezza contro gli attacchi informatici.

Il modello CAPEC (Common Attack Pattern Enumeration and Classification) [13], sviluppato da MITRE, fornisce un catalogo dettagliato e gerarchico di metodi di attacco. Esso permette di identificare e classificare le tecniche d'attacco in modo strutturato, evidenziando le relazioni tra pattern, le dipendenze e le variazioni tra metodi. Grazie alla sua organizzazione gerarchica, CAPEC consente agli analisti di passare da una visione generale a livelli di dettaglio specifici, supportando così la progettazione di contromisure mirate e l'analisi comparativa dei rischi.

Il framework ATT&CK (Adversarial Tactics, Techniques, and Common

Knowledge) [14], anch'esso sviluppato da MITRE, si basa sull'osservazione di tattiche, tecniche e procedure (TTP) adottate dagli attaccanti. Esso offre una matrice che organizza in maniera dettagliata il "cosa" e il "come" degli attacchi, basandosi su dati reali raccolti da incidenti. Tuttavia, non include in modo esplicito i prerequisiti operativi necessari all'esecuzione di una tecnica, e la sua struttura, pur essendo molto dettagliata sul livello operativo, non fornisce una visione sequenziale o contestuale dell'intero flusso dell'attacco [15].

Il modello Cyber Kill Chain [16], sviluppato da Lockheed Martin, suddivide l'attacco in fasi sequenziali. Le fasi sono le stesse per ogni attacco: Reconnaissance, Weaponization, Delivery, Exploitation, Installation, Command and Control, and Actions on Objectives. L'idea chiave è che ogni attacco può essere rappresentato come una catena di attività, e il suo successo dipende dal rispetto della sequenza temporale. Interrompere questa catena, bloccando anche solo una fase, impedisce all'attaccante di raggiungere il suo obiettivo finale. Questo approccio lineare facilita la comprensione del flusso dell'attacco e l'identificazione dei punti critici di intervento [15]. Nonostante ciò, la rigidità del modello ed il suo orientamento verso una sequenza predefinita non consentono di catturare efficacemente attacchi complessi, che possono includere fasi iterative, parallelismi o flussi non lineari, limitando così la sua applicabilità a scenari operativi moderni.

4.2 Il Modello Proposto: Un Approccio Integrato e Dettagliato

Il modello integrato che proponiamo nasce dall'esigenza di superare le limitazioni dei framework esistenti, combinando i loro punti di forza in una struttura operativa e flessibile. Tale approccio classifica gli attacchi raggruppandoli per tecnica e li caratterizza ulteriormente attraverso altre dimensioni essenziali, quali i prerequisiti, le fasi operative e le contromisure.

Un aspetto distintivo di questo modello riguarda l'esplicitazione dei prerequisiti: vengono chiaramente indicate le condizioni e le risorse che l'attaccante deve possedere per eseguire l'attacco. Ad esempio, si considerano fattori come l'accesso fisico o remoto alla rete, la presenza di vulnerabilità specifiche o l'assenza di adeguati meccanismi di autenticazione. Tale dimensione, che risulta assente in maniera esplicita in framework come ATT&CK e nella Cyber Kill Chain, è fondamentale per una corretta valutazione del rischio e per orientare tempestivamente le contromisure.

Il modello suddivide inoltre l'attacco in fasi operative, consentendo di de-

scrivere il flusso in maniera flessibile. Pur ispirandosi al concetto di fasi della Cyber Kill Chain, il nostro approccio non si limita a una sequenza lineare predefinita, ma grazie alla flessibilità di rappresentazione di tali fasi si adatta a scenari complessi in cui esse possono essere iterative o non sequenziali. In questo modo, le "fasi operative" sono definite come una sequenza funzionale di eventi rilevanti per l'esecuzione e la difesa dell'attacco. Per ogni tecnica e attacco il modello integra una componente relativa alle contromisure, fornendo indicazioni pratiche per il rilevamento e la mitigazione. Questo approccio non solo descrive le tecniche d'attacco, ma supporta attivamente la progettazione di strategie difensive efficaci.

Sebbene CAPEC [13] offra un catalogo estremamente dettagliato e strutturato delle tecniche d'attacco, esso risulta talvolta troppo complesso per essere applicato in contesti operativi dove la rapidità di risposta è fondamentale. In tali situazioni, il modello proposto presenta alcuni vantaggi. Ad eesempio, la struttura sintetica evidenzia le fasi essenziali dell'attacco e le contromisure immediate, facilitando una risposta tempestiva in situazioni di incident response. Inoltre, la chiarezza e la modularità del modello lo rendono particolarmente utile per la formazione del personale tecnico, consentendo una comprensione rapida dei meccanismi d'attacco e delle strategie difensive.

I principali vantaggi di questo modello sono riassunti di seguito:

- **Completezza Operativa:** Integrazione delle dimensioni della tecnica, dei prerequisiti, delle fasi operative e delle contromisure per una visione completa e immediatamente applicabile.
- **Flessibilità e Adattabilità:** Capacità di rappresentare dinamicamente attacchi in scenari non lineari, in alternativa alla rigidità dei modelli tradizionali.
- **Orientamento alla Difesa:** Esplicitazione dei prerequisiti e delle contromisure per un intervento tempestivo e mirato.
- **Utilità Formativa:** Struttura modulare e chiara che facilita la formazione e l'aggiornamento del personale tecnico.

Chapter 5

Classificazione degli Attacchi di Rete secondo la Tassonomia Proposta

Questo capitolo applica la tassonomia definita nel Capitolo 4 per classificare specifici attacchi di rete. Questa applicazione pratica dimostra l'utilità e la granularità della tassonomia proposta, fornendo una struttura chiara per l'analisi e la comprensione delle diverse tipologie di attacco. Gli attacchi sono raggruppati a seconda della tecnica utilizzata per realizzarli.

5.1 Spoofing

Lo spoofing consiste nella falsificazione dell'identità e degli indirizzi, per ingannare i sistemi di rete, inducendoli a credere che un pacchetto provenga da una fonte diversa da quella reale. L'obiettivo è mascherare la vera identità dell'attaccante o impersonare un'entità legittima per ottenere un vantaggio illecito, come l'accesso non autorizzato a risorse o la compromissione di sistemi [3].

- **ARP Spoofing:** L'attaccante invia pacchetti ARP (Address Resolution Protocol) falsificati sulla rete locale, associando il proprio indirizzo MAC all'indirizzo IP di un altro host (ad esempio, il gateway predefinito). Questo devia il traffico destinato a quest'ultimo verso l'attaccante, che può quindi intercettarlo, modificarlo o inoltrarlo alla vittima. L'ARP spoofing sfrutta la natura senza autenticazione del protocollo ARP [17].

  Prerequisiti: [18]

- L'attaccante deve avere accesso alla rete.
- L'attaccante deve conoscere o essere in grado di scoprire gli indirizzi IP e MAC dei dispositivi che desidera attaccare. Questo può avvenire tramite scansione della rete o intercettazione del traffico.

Fasi dell'attacco [18]:

1. L'attaccante scansiona la rete per determinare gli indirizzi IP dei dispositivi target.
2. L'attaccante invia risposte ARP falsificate.
3. Le risposte ARP falsificate associano l'indirizzo MAC dell'attaccante agli indirizzi IP dei dispositivi target.
4. I dispositivi target aggiornano le loro tabelle ARP con le informazioni falsificate.
5. Il traffico tra i dispositivi target viene deviato verso l'attaccante.
6. L'attaccante può intercettare, modificare o inoltrare il traffico.

Contromisure:

- *Monitoraggio passivo:* Monitoraggio del traffico ARP e ricerca di incongruenze nelle mappature Ethernet-IP. Questo metodo presenta un ritardo nell'apprendimento e nel rilevamento dello spoofing [19].
- *Monitoraggio attivo:* Iniezione di pacchetti di richiesta ARP e TCP SYN nella rete per verificare la presenza di incongruenze. Questo approccio è più veloce, intelligente, scalabile e affidabile rispetto ai metodi passivi [19].
- *Analisi delle tabelle ARP:* La presenza di due IP con lo stesso MAC address nella tabella ARP di un dispositivo è un chiaro segnale di un attacco ARP spoofing in corso. [18] È possibile utilizzare il comando 'arp -a' (Windows/Linux) per visualizzare la tabella ARP. [18]
- *Packet Filtering:* L'utilizzo di filtri di pacchetti può proteggere la rete da pacchetti trasmessi in modo dannoso sulla rete, nonché da indirizzi IP sospetti. [20]
- *Static ARP:* Definire manualmente delle associazioni IP-MAC statiche per i dispositivi critici (es. gateway) può prevenire che vengano modificate da un attacco. [18]

- **IP Spoofing:** L'attaccante falsifica l'indirizzo IP sorgente dei pacchetti IP, rendendo difficile tracciare l'origine dell'attacco o consentendo di bypassare filtri basati sull'indirizzo IP. Questo è spesso utilizzato negli attacchi DDoS per amplificare il traffico, in quanto rende difficile identificare le singole sorgenti dell'attacco [3].

  Prerequisiti [21]:

  - L'attaccante deve essere in grado di inviare pacchetti IP con indirizzi sorgente falsificati. Questo può essere fatto da qualsiasi punto della rete, anche al di fuori della rete bersaglio.
  - L'attaccante deve conoscere l'indirizzo IP della vittima (il sistema che si vuole attaccare) e l'indirizzo IP che vuole falsificare (il sistema che vuole impersonare).

  Fasi dell'attacco [21]:

  1. L'attaccante costruisce pacchetti IP con indirizzi sorgente falsificati.
  2. L'attaccante invia i pacchetti alla vittima (o a un sistema intermedio che li inoltra alla vittima).
  3. La vittima (o il sistema intermedio) riceve i pacchetti e li elabora, credendo che provengano dalla fonte indicata nell'indirizzo IP falsificato.
  4. A seconda dell'obiettivo, l'attaccante può continuare a inviare pacchetti falsificati per raggiungere diversi scopi (es. DDoS, bypass di filtri, ecc.).

  Contromisure:

  - *Ingress Filtering:* Blocca i pacchetti con indirizzi IP sorgente che non appartengono alla rete locale [22].
  - *Egress Filtering:* Blocca i pacchetti in uscita dalla rete con indirizzi IP sorgente che non corrispondono alla rete interna [22].
  - *Reverse Path Forwarding (RPF):* Verifica che il percorso di ritorno di un pacchetto corrisponda al percorso di andata previsto [23].
  - *Analisi del traffico e delle anomalie:* Ricerca pattern di traffico insoliti che potrebbero indicare un attacco di spoofing [24].
  - *Utilizzo di VPN su reti Wi-Fi pubbliche* [24]: Utilizzare una VPN su reti pubbliche rende più difficile l'IP spoofing perché maschera

l'indirizzo IP reale con quello del server VPN, rendendo più difficile per un attaccante falsificare l'indirizzo e dirottare il traffico.

- *Verifica che i siti Web visitati siano HTTPS* [24]: HTTPS protegge le comunicazioni web da IP spoofing tramite autenticazione, cifratura e integrità dei dati, impedendo l'accesso a informazioni sensibili e proteggendo da siti web falsi.

- **MAC Spoofing:** L'attaccante modifica l'indirizzo MAC del proprio dispositivo per impersonare un altro host sulla rete locale, ad esempio per superare filtri basati su MAC address o per ottenere accesso non autorizzato a risorse di rete. Questo è possibile poiché l'indirizzo MAC è un identificativo a livello di Data Link e può essere alterato a livello software [25].

  Prerequisiti [25]:

  - L'attaccante deve avere accesso alla rete locale (LAN).
  - L'attaccante deve disporre di un dispositivo (ad esempio, un computer) e dei privilegi necessari per modificare l'indirizzo MAC.
  - L'attaccante deve conoscere (o essere in grado di determinare) l'indirizzo MAC del dispositivo che desidera impersonare (opzionale, ma spesso utile per bersagli specifici).

  Fasi dell'attacco [25]:

  1. L'attaccante si connette alla rete locale.
  2. L'attaccante utilizza un software (o tool) per modificare l'indirizzo MAC del proprio dispositivo. Diversi metodi sono disponibili, tra cui l'utilizzo dei comandi 'ip' (iproute2) o 'macchanger'.
  3. L'attaccante imposta l'indirizzo MAC del proprio dispositivo uguale all'indirizzo MAC del dispositivo che vuole impersonare.
  4. L'attaccante può iniziare a inviare e ricevere traffico con l'indirizzo MAC falsificato.
  5. La rete (switch, firewall, etc.) identifica l'attaccante come il dispositivo legittimo che ha lo stesso MAC address.
  6. L'attaccante può superare i filtri basati su MAC address e ottenere accesso non autorizzato a risorse.

  Contromisure:

- *Configurazione di funzionalità specifiche negli switch:* Configurare funzionalità come port security, DHCP snooping e Dynamic ARP Inspection [25, 26]: Queste tre tecnologie, agendo sinergicamente, proteggono la rete da MAC spoofing impedendo l'accesso non autorizzato, bloccando l'assegnazione di IP falsi e invalidando le modifiche illecite alle tabelle ARP. In altre parole, rendono molto più difficile per un attaccante impersonare un altro dispositivo sulla rete.
- *Abilitazione dell'autenticazione 802.1x nella LAN:* Se lo switch lo supporta, è possibile abilitare l'autenticazione 802.1x nella LAN [26].
- *Monitoraggio delle tabelle MAC address degli switch:* Verificare la presenza di cambiamenti inattesi nelle associazioni MAC-porta [25].
- *Rilevamento di conflitti di MAC address:* Identificare la presenza di due o più dispositivi con lo stesso MAC address sulla rete [25].
- *Crittografia dei dati di rete:* Proteggere i dati con la crittografia rende più difficile per gli attaccanti leggerli o modificarli, anche se riescono a intercettarli [25].
- *Segmentazione della rete:* Dividere la rete in sottoreti più piccole per limitare la diffusione degli attacchi [25].
- *Intrusion Detection Systems (IDS):* Implementare sistemi IDS per identificare e segnalare attività di spoofing [25].

5.2 Reindirizzamento del Traffico (Hijacking)

Gli attacchi di "Hijacking" (dirottamento) mirano a deviare il traffico di rete verso destinazioni non autorizzate, prendendo il controllo del flusso di comunicazione.

- **DNS Hijacking:** L'attaccante compromette un server DNS autoritativo o intercetta le comunicazioni tra un client e un server DNS per reindirizzare le query DNS verso un server controllato dall'attaccante, che ottiene così il controllo sulla risoluzione dei nomi di dominio. Il DNS Hijacking può avvenire in diversi punti della catena di query DNS [27].

  Prerequisiti [27]:

  - L'attaccante può infettare il computer o il router della vittima con malware che altera le impostazioni DNS .

- L'attaccante può compromettere un server DNS, il che richiede un livello di competenza tecnica maggiore.
- In alcuni casi, il provider di servizi internet (ISP) può effettuare DNS hijacking per mostrare pubblicità o raccogliere statistiche.

Fasi dell'attacco:

1. L'attaccante identifica una vulnerabilità XSS nell'applicazione web.
2. L'attaccante crea un codice JavaScript malevolo (payload).
3. L'attaccante inietta il payload nella pagina web attraverso la vulnerabilità. Questo può avvenire tramite:
   - *Reflected XSS:* L'attaccante invia un URL appositamente costruito alla vittima (es. tramite email o social media).
   - *Stored XSS:* L'attaccante inserisce il payload in un campo di input (es. commento, forum) che viene poi visualizzato da altri utenti.
   - *DOM-based XSS:* L'attaccante manipola il DOM della pagina web tramite JavaScript.
4. Il browser della vittima esegue il codice malevolo, che può rubare cookie, reindirizzare l'utente, modificare la pagina web o compiere altre azioni dannose.

Contromisure:

- *Utilizzo di software antivirus affidabile e aggiornato:* Proteggere il dispositivo da malware che modifica le impostazioni DNS è fondamentale [27]. Si consiglia anche l'utilizzo di strumenti di protezione dalle minacce.
- *Evitare link sospetti:* Non cliccare su link provenienti da fonti sconosciute o non attendibili [27]. Controllare sempre l'URL prima di cliccare.
- *Utilizzo di una VPN con DNS privato:* Una VPN cripta il traffico e protegge le query DNS da terze parti, impedendo l'intercettazione e la manipolazione [27].
- *Utilizzo di server DNS sicuri:* Configurare il dispositivo per utilizzare server DNS pubblici e sicuri, che offrono protezione da attacchi di DNS hijacking e altre minacce [27].

- *Implementazione di DNSSEC:* Implementare DNSSEC sui server DNS autoritativi per firmare digitalmente le zone DNS e sui resolver per validare le firme, garantendo l'integrità e l'autenticità delle risposte DNS [28].

- **Session Hijacking:** L'attaccante intercetta l'ID di sessione di un utente autenticato, ottenendo accesso non autorizzato al suo account. Questo può avvenire tramite sniffing del traffico di rete, cross-site scripting (XSS), malware, o altre tecniche, consentendo all'attaccante di impersonare l'utente e compiere azioni a suo nome. Questo attacco è particolarmente pericoloso per le applicazioni web che utilizzano cookie per la gestione delle sessioni [29].

  Prerequisiti [29]:

  - L'attaccante deve essere in grado di intercettare il traffico di rete tra l'utente e il server web, ad esempio tramite sniffing su reti Wi-Fi pubbliche o compromettendo il router dell'utente.
  - L'attaccante può sfruttare vulnerabilità nel sito web o nell'applicazione per eseguire cross-site scripting (XSS) e rubare i cookie di sessione.
  - L'attaccante può utilizzare malware per rubare i cookie di sessione o intercettare le comunicazioni.

  Fasi dell'attacco [29]:

  1. L'utente si autentica su un sito web, ottenendo un ID di sessione (memorizzato in un cookie).
  2. L'attaccante intercetta l'ID di sessione dell'utente. Questo può avvenire tramite:
     - *Session side jacking:* Sniffing del traffico di rete, spesso su Wi-Fi pubblici o siti web senza crittografia SSL .
     - *Session fixation:* L'attaccante induce l'utente a utilizzare un ID di sessione creato dall'attaccante stesso, ad esempio tramite link di phishing.
     - *Brute-force:* Tentativi di indovinare l'ID di sessione, anche se questo metodo è poco efficace.
     - *Cross-site scripting (XSS):* Sfruttamento di vulnerabilità nei siti web per inserire codice dannoso che ruba i cookie degli utenti.
     - *Malware:* Utilizzo di software malevolo per rubare i cookie di sessione.

- *IP spoofing:* L'attaccante modifica il proprio indirizzo IP per sembrare l'utente durante il three-way TCP handshake, permettendo l'intercettazione della sessione.

3. L'attaccante utilizza l'ID di sessione rubato per impersonare l'utente e accedere al suo account.
4. L'attaccante può compiere azioni a nome dell'utente, come effettuare transazioni, modificare informazioni personali o rubare dati.

Contromisure:

- *HTTPS (HTTP Secure):* Utilizzare HTTPS per crittografare le comunicazioni tra client e server, proteggendo gli ID di sessione dall'intercettazione durante la trasmissione sulla rete [**?**,29].
- *Utilizzo di cookie sicuri (HttpOnly, Secure):* Configurare i cookie di sessione con gli attributi HttpOnly e Secure per prevenire l'accesso tramite script client (XSS) e la trasmissione su connessioni non sicure (HTTP), riducendo il rischio di furto dei cookie [30].
- *Rigenerazione periodica degli ID di sessione:* Rigenerare periodicamente gli ID di sessione, ad esempio dopo il login o dopo un certo periodo di inattività, per limitare la finestra di opportunità per un attacco nel caso in cui un ID di sessione venga compromesso [31].
- *Timeouts di sessione:* Impostare timeouts di sessione per terminare automaticamente le sessioni inattive, riducendo il periodo di tempo in cui un ID di sessione rubato può essere utilizzato [31].
- *Monitoraggio del comportamento degli utenti:* Rilevare anomalie nel comportamento degli utenti, come accessi da posizioni geografiche inusuali, orari insoliti o attività sospette che non corrispondono al profilo tipico dell'utente [32].
- *Utilizzo di tecniche di rilevamento delle intrusioni (IDS/IPS):* Configurare IDS/IPS per rilevare pattern di traffico sospetti associati a session hijacking, come l'improvvisa comparsa di traffico proveniente da un indirizzo IP diverso da quello previsto per una determinata sessione [33].
- *Non visitare siti web non sicuri:* Evitare siti web che non utilizzano HTTPS (e quindi non hanno il lucchetto nel browser) [29].
- *Utilizzare software anti-malware:* Proteggere il dispositivo da malware che potrebbe rubare cookie o intercettare sessioni [29].

Replay Attack: In un replay attack, l'attaccante intercetta una comunicazione di rete (ad esempio, una richiesta di autenticazione) e la ritrasmette in seguito per ottenere un accesso non autorizzato o per ripetere un'azione. L'attaccante può impersonare una delle parti comunicanti, ingannando così quella che riceve [34].

Prerequisiti:

- L'attaccante deve essere in grado di intercettare le comunicazioni di rete tra le due parti. Questo può avvenire tramite sniffing del traffico, compromissione di un nodo di rete o utilizzo di tecniche "man-in-the-middle" [34].

Fasi dell'attacco:

1. Una delle parti (es. l'utente) invia una comunicazione (es. una richiesta di autenticazione) all'altra parte (es. il server).
2. L'attaccante intercetta la comunicazione.
3. L'attaccante può ritardare, modificare o semplicemente ritrasmettere la comunicazione originale in un momento successivo.
4. Il sistema ricevente, non essendo in grado di distinguere la comunicazione ritrasmessa dalla comunicazione originale, può eseguire l'azione richiesta (es. concedere l'accesso).

Contromisure [34]:

- *Timestamp:* Aggiungere timestamp alle comunicazioni permette di verificare la loro validità temporale e di scartare messaggi troppo vecchi.
- *Nonce:* Utilizzare nonce (numeri casuali univoci) in ogni comunicazione rende impossibile la ritrasmissione dello stesso messaggio, poiché il nonce cambierà ad ogni invio.
- *Sequenze numeriche:* Includere numeri di sequenza nei messaggi permette di verificare che non vi siano messaggi mancanti o duplicati.
- *Autenticazione a due fattori:* Richiedere un secondo fattore di autenticazione (es. codice OTP) rende più difficile per l'attaccante sfruttare una comunicazione intercettata.
- *Crittografia SSL/TLS:* Utilizzare connessioni crittografate (HTTPS) protegge le comunicazioni dall'intercettazione e dalla manipolazione.

- *Token di sessione:* Utilizzare token di sessione con scadenza e rigenerazione periodica per limitare la validità di eventuali comunicazioni intercettate.
- *Monitoraggio del traffico:* Monitorare il traffico di rete per rilevare anomalie o pattern sospetti che potrebbero indicare un attacco di replay.
- *Utilizzo di VPN:* Una VPN crea un tunnel crittografato per il traffico internet, rendendo più difficile l'intercettazione delle comunicazioni.
- *Evitare reti Wi-Fi pubbliche non sicure:* Le reti Wi-Fi pubbliche non protette sono un terreno fertile per attacchi di intercettazione e replay.
- *Non visitare siti HTTP:* Evitare siti web che utilizzano il protocollo HTTP non sicuro, preferendo sempre HTTPS.
- *Software anti-malware aggiornato:* Proteggere i dispositivi da malware che potrebbero intercettare comunicazioni o rubare dati sensibili.

5.3 Manipolazione del Contenuto Web (Injection di Script)

Gli attacchi di "Injection di Script" consistono nell'iniettare codice malevolo all'interno di pagine web o applicazioni, cercando di alterarne il comportamento, rubare informazioni o eseguire azioni non autorizzate nel contesto dell'utente. Questa categoria comprende diverse tipologie di attacco, che sfruttano vulnerabilità nelle applicazioni per iniettare codice, tipicamente lato client (come JavaScript) ma anche lato server (come SQL o comandi del sistema operativo) [35].

- **Cross-Site Scripting (XSS):** L'attaccante sfrutta vulnerabilità nelle applicazioni web per iniettare script malevoli nel contesto di un sito web attendibile. Questo codice può essere utilizzato per rubare cookie, reindirizzare gli utenti a siti web malevoli, defacciare siti web o eseguire altre azioni dannose [37]. Esistono diverse tipologie di XSS [37]:
  - **Reflected XSS:** Il codice malevolo viene iniettato tramite parametri URL o input di form e viene "riflesso" dal server nella risposta HTTP.

- **Stored XSS:** Il codice malevolo viene memorizzato sul server (ad esempio, in un database o in un forum) e viene eseguito quando un utente visualizza la pagina che contiene il codice.
- **DOM-based XSS:** Il codice malevolo viene eseguito a seguito di modifiche al Document Object Model (DOM) nel browser dell'utente.

Gli attacchi XSS possono essere utilizzati per realizzare altre tipologie di attacco, come il Session Hijacking (rubando i cookie di sessione) [37].

Prerequisiti:

- L'attaccante deve trovare una vulnerabilità nell'applicazione web che permetta l'iniezione di codice (es. input non validato, output non codificato) [37].

Fasi dell'attacco [38]:

1. L'utente inserisce un nome di dominio (ad esempio, `www.esempio.it`) nella barra degli indirizzi del browser.
2. Il dispositivo dell'utente avvia una query DNS per risolvere il dominio e ottenere l'indirizzo IP corrispondente.
3. In questa fase, il percorso di attacco può divergere in maniera non lineare, a seconda del vettore sfruttato:
   - *Malware:* Un malware presente sul dispositivo intercetta la query DNS, reindirizzandola verso un server DNS controllato dall'attaccante.
   - *Server DNS Compromesso:* Un server DNS legittimo lungo la catena di risoluzione viene compromesso e risponde con dati manipolati.
   - *Interferenza dell'ISP:* L'ISP o un nodo intermedio intercetta la query e fornisce una risposta alterata, tipicamente in scenari relativi a domini inesistenti (NXDOMAIN).
4. Il browser dell'utente viene, infine, reindirizzato verso un sito web diverso da quello originariamente richiesto, il quale può ospitare pagine di phishing, siti contenenti malware o pagine pubblicitarie ingannevoli.

Contromisure:

- *Sanitizzazione degli input:* Validare e sanitizzare tutti gli input forniti dall'utente per rimuovere o codificare caratteri speciali che potrebbero essere interpretati come codice [36, 37].

- *Output Encoding:* Codificare l'output dell'applicazione per evitare che il codice iniettato venga interpretato dal browser [39].
- *Content Security Policy (CSP):* Implementare una CSP per definire le origini da cui il browser può caricare risorse, riducendo il rischio di esecuzione di script non autorizzati [37].
- *HttpOnly cookie attribute:* Impostare l'attributo HttpOnly per i cookie per prevenire l'accesso da parte di script client [30, 37].
- *Utilizzo di framework di sicurezza:* Utilizzare framework che offrono protezione integrata contro le vulnerabilità XSS [37].
- *Aggiornamenti software:* Mantenere aggiornati software e librerie per correggere vulnerabilità note [37].
- *Test di sicurezza:* Eseguire regolarmente test di sicurezza (es. penetration testing, analisi del codice) per identificare e correggere vulnerabilità XSS [37].
- *Web Application Firewall (WAF):* I WAF possono essere configurati per rilevare e bloccare tentativi di XSS [39].

- **SQL Injection:** L'attaccante sfrutta vulnerabilità nelle applicazioni web che interagiscono con database SQL per iniettare codice SQL malevolo all'interno di input di form o parametri URL, con l'obiettivo di ottenere accesso non autorizzato ai dati, modificarli o eseguire comandi sul server [40].

  Prerequisiti:

  - L'applicazione web deve essere vulnerabile a SQL Injection, ovvero deve consentire l'inserimento di input non validato direttamente all'interno di query SQL [40].

  Fasi dell'attacco [40]:

  1. L'attaccante identifica un punto di ingresso vulnerabile (es. campo di input, parametro URL) .
  2. L'attaccante elabora un input SQL malevolo (payload) che, se eseguito, gli permetterà di raggiungere i suoi obiettivi (es. furto di dati, modifica di dati, esecuzione di comandi).
  3. L'attaccante inietta il payload nel punto di ingresso vulnerabile.
  4. L'applicazione web esegue la query SQL, inclusivo del payload malevolo.

5. Il database esegue i comandi SQL presenti nel payload, consentendo all'attaccante di raggiungere i suoi obiettivi.

Contromisure:

- *Prepared statements (o parameterized queries):* Utilizzare prepared statements per separare il codice SQL dai dati forniti dall'utente, prevenendo l'interpretazione di input malevoli come codice [40].
- *Input validation:* Validare e sanitizzare gli input dell'utente per rimuovere o codificare caratteri speciali che potrebbero essere utilizzati in attacchi SQL Injection [36, 40].
- *Code review:* Analizzare il codice sorgente per identificare query SQL costruite concatenando stringhe con input dell'utente è una pratica fondamentale per individuare potenziali vulnerabilità SQL Injection [40, 41].
- *Fuzzing e test di penetrazione:* Testare l'applicazione con input specificamente progettati per rivelare vulnerabilità SQL Injection [40].
- *Web Application Firewalls (WAF):* I WAF possono essere configurati per rilevare e bloccare tentativi di SQL Injection analizzando il traffico HTTP e intercettando pattern di attacco noti, come l'uso di parole chiave SQL o la presenza di caratteri speciali [40, 42].

- **Command Injection:** Questo attacco si verifica quando un'applicazione web esegue comandi del sistema operativo basati su input forniti dall'utente senza una corretta validazione. L'attaccante può iniettare comandi del sistema operativo all'interno di questi input, ottenendo il controllo del sistema o accedendo a file sensibili [43].

  Prerequisiti:

  - L'applicazione web deve essere vulnerabile a Command Injection, ovvero deve consentire l'esecuzione di comandi del sistema operativo con input non validato [43].

  Fasi dell'attacco [43]:

  1. L'attaccante identifica un punto di ingresso vulnerabile (es. campo di input, parametro URL) che viene utilizzato per costruire comandi del sistema operativo .

2. L'attaccante elabora un input malevolo contenente comandi del sistema operativo (payload) che, se eseguiti, gli permetteranno di raggiungere i suoi obiettivi (es. accesso non autorizzato, modifica di file, esecuzione di codice).
3. L'attaccante inietta il payload nel punto di ingresso vulnerabile.
4. L'applicazione web esegue il comando del sistema operativo, includendo il payload malevolo.
5. Il sistema operativo esegue i comandi presenti nel payload, consentendo all'attaccante di raggiungere i suoi obiettivi.

Contromisure:

- *Input validation:* Validare e sanitizzare gli input dell'utente per rimuovere o codificare caratteri speciali che potrebbero essere utilizzati per iniettare comandi [36, 43].
- *Utilizzo di funzioni sicure per l'esecuzione di comandi:* Evitare l'uso di funzioni che eseguono direttamente comandi del sistema operativo con input dell'utente. Se necessario, utilizzare funzioni che offrono un maggiore controllo sull'esecuzione dei comandi o che permettono di specificare argomenti in modo sicuro. OWASP Command Injection Prevention Cheat Sheet fornisce esempi di funzioni sicure e non sicure nei vari linguaggi di programmazione [43].
- *Principio del minimo privilegio:* Eseguire i processi dell'applicazione con i minimi privilegi necessari. Questo principio è fondamentale per la sicurezza e viene discusso in diverse risorse, tra cui quelle del NIST [44].
- *Code review:* Analizzare il codice sorgente per identificare l'uso di funzioni che eseguono comandi del sistema operativo con input non validati è una pratica fondamentale. La guida di OWASP al Code Review fornisce indicazioni su come cercare queste vulnerabilità [41, 43].
- *Fuzzing e test di penetrazione:* Testare l'applicazione con input contenenti caratteri speciali o comandi del sistema operativo è una tecnica efficace per scoprire vulnerabilità di Command Injection. OWASP offre risorse sul penetration testing che includono tecniche di fuzzing e input injection [43].

- **Code Injection:** Questo termine generico si riferisce all'iniezione di codice arbitrario all'interno di un'applicazione, con l'obiettivo di al-

terarne il comportamento o di eseguire codice malevolo. Questo può includere l'iniezione di codice in linguaggi di scripting lato server (come PHP, Python, ecc.) o l'iniezione di codice compilato (ad esempio, tramite buffer overflow o DLL injection) [45]. Data la sua generalità, le tecniche di rilevamento e mitigazione sono simili a quelle descritte per XSS, SQL Injection e Command Injection, ma con alcune specificazioni:

Prerequisiti:

- L'applicazione deve presentare una vulnerabilità che permetta l'iniezione e l'esecuzione di codice non autorizzato. La natura specifica della vulnerabilità varia a seconda del tipo di Code Injection (es. input non validato per scripting lato server, buffer overflow per codice compilato) [45].

Fasi dell'attacco [45]:

1. L'attaccante identifica una vulnerabilità sfruttabile per l'iniezione di codice .
2. L'attaccante elabora un payload contenente codice malevolo (script, codice compilato, ecc.) che, se eseguito, gli permetterà di raggiungere i suoi obiettivi.
3. L'attaccante inietta il payload nell'applicazione attraverso la vulnerabilità.
4. L'applicazione esegue il codice iniettato, consentendo all'attaccante di raggiungere i suoi obiettivi (es. controllo del sistema, accesso a dati sensibili, esecuzione di codice arbitrario).

Contromisure:

- *Input validation e sanitizzazione:* Validare e sanitizzare *tutti* gli input dell'utente, indipendentemente dal contesto in cui vengono utilizzati. Questo include la rimozione o la codifica di caratteri speciali e la verifica che gli input rispettino un formato atteso [36, 45].
- *Utilizzo di linguaggi e framework sicuri:* Utilizzare linguaggi di programmazione e framework che offrono meccanismi di protezione contro Code Injection, come la tipizzazione forte o l'escaping automatico, può ridurre significativamente il rischio di queste vulnerabilità. OWASP sottolinea l'importanza di considerare la sicurezza durante la fase di progettazione e di scegliere tecnologie che offrano difese integrate [45].

- *Principio del minimo privilegio:* Eseguire i processi dell'applicazione con i minimi privilegi necessari per limitare l'impatto di un'eventuale iniezione di codice [44].
- *Code Access Security (CAS) (per .NET):* In ambienti .NET, utilizzare il Code Access Security per limitare i permessi del codice eseguito. Sebbene il CAS sia stato in gran parte sostituito da altre tecnologie in versioni più recenti di .NET, la documentazione di Microsoft fornisce informazioni sul suo funzionamento e su come veniva utilizzato per la sicurezza [45]. È importante notare che per le nuove applicazioni .NET si raccomandano approcci differenti.
- *Ambienti di esecuzione sicuri (Sandbox):* Eseguire il codice fornito dall'utente in ambienti isolati (sandbox) per limitarne l'accesso al sistema è una tecnica efficace per mitigare il Code Injection. Diverse tecnologie e piattaforme offrono meccanismi di sandboxing, come i container Docker o le macchine virtuali [45].
- *Analisi del codice sorgente (Code review):* Come per le altre forme di injection, l'analisi del codice sorgente è fondamentale per identificare potenziali vulnerabilità. In questo caso, si cerca l'uso non sicuro di funzioni che interpretano o eseguono codice fornito dall'utente, come `eval()` in JavaScript o funzioni simili in altri linguaggi. Le guide di OWASP sul Code Review forniscono indicazioni utili [41].
- *Analisi dinamica (Dynamic Analysis) e Fuzzing:* Testare l'applicazione con input inattesi o maleformati, inclusi input che contengono codice in diversi linguaggi, può rivelare vulnerabilità di Code Injection. Strumenti di dynamic analysis e fuzzing possono automatizzare questo processo [46].
- *Analisi della memoria (Memory Analysis):* In caso di Code Injection che coinvolge codice compilato (come buffer overflow), l'analisi della memoria del processo può rivelare l'iniezione di codice malevolo. Questa tecnica, spesso utilizzata in analisi forensi o reverse engineering, permette di identificare la presenza di codice non previsto all'interno dello spazio di memoria del processo [45].
- *Strumenti di Static Application Security Testing (SAST) e Dynamic Application Security Testing (DAST):* Esistono strumenti SAST e DAST specifici per il rilevamento di vulnerabilità di Code Injection. Gli strumenti SAST analizzano il codice sorgente alla ricerca di pattern che indicano potenziali vulnerabilità, mentre gli strumenti DAST testano l'applicazione in esecuzione per identi-

ficare comportamenti anomali che potrebbero essere causati da Code Injection [47, 48].

5.4 Manipolazione del tagging VLAN

Questi attacchi mirano a superare la segmentazione delle reti VLAN.

- **VLAN Hopping:** Questa tecnica permette a un attaccante di superare la segmentazione logica delle reti VLAN (Virtual LAN). Esistono diverse tecniche di VLAN Hopping, come il double tagging (inserimento di due tag VLAN in un pacchetto) o lo switch spoofing (attraverso l'uso del Dynamic Trunking Protocol - DTP), che consentono all'attaccante di inviare traffico a VLAN diverse da quella a cui è connesso [49].

  Prerequisiti:

  - L'attaccante deve avere accesso a una porta di uno switch che gli consenta di "saltare" tra diverse VLAN, ovvero inviare traffico 802.1Q (tagged). Questo è possibile se la porta è configurata per il trunking o se lo switch è in grado di negoziare automaticamente il trunking (ad esempio, tramite DTP). [49].

  Fasi dell'attacco [49]:

  1. L'attaccante si connette a una porta dello switch configurata per il trunking o in grado di negoziare automaticamente il trunking.
  2. L'attaccante utilizza una delle tecniche di VLAN Hopping (es. double tagging, switch spoofing) per inviare traffico verso una VLAN diversa da quella a cui è connesso.
  3. Lo switch, se vulnerabile, inoltra il traffico verso la VLAN di destinazione, consentendo all'attaccante di comunicare con dispositivi presenti in altre VLAN.

  Contromisure [49]:

  - *Disabilitazione di DTP e utilizzo di trunking statico:* Disabilitare il Dynamic Trunking Protocol (DTP) e configurare manualmente le porte trunk con comandi come `switchport mode trunk` per le porte trunk e `switchport mode access` e `switchport nonegotiate` per le porte di accesso. Questo impedisce ad attaccanti di negoziare connessioni trunk non autorizzate.

- *Implementazione di Layer 2 Mapping:* Creare una rappresentazione visiva dell'infrastruttura di Layer 2, dettagliando assegnazioni VLAN, configurazioni delle porte degli switch e interconnessioni tra switch. Questo aiuta a identificare misconfigurazioni e potenziali vulnerabilità.
- *Mantenimento di un inventario dei dispositivi VLAN:* Mantenere un inventario dettagliato dei dispositivi connessi a ciascuna VLAN, includendo informazioni come tipo di dispositivo, indirizzi IP, indirizzi MAC e VLAN a cui sono assegnati. Questo aiuta a identificare dispositivi non autorizzati o modifiche impreviste nella rete.
- *Implementazione dell'autenticazione 802.1X:* Richiedere ai dispositivi di autenticarsi prima di poter accedere alla rete utilizzando il protocollo 802.1X. Questo assicura che solo i dispositivi autorizzati possano comunicare sulla rete.
- *Monitoraggio del traffico di rete:* Monitorare regolarmente il traffico di rete utilizzando strumenti di monitoraggio e sistemi di rilevamento delle intrusioni (IDS) per identificare anomalie, come traffico insolito tra VLAN, picchi di traffico inaspettati o altre irregolarità che potrebbero indicare attività dannose.
- *Configurazione corretta del trunking:* Assicurarsi che il trunking sia configurato correttamente, utilizzando protocolli sicuri e limitando le VLAN consentite su ciascuna porta trunk.
- *Utilizzo di VLAN private:* Implementare VLAN private per isolare ulteriormente gli host all'interno di una VLAN, limitando la comunicazione diretta tra di essi.
- *Implementazione di Port Security:* Configurare la Port Security sugli switch per limitare il numero di indirizzi MAC consentiti su ciascuna porta, prevenendo l'uso di tecniche di spoofing che possono essere utilizzate in attacchi di VLAN Hopping.
- *Aggiornamenti di sicurezza degli switch:* Mantenere gli switch aggiornati con le patch di sicurezza più recenti per correggere eventuali vulnerabilità note.

5.5 Attacchi basati su Esaurimento di Risorse (DoS/DDoS)

Questi attacchi mirano a consumare le risorse del sistema target, rendendolo incapace di rispondere alle richieste legittime. Possono essere suddivisi in

base alla tecnica utilizzata per esaurire le risorse:

- **UDP Flood:** L'attaccante inonda il server con un elevato volume di pacchetti UDP (User Datagram Protocol), saturando la banda disponibile e le risorse di elaborazione del server e della rete. L'UDP è un protocollo senza connessione (connectionless), il che rende questo tipo di attacco particolarmente efficace per sovraccaricare la rete, poiché non richiede l'handshake a tre vie del TCP e quindi può essere facilmente amplificato [50,51].

  Prerequisiti:

  - L'attaccante deve avere la capacità di generare un elevato volume di traffico UDP (spesso utilizzando una botnet) [50,51].

  Fasi dell'attacco:

  1. L'attaccante invia un grande numero di pacchetti UDP al server target, da diverse fonti (spesso utilizzando una botnet) [50,51].
  2. Il server target e la rete vengono sopraffatti dal volume di traffico UDP, rendendo il server non disponibile per gli utenti legittimi [50,51].

  Contromisure:

  - *Filtraggio del traffico UDP:* I firewall possono essere configurati per bloccare o limitare il traffico UDP da fonti sospette (ad esempio, indirizzi IP noti per attività malevole) o verso porte specifiche che non sono utilizzate dai servizi legittimi [51].
  - *Rate limiting:* Il rate limiting limita il numero di pacchetti UDP che possono essere ricevuti da un singolo indirizzo IP in un determinato periodo di tempo, prevenendo la saturazione della banda [51].
  - *Blackholing/Sinkholing:* Il blackholing (o sinkholing) consiste nell'instradare il traffico dannoso verso un "black hole" o "sinkhole", ovvero un percorso di rete che scarta il traffico senza inoltrarlo alla destinazione. Questa tecnica è comunemente utilizzata come tecnica di mitigazione DDoS, specialmente in combinazione con altre tecniche, per bloccare rapidamente grandi volumi di traffico dannoso [51].
  - *Utilizzo di servizi di mitigazione DDoS:* Servizi specializzati di mitigazione DDoS offrono protezioni avanzate contro gli attacchi

UDP Flood, tra cui scrubbing del traffico, analisi comportamentale e mitigazione automatica [50,51].

- *Monitoraggio del traffico UDP:* Un aumento improvviso e significativo del traffico UDP diretto al server indica un possibile attacco UDP Flood. Strumenti di monitoraggio del traffico di rete come `tcpdump`, Wireshark, o strumenti di analisi del flusso di rete (NetFlow, sFlow) possono essere utilizzati per analizzare il traffico UDP e identificare pattern anomali, come un elevato numero di pacchetti UDP diretti a porte specifiche o a un ampio range di porte [51].
- *Definizione di un limite minimo di velocità dati in ingresso:* Definire un limite minimo di velocità dati in ingresso e interrompere tutte le connessioni al di sotto di tale velocità. Questo può aiutare a mitigare gli attacchi Slow HTTP, che inviano richieste HTTP molto lentamente per esaurire le risorse del server [51].
- *Definizione di un limite di carico:* Definire un limite di carico, che specifica il numero di utenti autorizzati ad accedere a una determinata risorsa in un determinato momento [51].

- **ICMP Flood (Ping Flood):** L'attaccante inonda il server o la rete target con un elevato numero di pacchetti ICMP Echo Request (Ping), sovraccaricando la banda disponibile, le risorse di elaborazione (CPU) del server e le risorse di rete (router, firewall). Questo tipo di attacco è anche noto come "Ping of Death" quando i pacchetti ICMP sono di dimensioni eccessive, anche se questa variante è meno comune oggi a causa delle protezioni implementate nei sistemi operativi moderni. L'ICMP è un protocollo di livello di rete (Layer 3) utilizzato per la diagnostica e il controllo, ma il suo abuso può portare a Denial of Service [51,52].

  Prerequisiti:

  - L'attaccante deve avere la capacità di generare un elevato volume di traffico ICMP (spesso utilizzando una botnet) [51,52].

  Fasi dell'attacco [52]:

  1. L'attaccante invia un grande numero di pacchetti ICMP Echo Request (Ping) al server target, da diverse fonti (spesso utilizzando una botnet) .
  2. Il server target e la rete vengono sopraffatti dal volume di traffico ICMP, rendendo il server non disponibile per gli utenti legittimi.

Contromisure:

- *Filtraggio del traffico ICMP:* I firewall possono essere configurati per bloccare o limitare il traffico ICMP in entrata o in uscita, a seconda delle esigenze di sicurezza. È possibile filtrare per tipo di messaggio ICMP (ad esempio, bloccando solo gli Echo Request) o per indirizzo IP di origine [51].
- *Rate limiting:* Il rate limiting limita il numero di pacchetti ICMP che possono essere ricevuti da un singolo indirizzo IP in un determinato periodo di tempo, prevenendo la saturazione della banda e delle risorse del server [51].
- *Disabilitazione delle risposte ICMP Echo Reply:* In alcuni casi, per server che non necessitano di rispondere ai ping (ad esempio, server di produzione che non devono essere accessibili direttamente da internet), è possibile disabilitare completamente le risposte ICMP Echo Reply [51].
- *Monitoraggio del traffico ICMP:* Un aumento improvviso e significativo del traffico ICMP, in particolare di pacchetti Echo Request (tipo 8) e Echo Reply (tipo 0), indica un possibile attacco ICMP Flood. Strumenti di monitoraggio di rete come `ping`, `tcpdump` o Wireshark [51] possono essere utilizzati per analizzare il traffico ICMP e identificare pattern anomali, come un elevato numero di richieste ping provenienti da un singolo indirizzo IP o da una rete [52].
- *Utilizzo di software di terze parti:* Alcuni fornitori offrono servizi che aiutano a mitigare gli attacchi DDoS, incluso il filtraggio del traffico dannoso [51].
- *Monitoraggio della rete e analisi del traffico:* Il monitoraggio del traffico di rete è fondamentale per una risposta rapida alle minacce. L'analisi del traffico aiuta a determinare il tipo di minaccia, la sua gravità e la migliore risposta possibile [51].

- **HTTP Flood:** L'attaccante inonda un server web con un elevato numero di richieste HTTP (GET o POST), consumando le risorse del server e rendendolo non disponibile per gli utenti legittimi. [7, 53].

  Prerequisiti:

  - L'attaccante deve avere la capacità di generare un elevato volume di richieste HTTP (spesso utilizzando una botnet) [53].

Fasi dell'attacco [53]:

1. L'attaccante invia un grande numero di richieste HTTP al server target, da diverse fonti (spesso utilizzando una botnet).
2. Il server target viene sopraffatto dal volume di richieste HTTP, rendendo il server non disponibile per gli utenti legittimi.

Contromisure:

- *WAF (Web Application Firewall):* I WAF filtrano le richieste HTTP malevole, bloccando IP sospetti, implementando rate limiting a livello applicativo, proteggendo da attacchi di injection (SQL injection, XSS) e offrendo altre protezioni a livello applicativo [42].
- *Rate limiting:* Il rate limiting limita il numero di richieste HTTP che possono essere ricevute da un singolo indirizzo IP o da un determinato intervallo di IP in un determinato periodo di tempo, prevenendo la saturazione delle risorse del server [51].
- *CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart):* I CAPTCHA vengono utilizzati per distinguere tra traffico umano e traffico automatizzato (bot), bloccando le richieste provenienti da bot che partecipano all'attacco [54].
- *CDN (Content Delivery Network):* I CDN distribuiscono il carico su una rete di server distribuiti geograficamente, assorbendo parte del traffico dell'attacco e migliorando la disponibilità del sito web [55].
- *Reverse proxy:* Un reverse proxy intercetta le richieste HTTP prima che raggiungano il server web, offrendo funzionalità di caching, load balancing e protezione dagli attacchi, incluso il rate limiting e il blocco di IP sospetti [56].

- **Amplification Attacks:** Questi attacchi sfruttano vulnerabilità in server che rispondono a richieste relativamente piccole con risposte molto più grandi, amplificando significativamente il traffico dell'attaccante e causando un elevato volume di traffico diretto al target. Questo meccanismo di amplificazione rende questi attacchi particolarmente efficaci nel generare attacchi di tipo Denial of Service (DoS) e Distributed Denial of Service (DDoS). Esempi comuni includono DNS Amplification, NTP Amplification e memcached Amplification [57, 58]. Data la necessità di una vasta rete di "amplificatori" (server vulnerabili), questi attacchi sono quasi sempre di tipo DDoS.

Prerequisiti:

- L'attaccante deve avere la capacità di compromettere un numero elevato di endpoint (spesso attraverso una botnet) e di inviare richieste ai server che possono essere utilizzati per l'amplificazione [58].

Fasi dell'attacco [57, 58]:

1. L'attaccante utilizza endpoint compromessi per inviare richieste a server che possono essere utilizzati per l'amplificazione (ad esempio, server DNS aperti) con indirizzi IP falsificati che corrispondono all'indirizzo IP della vittima.
2. I server "amplificatori" rispondono alle richieste con risposte di grandi dimensioni, che vengono indirizzate all'indirizzo IP della vittima.
3. La vittima viene inondata da un volume massiccio di traffico, causando un denial of service.

Contromisure [57]:

- *Configurazione sicura dei server (hardening):* La mitigazione più efficace consiste nel configurare correttamente i server che potrebbero essere sfruttati come amplificatori. Questo include:
  - *DNS:* Disabilitare la ricorsione per i server DNS pubblici, implementare controlli di accesso (ad esempio, limitando le query ai soli client autorizzati) e utilizzare risposte di dimensione limitata (response rate limiting) .
  - *NTP:* Aggiornare i server NTP all'ultima versione per correggere eventuali vulnerabilità e disabilitare il supporto per comandi che possono essere abusati per l'amplificazione (come `monlist`).
- *Filtraggio del traffico:* Implementare filtri a livello di rete per bloccare il traffico indesiderato o anomalo, come risposte di dimensioni eccessive o traffico proveniente da indirizzi IP sospetti.
- *Rate limiting:* Implementare rate limiting a livello di rete per limitare il numero di richieste che possono essere inviate ai server che potrebbero essere utilizzati come amplificatori.
- *Utilizzo di servizi di mitigazione DDoS:* Servizi specializzati di mitigazione DDoS offrono protezioni avanzate contro gli attacchi

di amplificazione, tra cui il rilevamento e il blocco del traffico amplificato a livello di rete.

5.6 Attacchi basati su Starvation (Esaurimento mirato)

Questi attacchi privano selettivamente un servizio o una risorsa delle risorse necessarie, senza flooding massicci. Si concentrano sul consumo di risorse specifiche a livello applicativo, causando un degrado delle prestazioni o la completa indisponibilità del servizio.

- **CPU Starvation:** Consuma la CPU del server con calcoli complessi, cicli infiniti, o richieste inefficienti, impedendo al server di eseguire altre attività [59].

  Prerequisiti:

  – L'attaccante deve avere la capacità di eseguire codice sul server target o di inviare un elevato numero di richieste che richiedono un uso intensivo della CPU [59].

  Fasi dell'attacco [59]:

  1. L'attaccante esegue codice dannoso sul server target (ad esempio, tramite un exploit o un'applicazione compromessa) che esegue calcoli complessi o cicli infiniti, oppure invia un elevato numero di richieste che richiedono un uso intensivo della CPU (ad esempio, richieste di elaborazione di immagini complesse o di calcoli matematici).
  2. Il server target viene sovraccaricato e la sua CPU viene consumata completamente, impedendo al server di eseguire altre attività, inclusi i servizi legittimi.

  Contromisure:

  – *Monitoraggio dell'utilizzo della CPU:* Un utilizzo elevato e persistente della CPU, non giustificato dal carico di lavoro previsto, indica un possibile attacco. Si usano strumenti di monitoraggio del sistema come `top`, `htop` (Linux) o Task Manager (Windows) per osservare l'utilizzo della CPU per processo [60].

- *Limitazione del tempo di esecuzione dei processi:* Con `ulimit` (Linux) o policy di sistema (Windows) è possibile limitare il tempo massimo di CPU che un processo può utilizzare [61].
- *Ottimizzazione del codice:* L'ottimizzazione del codice applicativo per ridurre il carico sulla CPU è una pratica fondamentale per prevenire attacchi di CPU starvation e migliorare le prestazioni generali [62].
- *Risorse di calcolo dedicate (Scaling):* L'aumento delle risorse di calcolo, tramite scaling orizzontale (aggiunta di server) o verticale (hardware più potente), può mitigare gli effetti di un attacco di CPU starvation, distribuendo il carico su più risorse [63, 64].

- **Memory Starvation:** Riempie la memoria del server, causando blocchi, swapping eccessivo o crash del sistema [65].

  Prerequisiti:

  - L'attaccante deve avere la capacità di eseguire codice sul server target o di inviare un elevato numero di richieste che richiedono un uso intensivo della memoria [65].

  Fase dell'attacco [65]:

  1. L'attaccante esegue codice dannoso sul server target (ad esempio, tramite un exploit o un'applicazione compromessa) che alloca grandi quantità di memoria senza rilasciarla, oppure invia un elevato numero di richieste che richiedono un uso intensivo della memoria (ad esempio, richieste di caricamento di file di grandi dimensioni o di elaborazione di dati complessi).
  2. Il server target viene sovraccaricato e la sua memoria viene consumata completamente, causando blocchi, swapping eccessivo o crash del sistema, impedendo al server di eseguire altre attività, inclusi i servizi legittimi.

  Contromisure:

  - *Monitoraggio dell'utilizzo della memoria:* Un utilizzo elevato e persistente della memoria, con conseguente swapping, indica un possibile attacco. Si usano strumenti di monitoraggio del sistema come `free`, `vmstat` (Linux) o Resource Monitor (Windows) per monitorare l'utilizzo della memoria e lo swapping [61].

- *Limitazione della memoria allocabile per processo:* Tramite impostazioni del sistema operativo o del linguaggio di programmazione è possibile limitare la quantità di memoria che un processo può allocare [61].
- *Garbage collection:* Nei linguaggi che lo supportano, il garbage collection automatizza la gestione della memoria, rilasciando la memoria non più utilizzata e prevenendo memory leak che possono portare a memory starvation [66].
- *Aumento della RAM disponibile:* Aumentare la RAM disponibile del server può fornire maggiore capacità di memoria e ridurre gli effetti di un attacco di memory starvation [63, 64].

- **Resource Starvation (a livello applicativo):** Si verifica quando un processo non può essere supportato dalle risorse di calcolo disponibili. Ciò può essere causato dalla mancanza di risorse o dall'esistenza di più processi che competono per le stesse risorse di calcolo [67, 70]. Questo tipo di attacco esaurisce risorse specifiche a livello applicativo, come connessioni a database, file descriptor, thread o processi, impedendo il corretto funzionamento dell'applicazione e potenzialmente causando il crash del server. A differenza degli attacchi di flooding, il resource starvation si concentra sul consumo mirato di risorse limitate, anche con un basso volume di traffico.

  Prerequisiti:

  - L'attaccante deve avere la capacità di inviare un numero elevato di richieste all'applicazione che richiedono un uso intensivo di risorse specifiche (ad esempio, connessioni al database, file descriptor, thread o processi) [70].

  Fasi dell'attacco [70]:

  1. L'attaccante invia un numero elevato di richieste all'applicazione che richiedono un uso intensivo di risorse specifiche (ad esempio, connessioni al database, file descriptor, thread o processi).
  2. L'applicazione viene sovraccaricata e le risorse specifiche vengono consumate completamente, impedendo all'applicazione di funzionare correttamente e potenzialmente causando il crash del server.

  Contromisure:

- *Monitoraggio delle risorse applicative:* Un elevato numero di connessioni al database, thread o file descriptor aperti, non giustificato dall'attività normale dell'applicazione, indica un possibile attacco. Si usano strumenti specifici per il database (es. `SHOW STATUS` in MySQL, 'netstat', 'lsof') o l'applicazione per monitorare queste risorse [71, 72].
- *Connection pooling:* Il connection pooling riutilizza le connessioni al database, riducendo il sovraccarico di creazione e chiusura di nuove connessioni e prevenendo l'esaurimento delle connessioni disponibili [75].
- *Limitazione del numero di connessioni:* Limitare il numero di connessioni per client o per utente può prevenire l'esaurimento delle risorse [68–70].
- *Ottimizzazione delle query SQL:* L'ottimizzazione delle query SQL per minimizzare il carico sul database può ridurre il consumo di risorse e prevenire il resource starvation [72].
- *Gestione efficiente dei thread:* L'uso di thread pool e tecniche di programmazione concorrente efficiente può ottimizzare l'utilizzo dei thread e prevenire il loro esaurimento [73, 74].

- **Low and Slow:** Oltre a Slowloris, esistono altre tipologie di attacchi "low and slow" che consumano risorse specifiche a livello applicativo con un basso volume di traffico, sfruttando vulnerabilità o inefficienze nel codice dell'applicazione per esaurire le risorse del server in modo mirato [77].

  Prerequisiti:

  - L'attaccante deve avere la capacità di inviare richieste HTTP, o di altro tipo, al server web target [77].

  Fasi dell'attacco [77]:

  1. L'attaccante invia richieste al server web target che, pur essendo a basso volume, sfruttano vulnerabilità o inefficienze nel codice dell'applicazione.
  2. Queste richieste causano un consumo di risorse sproporzionato rispetto al volume di traffico, portando a un graduale esaurimento delle risorse del server.
  3. Il server, sovraccaricato, diventa lento o non disponibile per gli utenti legittimi.

Contromisure:

- *Analisi del comportamento delle applicazioni:* L'analisi del comportamento delle applicazioni, tramite log, metriche di performance, monitoraggio delle risorse e profilazione del traffico, può aiutare a identificare anomalie indicative di attacchi "low and slow", come un aumento dei tempi di risposta per determinate funzioni o un consumo anomalo di risorse per specifiche richieste [78].
- *Limitazione del timeout delle connessioni:* Timeout brevi per le connessioni possono mitigare attacchi che mantengono le connessioni aperte senza inviare dati o con scambio di dati incompleto, liberando risorse sul server [77].
- *Limitazione del numero di richieste per IP:* Limitare le richieste per IP può prevenire l'esaurimento delle risorse causato da un numero eccessivo di richieste, anche a basso volume, provenienti da un singolo attaccante [77].
- *Meccanismi di protezione specifici per le vulnerabilità delle applicazioni:* Implementare patch e contromisure specifiche per le vulnerabilità note delle applicazioni è essenziale per prevenire attacchi "low and slow" che le sfruttano [65].

Slowloris: Per questo attacco si sovraccarica un server web aprendo molte connessioni HTTP e mantenendole aperte con delle richieste parziali (invio di header incompleti), consumando così le risorse del server e impedendo nuove connessioni ad esso. È un attacco "low and slow" che mira a esaurire le risorse del server in modo subdolo, rendendolo non disponibile per gli utenti legittimi [76].

Prerequisiti:

- L'attaccante deve avere la capacità di stabilire numerose connessioni TCP con il server web target [76].

Fasi dell'attacco [76]:

1. L'attaccante stabilisce numerose connessioni TCP con il server web target.
2. L'attaccante invia richieste HTTP parziali (con header incompleti) attraverso queste connessioni, mantenendole attive ma non completate.
3. Il server web, in attesa del completamento delle richieste, mantiene aperte le connessioni, consumando risorse (principalmente connessioni disponibili).

4. Il server web, raggiunto il limite massimo di connessioni, non è più in grado di accettare nuove connessioni, rendendolo non disponibile per gli utenti legittimi.

Contromisure [76]:

- *Monitoraggio delle connessioni HTTP:* Un numero elevato di connessioni HTTP attive con tempi di mantenimento insolitamente lunghi (senza un corrispondente scambio di dati significativo) indica un possibile attacco Slowloris.
- *Limitazione del timeout delle connessioni:* Configurare timeout brevi per le connessioni HTTP inattive (o con scambio di dati incompleto) può chiudere le connessioni aperte da un attacco Slowloris, liberando risorse sul server.
- *Limitazione del numero di connessioni per IP:* Limitare il numero di connessioni che possono essere aperte da un singolo indirizzo IP può mitigare l'impatto di un attacco Slowloris, impedendo a un singolo attaccante di saturare le risorse del server.
- *Reverse proxy e WAF:* L'utilizzo di reverse proxy e WAF può intercettare e filtrare il traffico malevolo, proteggendo il server web dagli attacchi Slowloris. I WAF, in particolare, possono essere configurati per rilevare pattern specifici di Slowloris, come la presenza di header incompleti o la mancanza di completamento delle richieste.

5.7 Attacchi che sfruttano la rete locale e protocolli specifici

Questi attacchi si concentrano su specifici protocolli o sulla rete locale, causando interruzioni del servizio e compromettendo la disponibilità e, in alcuni casi, la riservatezza delle informazioni.

- **Sovraccarico della rete locale (MAC Flooding/Switch Spoofing):** L'attaccante inonda lo switch con pacchetti con indirizzi MAC falsificati, sovraccaricando la Content Addressable Memory (CAM), la tabella che associa gli indirizzi MAC alle porte dello switch. Quando la tabella CAM è piena, lo switch è costretto a comportarsi come un hub, inoltrando il traffico a tutte le porte e esponendo la rete ad attacchi di sniffing [79].

  Prerequisiti:

– L'attaccante deve avere accesso alla rete locale (LAN) per poter inviare pacchetti con indirizzi MAC falsificati [79].

Fasi dell'attacco [79]:

1. L'attaccante inonda lo switch con pacchetti contenenti indirizzi MAC sorgente falsificati.
2. Lo switch, non riconoscendo gli indirizzi MAC come validi, li aggiunge alla tabella CAM, che ha una dimensione limitata.
3. La tabella CAM si riempie rapidamente a causa dell'elevato numero di indirizzi MAC falsificati.
4. Quando la tabella CAM è piena, lo switch entra in "fail-open mode" e inizia a inondare tutte le porte con il traffico, comportandosi come un hub.
5. L'attaccante, collegato a una porta dello switch, può ora intercettare il traffico destinato ad altre macchine sulla rete (sniffing).

Contromisure [79]:

– *Port Security:* Configurare il port security su uno switch permette di limitare il numero di indirizzi MAC che possono essere appresi su una singola porta, prevenendo il MAC flooding.

– *VLANs:* La segmentazione della rete in VLAN limita l'impatto di un attacco MAC flooding, in quanto l'attacco ad una VLAN non influenza le altre.

– *MAC address filtering:* Implementare filtri basati su indirizzi MAC può bloccare il traffico proveniente da MAC address non autorizzati, ma questa tecnica è meno efficace contro attacchi di spoofing che cambiano continuamente l'indirizzo MAC.

– *Network monitoring:* L'utilizzo di strumenti di monitoraggio della rete può aiutare a rilevare segnali di MAC flooding, come un improvviso aumento del traffico o una diminuzione della velocità della rete.

5.8 Sfruttamento di Vulnerabilità Software (Crash)

A differenza degli attacchi basati su esaurimento di risorse, questi sfruttano delle vulnerabilità specifiche nel software per causarne un malfunzionamento, portando in queso modo all'interruzione del servizio. Si tratta quindi di

sfruttare un bug nel codice che causa un comportamento inatteso, come un accesso a memoria non valido o un errore di logica interna.

- **Buffer Overflow:** L'attaccante invia input che superano la dimensione del buffer di memoria allocato, sovrascrivendo altre aree di memoria e potenzialmente causando il crash del programma o l'esecuzione di codice arbitrario [80].

  Prerequisiti:

  – L'attaccante deve essere in grado di fornire input al programma target, sia direttamente (es. tramite input utente) che indirettamente (es. tramite file o rete) [80].

  Fasi dell'attacco [80]:

  1. L'attaccante identifica una vulnerabilità di buffer overflow nel programma target, ovvero un punto in cui è possibile scrivere dati oltre i limiti di un buffer allocato.
  2. L'attaccante prepara un input dannoso che, quando elaborato dal programma, supera la dimensione del buffer e sovrascrive altre aree di memoria.
  3. L'input dannoso può essere progettato per causare un crash del programma o per sovrascrivere l'instruction pointer (IP) con un indirizzo che punta a codice arbitrario (shellcode) fornito dall'attaccante, permettendogli di prendere il controllo del programma.
  4. L'attaccante invia l'input dannoso al programma target, innescando il buffer overflow e, se l'attacco ha successo, ottenendo il controllo del programma o causandone il crash.

  Contromisure:

  – *Input Validation:* Validare e sanificare gli input prima di utilizzarli per prevenire la scrittura oltre i limiti del buffer [65, 80].

  – *Bounds Checking:* Utilizzare controlli sui limiti degli array e dei buffer durante la programmazione per evitare la scrittura al di fuori dello spazio allocato [80].

  – *Utilizzo di linguaggi/librerie sicure:* Utilizzare linguaggi di programmazione o librerie che offrono protezioni automatiche contro i buffer overflow (es. linguaggi con garbage collection, librerie con controlli automatici sui buffer) [81].

- *Address Space Layout Randomization (ASLR):* Randomizzare la posizione delle aree di memoria (stack, heap, librerie) per rendere più difficile per un attaccante prevedere l'indirizzo di ritorno e quindi sfruttare un buffer overflow per eseguire codice arbitrario [82].
- *Data Execution Prevention (DEP):* Impedire l'esecuzione di codice da aree di memoria destinate ai dati (es. stack, heap) per bloccare l'esecuzione di codice iniettato tramite buffer overflow [83].

- **Integer Overflow:** L'attaccante manipola input numerici per causare un overflow di un intero, ovvero un risultato numerico che supera la capacità di rappresentazione del tipo di dato intero utilizzato. Questo può portare a comportamenti inattesi, come allocazioni di memoria errate, errori di calcolo o accessi a memoria non validi, e potenzialmente al crash del programma [84].

  Prerequisiti:

  - L'attaccante deve essere in grado di fornire input numerici al programma target, sia direttamente (es. tramite input utente) che indirettamente (es. tramite file o rete) [84].

  Fasi dell'attacco [84]:

  1. L'attaccante identifica una vulnerabilità di integer overflow nel programma target, ovvero un punto in cui un'operazione aritmetica può portare a un risultato al di fuori del range del tipo di dato utilizzato.
  2. L'attaccante prepara un input numerico dannoso che, quando elaborato dal programma, causa un integer overflow.
  3. L'integer overflow può portare a comportamenti inattesi, come allocazioni di memoria errate, errori di calcolo o accessi a memoria non validi, che possono essere sfruttati dall'attaccante per compromettere il programma.
  4. L'attaccante invia l'input numerico dannoso al programma target, innescando l'integer overflow e, se l'attacco ha successo, ottenendo il controllo del programma o causandone il crash.

  Contromisure:

  - *Input Validation:* Validare gli input numerici per assicurarsi che rientrino nei limiti consentiti dal tipo di dato utilizzato [65, 84].

- *Utilizzo di tipi di dato con range più ampio:* Utilizzare tipi di dato con un range di valori più ampio per ridurre la probabilità di overflow [84].
- *Controlli espliciti di overflow:* Inserire controlli espliciti nel codice per verificare la presenza di overflow prima di eseguire operazioni che potrebbero causarlo [84].
- *Linguaggi/Compilatori con Protezioni Integrate:* Utilizzare linguaggi di programmazione o compilatori che offrono protezioni automatiche contro gli integer overflow (es. Solidity 0.8.0 e successivi con controlli automatici, o l'utilizzo di librerie SafeMath) [84].

- **Format String Attack:** L'attaccante sfrutta vulnerabilità nella gestione delle stringhe di formato (usate in funzioni come `printf` in C) per leggere o scrivere in aree di memoria arbitrarie, potendo così causare il crash del programma o l'esecuzione di codice arbitrario [85].

  Prerequisiti:

  - L'attaccante deve essere in grado di fornire input al programma target che vengano interpretati come stringhe di formato [85].

  Fasi dell'attacco [85]:

  1. L'attaccante identifica una vulnerabilità di format string nel programma target, ovvero un punto in cui una stringa di formato controllata dall'utente viene utilizzata in una funzione come `printf`.
  2. L'attaccante prepara un input dannoso che contiene specificatori di formato speciali (es. `%n`, `%s`) che permettono di leggere o scrivere in aree di memoria arbitrarie.
  3. L'attaccante invia l'input dannoso al programma target, innescando l'attacco format string e, se l'attacco ha successo, ottenendo la capacità di leggere o scrivere in memoria arbitraria, potenzialmente causando il crash del programma o l'esecuzione di codice arbitrario.

  Contromisure:

  - *Non usare stringhe di formato controllate dall'utente:* Evitare di usare stringhe di formato fornite direttamente dall'utente. Utilizzare formati stringa fissi e passare le variabili come argomenti separati [65, 85].

- *Validazione degli input:* Implementare controlli per assicurarsi che gli input non contengano specificatori di formato pericolosi se proprio devono essere usati input esterni [85].

- **Teardrop Attack:** L'attaccante invia pacchetti IP frammentati con offset errati, causando problemi di riassemblaggio sul sistema target. Questo può portare a un sovraccarico del sistema, con conseguente crash o interruzione del servizio [86]. Questo attacco sfrutta vulnerabilità nell'implementazione del protocollo IP, in particolare nel processo di frammentazione e riassemblaggio dei pacchetti.

  Prerequisiti:

  - L'attaccante deve essere in grado di inviare pacchetti IP al sistema target.

  Fasi dell'attacco [86]:

  1. L'attaccante invia pacchetti IP frammentati al sistema target, modificando intenzionalmente gli offset in modo errato o sovrapposto.
  2. Il sistema target, durante il processo di riassemblaggio dei pacchetti, incontra difficoltà a causa degli offset errati, potenzialmente causando un sovraccarico di risorse (memoria, CPU).
  3. Il sovraccarico di risorse può portare a un crash del sistema o a un'interruzione del servizio, rendendolo non disponibile per gli utenti legittimi.

  Contromisure [86]:

  - *Firewall e IDS/IPS:* Firewall e sistemi di rilevamento/prevenzione delle intrusioni (IDS/IPS) possono essere configurati per rilevare anomalie nei pacchetti IP frammentati, come offset errati o sovrapposti, contribuendo a bloccare l'attacco.
  - *Aggiornamenti del sistema operativo:* Mantenere il sistema operativo aggiornato con le patch di sicurezza più recenti è fondamentale per correggere vulnerabilità note, comprese quelle relative alla gestione dei pacchetti IP frammentati.
  - *Blocco delle porte:* Disabilitare porte non necessarie, come le porte 139 e 445, può ridurre la superficie di attacco e prevenire l'exploit di vulnerabilità legate a specifici servizi.

- *Firewall:* L'utilizzo di un firewall robusto, sia a livello di sistema che di rete, può aiutare a filtrare il traffico dannoso e bloccare tentativi di attacco Teardrop.

- **LAND Attack:** L'attaccante invia un pacchetto SYN TCP con lo stesso indirizzo IP sorgente e destinazione, sfruttando una vulnerabilità nel protocollo TCP/IP. Questo causa confusione nel sistema target, che tenta di elaborare una richiesta di connessione non valida, portando potenzialmente a un blocco, un freeze o un crash del sistema [87].

  Prerequisiti:

  - L'attaccante deve essere in grado di inviare pacchetti TCP al sistema target.

  Fasi dell'attacco [87]:

  1. L'attaccante invia un pacchetto SYN TCP al sistema target, impostando sia l'indirizzo IP sorgente che quello di destinazione con l'indirizzo IP del sistema target stesso.
  2. Il sistema target riceve il pacchetto SYN con indirizzi IP uguali e, a causa di una vulnerabilità nel protocollo TCP/IP, entra in uno stato di confusione, tentando di elaborare una richiesta di connessione non valida.
  3. Questa elaborazione anomala può portare a un blocco, un freeze o un crash del sistema, rendendolo non disponibile per gli utenti legittimi.

  Contromisure [87]:

  - *Firewall e IDS/IPS:* Firewall e sistemi di rilevamento/prevenzione delle intrusioni (IDS/IPS) possono essere configurati per rilevare pacchetti TCP con indirizzi IP sorgente e destinazione identici, bloccando l'attacco.
  - *Aggiornamenti del sistema operativo:* Mantenere i sistemi operativi aggiornati con le patch di sicurezza più recenti è fondamentale per correggere vulnerabilità note, inclusi exploit come l'attacco LAND.
  - *Firewall:* L'utilizzo di un firewall configurato per bloccare pacchetti con indirizzi IP sorgente e destinazione identici è una misura preventiva efficace.

- *Monitoraggio della rete:* Implementare strumenti di monitoraggio della rete per rilevare anomalie nel traffico, come pacchetti con caratteristiche sospette (ad esempio, dimensione insolita) che potrebbero indicare un attacco LAND.

- **Ping of Death:** L'attaccante invia pacchetti ICMP (Ping) frammentati con dimensioni superiori a quelle consentite (65535 byte), sfruttando una vulnerabilità nel meccanismo di riassemblaggio dei pacchetti IP. Questo può causare un buffer overflow nel sistema target, portando a un blocco del servizio o, in alcuni casi, a un crash del sistema [88].

  Prerequisiti:

  - L'attaccante deve essere in grado di inviare pacchetti ICMP al sistema target.

  Fasi dell'attacco [88]:

  1. L'attaccante crea pacchetti ICMP (Ping) frammentati con dimensioni superiori al limite consentito di 65535 byte.
  2. I pacchetti vengono inviati al sistema target, che tenta di riassemblarli.
  3. A causa delle dimensioni eccessive e della vulnerabilità nel meccanismo di riassemblaggio, il sistema target può subire un buffer overflow, con conseguente blocco del servizio o crash del sistema.

  Contromisure [88]:

  - *Aggiornamenti del sistema operativo:* L'applicazione di patch di sicurezza per correggere la vulnerabilità nel riassemblaggio dei pacchetti IP è la principale misura di mitigazione.
  - *Configurazione del firewall:* I firewall possono essere configurati per limitare le dimensioni dei pacchetti ICMP o per bloccare pacchetti ICMP frammentati, riducendo il rischio di attacchi Ping of Death.
  - *Rilevamento con IDS/IPS:* Sistemi di rilevamento/prevenzione delle intrusioni (IDS/IPS) possono essere configurati per rilevare pacchetti ICMP di dimensioni eccessive o frammentati in modo anomalo, bloccando l'attacco.

Chapter 6

Mitigazione degli Attacchi di Rete con React-VEREFOO

6.1 Introduzione a React-VEREFOO

React-VEREFOO rappresenta un framework avanzato per la gestione automatizzata della sicurezza di rete, con particolare focus sulla configurazione e riconfigurazione dinamica dei firewall distribuiti [89–91]. è un aproccio che nasce dall'esigenza di proteggere le infrastrutture di rete moderne, caratterizzate da elevata dinamicità, complessità e scalabilità. La crescente adozione di paradigmi come Network Functions Virtualization (NFV) e Software-Defined Networking (SDN) ha portato ad un aumento della complessità nella gestione della sicurezza di rete, rendendo le soluzioni manuali inadeguate. Con NFV si intende la virtualizzazione di funzioni di rete (come firewall, NAT o load balancer) su piattaforme software, liberandole dal vincolo di dispositivi hardware dedicati; SDN, invece, prevede la separazione tra il piano di controllo e il piano dati, consentendo una gestione centralizzata e programmabile dell'infrastruttura. [89, 90].

A differenza dei firewall tradizionali, centralizzati e statici, i firewall distribuiti sono dislocati in più punti della rete, garantendo un controllo di sicurezza più granulare e vicino ai dispositivi finali. Tuttavia, la gestione manuale di queste istanze distribuite è un compito oneroso e soggetto a errori. React-VEREFOO affronta questa problematica automatizzando il posizionamento, la configurazione e la riconfigurazione dei firewall, ottimizzando così la sicurezza di rete [89–91].

6.1.1 Vantaggi della riconfigurazione automatizzata

L'approccio adottato da React-VEREFOO offre numerosi benefici rispetto alle soluzioni tradizionali:

- **Adattabilità:** Le politiche di sicurezza si adeguano dinamicamente alle esigenze della rete e alle condizioni operative [89–91].
- **Efficienza:** Ottimizza le risorse, evitando riconfigurazioni superflue e riducendo il carico computazionale [89–91].
- **Scalabilità:** Supporta reti di grandi dimensioni, garantendo un controllo efficace anche in ambienti altamente distribuiti [89–91].

6.2 Mitigazione degli Attacchi di Rete con React-VEREFOO

Questa sezione esplora come React-VEREFOO, grazie alla sua capacità di riconfigurazione dinamica e riallocazione delle istanze di firewall, possa contrastare diverse tipologie di attacchi di rete. Nella versione attuale non sono ancora supportate funzionalità come il rate limiting e l'integrazione con sistemi di Intrusion Detection System (IDS). Di conseguenza, alcune delle mitigazioni descritte si basano sulle capacità già implementate (quali filtri dinamici e riallocazione dei firewall), mentre altre rappresentano possibili estensioni future.

6.2.1 Attacchi Mitigabili Direttamente con React-VEREFOO

Questa sezione descrive gli attacchi che possono essere contrastati direttamente attraverso la riallocazione dinamica e la riconfigurazione dei firewall.

Spoofing

- **ARP Spoofing:** Attacco volto a falsificare l'associazione tra indirizzi IP e MAC address. ***Mitigabile direttamente*** mediante filtri ARP e controlli di coerenza (ad es., validazione che un IP corrisponda sempre allo stesso MAC address). React-VEREFOO può anche limitare il numero di risposte ARP per ogni MAC address per prevenire la saturazione della tabella ARP.
- **IP Spoofing:** Tecnica in cui l'attaccante falsifica l'indirizzo IP di origine dei pacchetti. ***Mitigabile direttamente*** applicando filtri su IP

address (ad es., bloccando pacchetti con indirizzi di origine non validi o privati provenienti dall'esterno) e, laddove possibile, tramite controlli di routing inverso.

- **MAC Spoofing:** Attacco che sfrutta la falsificazione del MAC address per ingannare dispositivi di rete. ***Mitigabile direttamente*** mediante filtri su MAC address (ad es., bloccando MAC non autorizzati) e limitando il numero di MAC appresi per porta.

Attacchi DoS/DDoS

  - **UDP Flood:** Inondazione di pacchetti UDP. ***Mitigabile direttamente*** in futuro tramite meccanismi di rate limiting e blocco dinamico del traffico anomalo; tuttavia, nella versione attuale, React-VEREFOO si basa esclusivamente sull'identificazione e blocco del traffico sospetto.
  - **ICMP Flood (Ping Flood):** Inondazione di pacchetti ICMP. ***Mitigabile direttamente*** con un'ipotetica estensione per il rate limiting e il blocco di ping flood. Attualmente, la protezione si fonda sulla riallocazione delle istanze di firewall per "avvicinare" i filtri ai punti di ingresso del traffico malevolo.
  - **HTTP Flood:** Inondazione di richieste HTTP. ***Mitigabile direttamente*** mediante analisi del traffico HTTP e blocco dinamico di richieste anomale. Anche se il rate limiting, che rappresenterebbe una protezione aggiuntiva, non è ancora supportato, la riallocazione delle istanze consente di proteggere server o applicazioni specifiche.
  - **Amplification Attacks:** Attacchi che sfruttano server (come DNS o NTP) per amplificare il volume di traffico malevolo. ***Mitigabile direttamente*** attraverso filtri sul traffico e blocco dinamico di risposte eccessive; il supporto a meccanismi di rate limiting potrà essere integrato in una versione futura.

6.2.2 Attacchi Mitigabili Parzialmente con Estensioni di React-VEREFOO

Questa sezione illustra attacchi che, sebbene non completamente mitigabili con le funzionalità di base, potrebbero beneficiare di estensioni future (ad esempio, l'integrazione con IDS e l'implementazione di meccanismi avanzati di rate limiting).

Reindirizzamento del Traffico (Hijacking)

- **DNS Hijacking:** Attacco che mira a reindirizzare il traffico tramite la manipolazione delle risposte DNS. ***Mitigabile parzialmente*** attraverso il blocco di risposte DNS sospette, mentre una protezione completa richiederebbe estensioni come l'integrazione con DNSSEC per validare l'autenticità delle risposte.

- **Session Hijacking:** Tecnica atta a intercettare o rubare identificativi di sessione. ***Mitigabile parzialmente*** con possibili estensioni che integrino sistemi di analisi comportamentale e di gestione delle sessioni, anche se il supporto IDS, che potrebbe facilitare questa mitigazione, non è attualmente disponibile.

Attacchi basati su Starvation (Esaurimento delle Risorse)

- **CPU Starvation:** Attacchi mirati a saturare la CPU di un sistema. ***Mitigabile parzialmente*** attraverso possibili estensioni per il rate limiting e il monitoraggio dell'utilizzo delle risorse, che permetterebbero di attivare contromisure come il blocco dinamico delle connessioni.

- **Memory Starvation:** Tecniche atte a esaurire la memoria disponibile. ***Mitigabile parzialmente*** con estensioni che consentano il monitoraggio dell'uso della memoria e l'adozione di politiche di controllo delle risorse.

- **Resource Starvation (a Livello Applicativo):** Attacchi che puntano a saturare risorse specifiche (ad es., connessioni al database o spazio su disco). ***Mitigabile parzialmente*** integrando React-VEREFOO con sistemi di monitoraggio delle applicazioni per attivare contromisure dinamiche.

- **Slowloris & Application Layer Attacks (Low and Slow):** Attacchi che mantengono connessioni HTTP aperte e a bassa velocità per esaurire le risorse del server. ***Mitigabile parzialmente*** con possibili estensioni per un rate limiting avanzato e un monitoraggio dei tempi di connessione, anche se la distinzione dal traffico legittimo risulta complessa.

Replay Attack

- Attacco in cui una comunicazione valida viene catturata e ritrasmessa successivamente. ***Non direttamente mitigabile*** da React-VEREFOO,

poiché il firewall non può distinguere tra pacchetti originali e pacchetti "replay". ***Mitigazione parziale (con estensioni):*** potenzialmente tramite l'integrazione con sistemi di autenticazione che utilizzano timestamp o numeri di sequenza, sebbene tale soluzione richieda una stretta collaborazione con altri componenti di sicurezza.

6.2.3 Attacchi Non Mitigabili con React-VEREFOO

Questa sezione elenca gli attacchi che non possono essere mitigati, nemmeno con estensioni ipotetiche, da React-VEREFOO, in quanto richiedono soluzioni di sicurezza specifiche o un'analisi a livello applicativo.

Manipolazione del Contenuto Web (Injection di Script)

- **Cross-Site Scripting (XSS), SQL Injection, Command Injection, Code Injection:** Attacchi che iniettano script o codice malevolo in applicazioni web. ***Non mitigabili*** con React-VEREFOO, in quanto questi attacchi sfruttano vulnerabilità nelle applicazioni web e richiedono l'intervento di un Web Application Firewall (WAF) per analizzare e filtrare il traffico HTTP/HTTPS a livello applicativo.

Sfruttamento di Vulnerabilità Software (Crash)

- **Buffer Overflow, Integer Overflow, Format String Attack, Teardrop Attack, Land Attack, Ping of Death:** Attacchi che sfruttano vulnerabilità nel codice delle applicazioni per causare crash o malfunzionamenti. ***Non mitigabili*** con React-VEREFOO, in quanto questi attacchi richiedono patch software e sistemi di rilevamento/prevenzione (IDS/IPS) per essere contrastati efficacemente. Pur potendo React-VEREFOO integrarsi con tali sistemi in future evoluzioni, da solo non è in grado di prevenire lo sfruttamento delle vulnerabilità.

Nota: La sicurezza di rete richiede un approccio "a strati". Mentre React-VEREFOO offre capacità dinamiche di riconfigurazione e riallocazione dei firewall, la protezione completa da attacchi complessi richiede l'integrazione con altre tecnologie (come WAF, IDS/IPS e sistemi di monitoraggio delle applicazioni).

Chapter 7

Conclusioni e Sviluppi Futuri

Questo lavoro aveva l'obiettivo di sviluppare un modello per la classificazione degli attacchi di rete, che cercasse di superare le attuali limitazioni delle tassonomie esistenti. Questa proposta integra i punti di forza di framework consolidati integrando insieme analisi dei prerequisiti per l'attaccante, fasi operative, contromisure adottabili, raggruppando. Tale modello risponde all'esigenza di affrontare in modo strutturato la crescente sofisticazione degli attacchi informatici. Questo tipo di approccio multidimensionale non solo facilita la comprensione delle tecniche adottate dagli attaccanti, ma fornisce anche una base solida per lo sviluppo di strategie difensive più efficaci.

Un ulteriore contributo di questo lavoro riguarda la valutazione dell'utilità di React-VEREFOO [91] in scenari di attacco. React-VEREFOO, grazie alla sua capacità di riallocazione e riconfigurazione automatica dei firewall, si propone come uno strumento innovativo per la mitigazione delle minacce in tempo reale. Questo lavoro propone anche come, attraverso possibili estensioni, il framework possa essere migliorato per contrastare attacchi che nella sua versione attuale non sono completamente mitigabili.

Uno degli obiettivi principali della tesi era quello di tradurre la conoscenza derivata dalla tassonomia in linee guida operative per la validazione di framework di sicurezza. In questo contesto, React-VEREFOO si distingue per la capacità di riallocazione e riconfigurazione automatica dei firewall, permettendo una risposta dinamica e proattiva alle minacce emergenti. La standardizzazione della terminologia e la categorizzazione dettagliata degli attacchi, come presentate nel modello, offrono un solido supporto per mappare le funzionalità di framework di sicurezza, facilitando così una valutazione comparativa delle contromisure adottate.

Nonostante i risultati promettenti, il rapido evolversi delle tecniche di attacco richiede aggiornamenti periodici della tassonomia per includere nuove metodologie e vulnerabilità emergenti. Infine, la validazione in ambienti re-

ali, come anche l'uso delle estensioni discusse riguardanti React-VEREFOO, necessita di ulteriori studi di caso e sperimentazioni. Un ulteriore sviluppo interessante è l'analisi approfondita dell'impatto (sia sulla sicurezza che sulle prestazioni) di queste possibili estensioni proposte per React-VEREFOO, al fine di comprenderne l'effettiva convenienza.

In un panorama in continua evoluzione, è fondamentale che la ricerca continui ad adattarsi alle nuove minacce, promuovendo un approccio integrato che unisca teoria, sperimentazione e applicazione pratica per garantire una protezione sempre più efficace dei sistemi informatici.

Bibliography

[1] CLUSIT. (2021). *Rapporto CLUSIT sulla sicurezza ICT in Italia*. https://clusit.it/wp-content/uploads/download/Rapporto-Clusit-ottobre-2021_web.pdf

[2] Check Point Software Technologies. (2023). *Cyber Security Report 2023*. https://www.checkpoint.com/resources/report-4fd2/report-cyber-security-report-2023

[3] Stallings, William. (2022). *Cryptography and Network Security: Principles and Practice*. Pearson. ISBN: 9789332585225, 9332585229.

[4] The Honeynet Project. (2004). *Know Your Enemy: Learning about Security Threats*. (2 ed.). Addison-Wesley. ISBN: 0321166469, 9780321166463.

[5] Kurose, James F., & Ross, Keith W. (2017). *Computer Networking: A Top-Down Approach* (7th ed.). Pearson. ISBN: 9780133594140, 0133594149.

[6] Forouzan, Behrouz A. (2013). *Data Communications and Networking* (5th ed.). McGraw-Hill. ISBN: 0071315861, 9780071315869.

[7] Zargar, S. T., Takabi, H., & Joshi, J. B. D. (2013). A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE Communications Surveys Tutorials*, *15*(4), 2046-2069. DOI: 10.1109/SURV.2013.031413.00127

[8] Kent, Stephen, Lynn, Charles, & Seo, Karen. (1999). Secure Border Gateway Protocol (Secure-BGP). In *Proceedings of the 6th Annual Network and Distributed System Security Symposium (NDSS '99')*. https://users.ece.cmu.edu/~adrian/731-sp04/readings/KLMS-SBGP.pdf

[9] Mirkovic, J., & Reiher, P. (2004). A taxonomy of DDoS attack and DDoS defense mechanisms. *ACM SIGCOMM Computer Communication Review*, *34*(2), 39-53. DOI: 10.1145/997150.997156

[10] Menezes, Alfred J., van Oorschot, Paul C., & Vanstone, Scott A. (1996). *Handbook of Applied Cryptography*. CRC Press. DOI: 10.1201/9781439821916

[11] National Institute of Standards and Technology (NIST). (2012). *Guide for Conducting Risk Assessments* (SP 800-30 Revision 1). https://doi.org/10.6028/NIST.SP.800-30r1

[12] International Organization for Standardization (ISO) & International Electrotechnical Commission (IEC). (2022). *Information technology — Security techniques — Information security risk management* (ISO/IEC 27005:2022).

[13] MITRE. CAPEC | Common Attack Pattern Enumeration and Classification. [Online]. Available: https://mitre-capec.org/

[14] MITRE ATT&CK. [Online]. Available: https://mitre-attack.org/

[15] Bader Al-Sada, Alireza Sadighian, and Gabriele Oligeri, *MITRE ATT&CK: State of the Art and Way Forward*. Hamad Bin Khalifa University, Qatar Foundation, Doha, Qatar. Accessed: March 4, 2025.

[16] Lockheed Martin. Lockheed Martin Cyber Kill Chain. [Online]. Available: https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html

[17] Wikipedia contributors. *ARP poisoning*. Wikipedia. [Online]. Available: https://it.wikipedia.org/wiki/ARP_poisoning

[18] impervarps. *ARP Spoofing*. [Online]. Available: https://www.impervarps.com/learn/application-security/arp-spoofing/

[19] Ramachandran, V., Nandi, S. (2005). Detecting ARP Spoofing: An Active Technique. In Information Systems Security (pp. 239-250). Springer, Berlin, Heidelberg.

[20] GeeksforGeeks. *What is ARP Spoofing Attack?*. GeeksforGeeks. [Online]. Available: https://www.geeksforgeeks.org/what-is-arp-spoofing-attack/

[21] Imperva. *IP address spoofing in application layer attacks*. [Online]. Available: https://www.imperva.com/learn/ddos/ip-spoofing/

[22] Ferguson, P., Senie, D. (2000). Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing. RFC 2827.

[23] Wikipedia contributors. *Reverse-path forwarding*. Wikipedia. [Online]. Available:https://en.wikipedia.org/wiki/Reverse-path_forwarding

[24] Kaspersky. *IP spoofing: what it is and how to prevent it*. [Online]. Available: https://www.kaspersky.com/resource-center/threats/ip-spoofing

[25] NordVPN. *What is a MAC spoofing attack? Learn how it works and how to detect and prevent it*. [Online]. Available: https://nordvpn.com/it/blog/mac-spoofing/

[26] Wikipedia contributors. *MAC spoofing*. Wikipedia. [Online]. Available: https://it.wikipedia.org/wiki/MAC_spoofing

[27] Ramonas, L. What is DNS hijacking? NordVPN. [Online]. Available: https://nordvpn.com/it/blog/what-is-dns-hijacking/

[28] Arends, R., et al. (2005). Protocol Modifications for the DNS Security Extensions. RFC 4035. https://www.rfc-editor.org/rfc/rfc4035.txt

[29] Šimkevičiūtė, S. What is session hijacking? NordVPN. [Online]. Available: https://nordvpn.com/it/blog/session-hijacking/

[30] Barth, A. (2011). HTTP State Management Mechanism. RFC 6265. https://www.rfc-editor.org/rfc/rfc6265.txt

[31] OWASP Foundation. (n.d.). Session Management Cheat Sheet. Retrieved from https://cheatsheetseries.owasp.org/cheatsheets/Session_Management_Cheat_Sheet.html

[32] NIST. (n.d.). Digital Identity Guidelines: Authentication and Lifecycle Management (SP 800-63B). https://nvlpubs.nist.gov/nistpubs/specialpublications/nist.sp.800-63b.pdf

[33] Cole, E. (2011). Network Security Bible (2nd ed.). Wiley.

[34] Einorytė, A. What is a replay attack? NordVPN. [Online]. Available: https://nordvpn.com/it/blog/replay-attack/

[35] OWASP. (n.d.). Injection. OWASP. Retrieved from https://owasp.org/Top10/A03_2021-Injection/.

[36] OWASP. *Input Validation Cheat Sheet*. OWASP Cheat Sheet Series. Available: https://cheatsheetseries.owasp.org/cheatsheets/Input_Validation_Cheat_Sheet.html.

[37] OWASP. (n.d.). Cross-Site Scripting (XSS). OWASP. Retrieved from https://owasp.org/www-community/attacks/xss/.

[38] Zieniūtė, U. XSS: Cos'è il cross-site scripting. NordVPN. [Online]. Available: https://nordvpn.com/it/blog/cose-il-xss/

[39] OWASP Foundation. (n.d.). *Cross Site Scripting (XSS) Prevention Cheat Sheet*. OWASP. Retrieved from https://cheatsheetseries.owasp.org/cheatsheets/Cross_Site_Scripting_Prevention_Cheat_Sheet.html.

[40] OWASP. (n.d.). SQL Injection. OWASP. Retrieved from https://owasp.org/www-project-top-ten/#injection.

[41] OWASP Foundation. (n.d.). OWASP Code Review Guide. OWASP. Retrieved from https://owasp.org/www-project-code-review-guide/.

[42] OWASP (Open Web Application Security Project). (n.d.). *Web Application Firewall*. OWASP. Retrieved from https://owasp.org/www-community/Web_Application_Firewall.

[43] OWASP. (n.d.). Command Injection. OWASP. Retrieved from https://owasp.org/www-project-top-ten/#owasp-top-ten-2021-a03-injection.

[44] CyberArk. (n.d.). Che cos'è il Privilegio Minimo? - Definizione. CyberArk. Retrieved from https://www.cyberark.com/it/what-is/least-privilege/.

[45] OWASP. Code Injection. https://owasp.org/www-community/attacks/Code_Injection

[46] OWASP (Open Web Application Security Project). (n.d.). *Fuzzing*. OWASP. Retrieved from https://owasp.org/www-community/Fuzzing.

[47] OWASP. Source Code Analysis Tools. https://owasp.org/www-community/Source_Code_Analysis_Tools

[48] OWASP. (n.d.). Dynamic Application Security Testing. https://owasp.org/www-project-devsecops-guideline/latest/02b-Dynamic-Application-Security-Testing

[49] Imperva. VLAN Hopping. [Online]. Available: https://www.imperva.com/learn/availability/vlan-hopping/

[50] NordVPN. *Cos'è un attacco DDoS e come difendersi?*. NordVPN Blog. Available: https://nordvpn.com/it/blog/attacco-ddos/ [Accessed: 2024-11-08]

[51] OWASP. Denial of Service Cheat Sheet. [Online]. Available: https://cheatsheetseries.owasp.org/cheatsheets/Denial_of_Service_Cheat_Sheet.html

[52] NordVPN. Ping flood attack: How it works and how you can defend against it. [Online]. Available: https://nordvpn.com/it/blog/ping-flood/

[53] Cloudflare. DDoS Mitigation. [Online]. Available: https://www.cloudflare.com/learning/ddos/ddos-mitigation/

[54] OWASP. (n.d.). OAT-009 CAPTCHA Defeat. https://owasp.org/www-project-automated-threats-to-web-applications/assets/oats/EN/OAT-009_CAPTCHA_Defeat

[55] IBM. Content Delivery Networks (CDN). [Online]. Available: https://www.ibm.com/it-it/topics/content-delivery-networks

[56] Wikipedia. Reverse proxy. [Online]. Available: https://it.wikipedia.org/wiki/Reverse_proxy

[57] Anagnostopoulos, M., et al. DNS amplification attack revisited. Computers Security 39 (2013): 475-485.

[58] Cloudflare. Che cos'è un attacco di amplificazione DNS? [Online]. Available: https://www.cloudflare.com/it-it/learning/ddos/dns-amplification-ddos-attack/

[59] Toprak, C., Turker, C., Erman, A. T. Detection of DHCP Starvation Attacks in Software Defined Networks: A Case Study. 2018 3rd International Conference on Computer Science and Engineering (UBMK), 2018, pp. 636-641.

[60] Bach, M. J. The design of the UNIX operating system. Prentice-Hall, Inc., 1986.

[61] Nemeth, E., Snyder, G., Hein, T. R. UNIX and Linux system administration handbook. Pearson Education, 2018.

[62] Martin, R. C. Clean code: a handbook of agile software craftsmanship. Pearson Education, 2018.

[63] Buyya, R., Yeo, C. S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, *25*(6), 599-616.

[64] Armbrust, M., Fox, A., Griffith, R., Joseph, A. D., Katz, R., Konwinski, A.,... Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, *53*(4), 50-58. DOI: https://doi.org/10.1145/1721654.1721672.

[65] Howard, Michael, & LeBlanc, David. (2002). *Writing Secure Code*. Pearson Education. ISBN: 9780735637405, 0735637407.

[66] Wikipedia contributors. Garbage collection (computer science). [Online]. Available: https://en.wikipedia.org/wiki/Garbage_collection_(computer_science)

[67] Stouffer, K., Pease, M., Tang, C., Zimmerman, T., Pillitteri, V., Lightman, S., Hahn, A., Saravia, S., Sherule, A., & Thompson, M. (2023). Guide to Operational Technology (OT) Security (NIST Special Publication 800-82 Revision 3). National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-82r3

[68] Özsu, M. T., Valduriez, P. (2011). *Principles of Distributed Database Systems*. Springer New York. ISBN: 9781441988348, 1441988343

[69] García-Molina, H., Ullman, J. D., Widom, J. (2008). *Database Systems: The Complete Book*. Pearson Education India.

[70] NordVPN. Resource exhaustion. [Online]. Available: https://nordvpn.com/it/cybersecurity/glossary/resource-exhaustion/

[71] Liu, L., & Reformat, M. (2005, April). Web application performance monitoring. In *Proceedings of the 14th international conference on World Wide Web* (pp. 642-650). DOI: https://doi.org/10.1145/1060745.1060812

[72] Date, C. J. (2006). *An Introduction to Database Systems*. Pearson Education. ISBN: 9788177585568, 8177585568

[73] Lea, D. (2000). *Concurrent Programming in Java: Design Principles and Patterns*. Addison-Wesley Professional.

[74] Goetz, B., Peierls, T., Bloch, J., Bowbeer, J., Holmes, D., Lea, D. (2006). *Java Concurrency in Practice*. Addison-Wesley Professional. ISBN: 9780321349606, 0321349601

[75] Connolly, T., Begg, C. Database systems: a practical approach to design, implementation, and management. Pearson Education, 2015.

[76] Imperva. Slowloris. [Online]. Available: https://www.imperva.com/learn/ddos/slowloris/

[77] Cloudflare. (2023). What is a low and slow attack?. Cloudflare Learning Center. Retrieved from https://www.cloudflare.com/it-it/learning/ddos/ddos-low-and-slow-attack/.

[78] The Honeynet Project. Know Your Enemy: Tracking Botnets. 2001. [Online]. Available: https://www.honeynet.org/papers/tracking-botnets/

[79] NordVPN. MAC flooding attack: Prevention and protection. [Online]. Available: https://nordvpn.com/it/blog/mac-flooding/

[80] OWASP. (n.d.). Buffer Overflow Attack. OWASP. Retrieved from https://owasp.org/www-community/attacks/Buffer_overflow_attack

[81] Dowd, M., McDonald, J., Schuh, J. (2006). *The Art of Software Security Assessment: Identifying and Preventing Software Vulnerabilities*. Pearson Education.

[82] IBM. Address Space Layout Randomization (ASLR). [Online]. Available: https://www.ibm.com/docs/en/zos/2.4.0?topic=overview-address-space-layout-randomization

[83] Microsoft. Data Execution Prevention. [Online]. Available: https://learn.microsoft.com/en-us/windows/win32/memory/data-execution-prevention

[84] OWASP. Integer Overflow. [Online]. Available: https://scs.owasp.org/sctop10/SC08-IntegerOverUnderFlow/

[85] NordVPN. Format string attack. [Online]. Available: https://nordvpn.com/it/cybersecurity/glossary/format-string-attack/

[86] Kaspersky. (n.d.). What is a Teardrop attack, and how to prevent them. Kaspersky. Retrieved from https://www.kaspersky.com/resource-center/definitions/teardrop-attack

[87] CDNetworks. (n.d.). What is a LAND Attack?. CDNetworks. Retrieved from https://www.cdnetworks.com/glossary/land-attacks/

[88] Wikipedia. (n.d.). Ping of Death. Wikipedia. Retrieved from https://it.wikipedia.org/wiki/Ping_of_Death

[89] Bringhenti, D., Marchetto, G., Sisto, R., Valenza, F., & Yusupov, J. (2020). Automated optimal firewall orchestration and configuration in virtualized networks. *NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*. IEEE, 2020.

[90] Bringhenti, D., Bussa, S., Sisto, R., Valenza, F. (2024). A Two-Fold Traffic Flow Model for Network Security Management. *IEEE Transactions on Network and Service Management*, *21*(2), 1147–1160.

[91] F. Pizzato, D. Bringhenti, R. Sisto, and F. Valenza, "Automatic and optimized firewall reconfiguration" in Proc. of IEEE/IFIP Network Operations and Management Symposium, Seoul, South Korea, May 2024.