POLITECNICO DI TORINO

DEPARTMENT OF CONTROL AND COMPUTER ENGINEERING (DAUIN)

Master Degree in Computer Engineering

Master Degree Thesis

Asset Discovery Tools Supporting Cybersecurity Inventory

Author: Giorgio Olivero

Advisor: Paolo Ernesto Prinetto
Co-Advisor: Fabio De Rosa, Nicolò Maunero

April, 2022

Acknowledgements

I would like to thank Prof. Paolo Prinetto for giving me the opportunity to work on this thesis and guiding me through the first part of the process. Then I would like to express my gratitude to Fabio De Rosa that was capable of inspiring me and helped me in thinking big and out of the box. A thanks go to the CINI team at Polito that supported me in this thesis path with a special mention to my Co-Advisor Nicolò Maunero that was unbelievably nice, assisting and motivating me during all the months spent working to this thesis. Then a message goes to the corinzi group: to whom it may concern, thank you. A big thank you to all of my childhood friends who were at my side at anytime in my life. I can't list all of you, but you are the main source of my happiness and I could not ask for better friends. A special thanks to Alexandra, I can not put in words how much I felt your support in these months and I am lucky to have an extraordinary person like in you in my life. And to conclude the biggest of my thanks go to my family. My father Carlo and my mother Claudia that did not stop for a single day to care about me and to give their support. You both are incredible, a model of what I want to become in my future. And finally the last thanks go to my sisters, Elena and Linda. You were my first teachers as a child and all of the goals that i have achieved and will hopefully achieve in my life are and will always be dedicated to you.

Abstract

The impact of digitalization and all kinds of computer sciences is very significant on everyday life, and its importance can be perceived in various fields that concern both the private and public life of a citizen. The rise of these new kind of technologies produced a lot of benefits but also some drawback, one above all the fact that there are now more risks involving security and the protection of sensitive data.

This has led to the creation of a new sub-field in the area of Information Technology (IT): cybersecurity. With this term is indicated the act of providing protection to certain assets (i.e., computer systems, data, networks, etc.) from information disclosure, theft or any kind of damage that could be caused to their integrity.

Since cybersecurity has become such an important topic, the Italian government has decided to set up a new legislation called *Perimetro di Sicurezza Nazionale Cibernetica* (*PSNC*) with the purpose of ensuring a high level of security to the entities that are part of it, i.e. public administrations and public or private institutions that provide essential functions to the State.

In the constitutive law it is explained that an entity belonging to the *PSNC* has various obligations, in particular for the purpose of this thesis the focus was put on the part regarding asset discovery and asset inventory. The idea was to be able to create, with the aid of open-source tools, a set of tools useful for the collection of the most important assets and to the creation of a map of the network.

Asset inventory is a crucial requirement in the domain of *PSNC* since it is needed to keep track of all the resources and to find possible vulnerabilities that could affect a certain system (e.g. outdated software, hardware malfunctions, etc.) and should be reported as soon as possible to reduce the window of exposure.

In this thesis different approaches were used and there was a notable part of study of the state of the art to find useful instruments in the field of asset discovery and asset inventory. This was done to understand which method

would be the most efficient to create a knowledge base that could eventually be used to perform security activities such as vulnerability assessment and risk analysis.

Another driving idea behind this thesis was being able to create a set of tools that could be used by all the entities belonging to the *PSNC*, even the ones that have less resources to invest in asset inventory. For this reason all the scripts and parts of code presented are based on open-source and free tools, instead of commercial ones, so that they can be accessible and used in a standard way.

In this document are presented all the strategies that were pursued with their positive and negative aspects, and it is also provided an explanation of the test environment and the context on which the research was focused.

Contents

List of Figures

Chapter 1

Introduction

Information Technology (IT) has become more and more important in recent times, since it affects both the private and the public lives of individuals in various ways. In fact, not only everyone is constantly using electronic devices to communicate and create large amount of data that are constantly sent across the Internet, but the same can be said for several companies in various fields, that nowadays can not avoid the use of IT. The spread of these new technologies comes with some drawbacks, one above all the risks derived from the possibility of these data to be stolen or compromised, and more in general all the problems related to physical or logical integrity of the assets.

The discipline that is in charge to face these problems is cybersecurity and the studies in this field are oriented towards the protection of the assets from possible attacks that could lead to a damage higher than a certain tolerance threshold. Cybersecurity is composed of multiple aspects, the *National Institute of Standards and Technology* (*NIST*) defines five core functions[1]:

- **Identify**: develop and implement an organizational understanding to manage cybersecurity risk. This core is the foundation to all the others;
- **Protect**: develop and implement appropriate safeguards to ensure delivery of the services;
- **Detect**: develop and implement appropriate activities to identify the occurrence of a cybersecurity event;
- **Respond**: develop and implement the activities to respond to a detected cybersecurity event;
- **Recover**: develop and implement appropriate activities to maintain

plans for resilience and to restore any capabilities or services that were impaired due to a cybersecurity incident.

The Italian government has decided to invest resources into the cybersecurity topic and in the "Legge 18 Novembre 2019, n.133" [20] a new instrument called *Perimetro di Sicurezza Nazionale Cibernetica* (*PSNC*) was set up; its aim is to ensure a high level of security of networks, informative systems services of public administrations and public or private institutions that provide an essential service for the State's interests.

The idea behind the *PSNC* is that there are some assets that need to be defended in a strong way and so the subjects that are included in the *PSNC* are required to follow certain measures and obligations that are defined in the constitutive law. In particular a subject included in the *PSNC* has to:

- Keep a list of the networks, informative systems and IT systems that are needed to provide the service for which the subject is part of the *PSNC*. Moreover this list needs to be updated at least once a year and must include relative architecture and components;
- Notify the incidents which impact the networks, informative systems and informatic systems to the italian *Computer Security Incident Response Team* (*CSIRT*) that promptly forwards these notifications to the *Dipartimento delle informazioni per la sicurezza* (*DIS*).

From these requirements is clear that identifying assets and managing them in the right way is crucial to make sure that a system is correctly protected, since this is the foundation for all the security techniques that are needed to make a system safe. Asset inventory is becoming increasingly important for companies and businesses that want to keep an high level of security and for this reason in recent years various commercial tools were created, with different costs and services provided.

The aim of this thesis is to explore this vast world of asset inventory and find the best solutions with the aim of meeting the requirements of the *PSNC*. In particular the idea is to look also at the economic aspect and try to combine different open-source instrument with the purpose of creating a new tool capable of speeding up and making the process of asset discovery and asset inventory easier and affordable also for entities belonging to the *PSNC* that have less resources to invest in this field.

To achieve this goal there was a notable part of study of the art in which the different existing tools were analyzed to figure out pros and cons and in general to understand the functioning and what could be created exploiting

these tools. The focus was put in developing solutions that could work in every environment and so all of the work was oriented towards simplicity and clarity of the final result.

The world of asset inventory is strictly correlated with another important branch of cybersecurity named *Vulnerability Assestment and Penetration Testing* (*VAPT*) that is the unification of two very important practices to guarantee security. Penetration Testing and Vulnerability Assessment perform two different tasks but are executed together to provide a more complete vulnerability analysis. In particular, Vulnerability Assessment is the process of identifying, quantifying, and prioritizing (or ranking) the vulnerabilities in a system, while a Penetration Test is an authorized simulated cyberattack on a computer system, performed to evaluate its security [30].

These kind of analysis need different tools that are designed to facilitate and make the execution of the testing faster. In this thesis some of them will be exploited and explained in their functionalities, but this world is in constant evolution and newer and better tools will be certainly created in the future.

Historically speaking asset inventory has been especially difficult and very manual labor (often requiring physical site visits) intensive for *Industrial Control System* (*ICS*) asset owners with "Grid-edge" devices (or field devices). The act of performing asset discovery can be done in two main ways:

- Actively, that is the more standard method and consists in using software that probes devices across a network or use discovering devices that attempt to log into devices in order to pull back a full inventory of connected applications. In this way the network could be slowed down and this could be a problem for time sensitive network like an *ICS*, which is why the second method is becoming more popular;

- Passively, that essentially consists in listening for traffic being sent around a network and removes the need to consume bandwidth. Nowadays, the progress in network security monitoring and protocol deep packet inspection has allowed asset owners to passively (non-intrusively) obtain real-time asset inventory information from devices communicating over serial or TCP/IP based communication channels by leveraging the built-in capabilities of grid-edge devices. Not only can passive asset management be achieved in this way but also passive network security monitoring which can detect a variety of network, security, and operational based anomalies.

The solutions explored on this thesis look at both of these approaches and try to exploit their strengths, looking for a way to create a tool that is proactive and able to identify new assets and managing them in the correct way in the most automatic way possible, without the use of too much manual labor and specialized personnel.

Asset inventory is needed to get a comprehensive inventory of all hardware, software, and network assets. For the purpose of this thesis and the requirements specified in the legislation of the *PSNC*, the final objective is to collect all the information about the asset through the asset discovery's tools and then create a list of all this information, so that it can be consulted easily and updated periodically.

The remainder of the thesis is organized as follows:

Chapter 2 gives an overview of asset discovery and asset inventory with explanation of their features and a presentation of one important protocol and one important tool;

Chapter 3 contains a brief explanation of the context in which this thesis takes part with an in-depth study of the *Perimetro di Sicurezza Nazionale Cibernetica* and its regulation.

Chapter 4 provides a study of the state of the art and a description of two commercial tools that are used in the asset management field;

In Chapter 5 are explained the functionality of the developed tool together with the process that was followed in the implementation;

In Chapter 6 the obtained results are presented along with an explanation of the reason behind some design choices;

Finally Chapter 7 presents the conclusions and explains which could be the possible steps in the research.

Chapter 2

Background - Asset Management

This chapter explores the concept of asset management with a focus on an important protocol used in this field and offers an overview on asset discovery and asset inventory with the reason for which they are important. Moreover a brief overview on the asset discovery tool used in this thesis is provided, with particular attention reserved to the techniques exploited by the tools.

2.1 Introduction

First of all is important to give a definition of an asset. The word asset can have different meanings to different fields of application. In the computer security area an asset is any data, device or other component of the environment that supports information-related activities. This can include hardware, software or any kind of confidential information [17]. In general an asset has to be protected from illicit access, use, disclosure, alteration, destruction, and/or theft, resulting in loss to the organization [18].

An IT asset can be distinguished in two major components, that are both fundamental to describe a system:

- Hardware asset, in this category are included all the physical aspect of the infrastructure with their components and the resources allocated to their needs.
- Software asset, namely all the information related to the programs installed on the devices that compose the infrastructure.

To these two definitions correspond respectively the concepts of *Hardware Asset Management*(*HAM*) and *Software Asset Management* (*SAM*) that are both crucial in the context of asset discovery and asset inventory to understand which are the assets that need to be collected and therefore protected.

2.2 Simple Network Management Protocol

To perform actions that concern configuration, managing and supervising the systems present in a network a internet protocol was defined by the *Internet Engineering Task Force* (*IETF*). The name of this protocol is *Simple Network Management Protocol* (*SNMP*). The information discovered through the use of this protocol are exposed in the form of variables and are organized in a *management information base* (*MIB*) which is used to describe the system status and configuration. On these variables remote queries can be applied using managing applications. Devices that typically support SNMP include cable modems, routers, switches, servers, workstations, printers, and more [21].

2.2.1 Overview and components

The typical use of SNMP consist in setting up one or more administrative computers, called *managers*, that have the task of monitoring or managing a group of other hosts or devices that form a computer network. On these managed systems a software component is installed and executed, that takes the name of *agent*, and its duty is to report the information via SNMP to the manager.

In a SNMP-managed network there are three key components:

- Managed devices, that are the subject of the analysis and the source of the information that are being collected;
- Agent, i.e. the software that runs on the managed devices;
- *Network Management Station* (*NMS*), that is the software running on the manager. It has the task of executing applications that monitor and control managed devices.

A managed device is a network node that implements an SNMP interface that allows unidirectional (read-only) or bidirectional (read and write) access to node-specific information. Managed devices exchange node-specific

information with the NMSs. In Figure 2.1 the principle of the SNMP communication is shown, and the role of the various actors is made clearer.

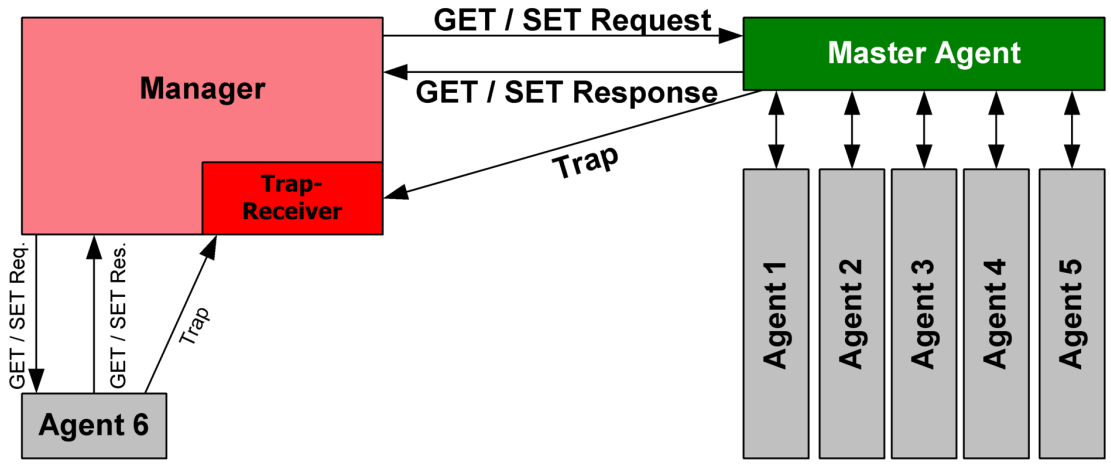

Figure 2.1. Principle of SNMP communication[1]

The SNMP protocol operates on layer 7 of OSI model and is composed by messages that are transported via *User Datagram Protocol* (*UDP*) [15]. Seven core *Protocol Data Units* (*PDUs*) are defined, five since SNMPv1, and the others were specified later in version 2 of the SNMP protocol. The construction of a SNMP PDU is shown in figure 2.2 and it is the same for every core PDUs.

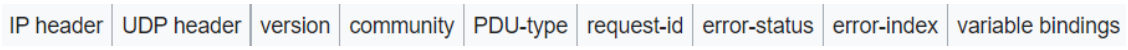

Figure 2.2. Composition of a SNMP Protocol Data Unit

The field *PDU-type* identifies the seven different SNMP PDUs that are the following:

- *GetRequest* - A manager-to-agent request that is sent in order to get the value of a variable or a list of variables;

[1] © Rene Britz, 2016

- *SetRequest* - A manager-to-agent request that is sent to change the value of a variable or a list of variables;
- *GetNextRequest* - A manager-to-agent request that is sent to discover if variables are available and their values;
- *GetBulkRequest* - Is an optimized version of *GetNextRequest*. In this case multiple iterations of *GetNextRequest* are sent. This was one of the PDUs type introducted in version 2 of the SNMP protocol;
- *Response* - It is the instrument with which variable bindings and acknowledgements are returned from agent to manager in response to the various manager-to-agent requests;
- *Trap* - It is an asynchronous notification from agent to manager. The difference between a Trap and a Response is that this type of PDU is not explicitly requested by the manager and are used to notify significant events;
- *InformRequest* - Originally was a manager-to-manager communication but in recent implementation also agent-to-manager messages are possible. The task of this PDU is to acknowledge that an asynchronous notification was received.

2.3 Asset Discovery

The first step to perform asset management consists in detecting and collecting all the information about the various assets present in a network. This process of finding and listing the IT assets and monitoring them on a regular basis is called *Asset Discovery* and it is crucial to maintain integrity of data that is spread across the infrastructure of an organization.

The basic scenario in which asset discovery takes place is through an application installed on a specific asset like a laptop that is responsible to scan the entire infrastructure to collect information about the assets in the network. This type of asset discovery is certainly preferable to a manual listing of all the data, since it guarantees that the information are up to date and it is not subject to human error. For this reason in recent times a lot of tools are being designed and the interest for this topic is increasingly high. In the following section an overview of one important tool is given.

2.3.1 Nmap

Figure 2.3. *Nmap Logo*

One of the most used software in the asset discovery field is *Nmap*. The most important aspect that contributed to its popularity is certainly that Nmap is freely available and open-source and for this reason a lot of other asset discovery software are based on Nmap for what concerns network scanning and security auditing. Nmap started as Linux utility published as an article with source code included [11], but it was later ported to other systems including Windows, macOS, and BSD [25].

Nmap offers a lot of different features [12], including:

- Host discovery - Possibility to perform a scan of the network to look for hosts that respond to a TCP or an ICMP request or have a particular open port;
- Port scanning - Enumerating and listing the open ports of an host;
- Version Detection - Interrogating active services of the hosts in the network to determine application name and version number;
- TCP/IP stack fingerprinting - Determining the operating system and the hardware characteristics of network devices based on observations of network activity of said devices;

- Scriptable interaction with the target - newest feature, that thanks to *Nmap Script Engine* (*NSE*)[27] permits to use simple scripts to automate a wide variety of tasks.

In addition to this Nmap can provide other information on targets such as reverse DNS names, device types or MAC addresses.

To perform these kind of analysis Nmap exploits various methods, so in this section a brief explanation of the most important procedure needed to provide the features mentioned above is given.

Nmap Host Discovery Techniques

Since host discovery needs are so diverse and the environment in which Nmap is used could differ greatly depending on the situation, this tool offers a wide variety of options to customize the techniques used. Usually in a network is common to have a small percentage of active (i.e. being used by an host or device) IPs at any given time. The strength of a tool like Nmap is the possibility to perform host discovery and find the active machines in a wide and sparsely allocated area of IP addresses in a very short time [23].

The standard way of performing host discovery in Nmap, not considering therefore the use of specific options that can change this behaviour, is the sending by Nmap of an ICMP echo request, a TCP SYN packet to port 443, a TCP ACK packet to port 80, and an ICMP timestamp request. (For IPv6, the ICMP timestamp request is omitted because it is not part of ICMPv6.) There is an exception to this regarding a local ethernet network in which ARP (for IPv4) and Neighbor Discovery (for IPv6) are used instead. If a UNIX user is unpriviliged the default scan will involve the use of SYN packets to port 80 and 443 using the *connect* system call.

In addition to the standard method of performing host discovery mentioned above there are still other techniques that can be specified by the use of particular options. These are UDP ping, SCTP INIT ping (i.e. the sending of a SCTP packet containing a minimal INIT chunk, used to suggest to the remote system that Nmap is trying to establish an association) and finally the IP protocol ping, that is one of the newer host discovery option and consists in sending IP packets with the specified protocol number set in their IP header.

Nmap Port Scanning Techniques

In the art of port scanning there are plenty of techniques and finding the right one that can be suitable to a certain task can be very difficult, so it is important to know the differences between all of those to understand which one to use in a specific context. Nmap supports all of these techniques and so in this section a brief explanation of them is provided [26]. Nearly all the scans that are listed below requires the user to have the administrator privileges.

- TCP SYN scan: the quickest and simplest scan, capable of probing thousands of ports per seconds on fast network not hampered by restrictive firewalls. It is also relatively unobtrusive and stealthy since the TCP connection is not completed;
- TCP connect scan: the default option if TCP SYN scan is not available, for example if a user does not have raw packet privileges. It exploits the *connect* system call and it is less efficient and more detectable than TCP SYN scan and therefore the latter is preferable when available [9].
- UDP scans: a lot of services uses UDP instead of TCP (e.g. DNS, SNMP, DHCP, etc.) and therefore these ports can be scanned to detect these services. UDP scan works by sending a packet to every targeted port;
- SCTP INIT scan: this is a relatively new alternative to TCP and UDP. The scan is the SCTP equivalent of a TCP SYN scan, and maintains the qualities of quickness and stealthiness;
- TCP NULL, FIN, and Xmas scans: these three scans exploit a flaw in the TCP RFC that permits to differentiate between open and closed ports. The key advantage of this type of scans is that they can overcome certain non-stateful firewalls and packet filtering routers;
- TCP ACK scan: differently from the scans cited above this scan never determines open ports and is used mainly to map out firewall rulesets, determining if they are stateful or not and which ports are filtered;
- TCP Window scan: very similar to the ACK scan but it exploits an implementation detail that can differentiate open ports from closed ones in certain systems. Relying on an implementation detail present in just a minority of the systems, this scan is not totally trustworthy and system that does not support it will usually return all ports closed;

- SCTP COOKIE ECHO scan: this is a more advanced SCTP scan. It has the advantage of being more stealth in comparison to a traditional SCTP INIT scan, but it comes with the downside of not being able to differentiate between open and filtered ports;

- IP protocol scan: this is not properly a port scan since it cycles through IP protocol numbers rather than TCP and UDP ports number. The aim of this scan is to determine which IP protocols are supported by the target systems.

Nmap OS detection

When a scan of a network is performed the ideal outcome is the collection of the larger number of information possible, and not just a list of active IPs and ports. OS detection is an important feature offered by Nmap [24] since it allows to understand, in most cases, not only which type of machine is the owner of a certain IP address, but also the exact version of the OS installed on the aforementioned machine. This can be very useful when performing every kind of vulnerability assessment but also, as it is used for the purpose of this thesis, to perform asset discovery and asset inventory.

The techniques behind Nmap OS detection are various and can be very complicated to fully understand, because they exploits a lot of different implementations and specific characteristics of the operative systems installed on the machine target of the scan. In general the OS detection method provided by Nmap consists in performing various tests at the end of which a fingerprint with the results of these text is produced. A typical example of a fingerprint produced by a Nmap OS detection scan can be seen in figure 2.4. The idea behind the creation of a fingerprint is that it can be compared with the ones created before and included in the Nmap database to understand which OS is running on the machine.

The standard methods of OS detection are divided into two macrocategories:

1. TCP/IP Fingerprinting: in this category, as the name suggests, are included all the tests related to TCP/IP ports and in general the network protocols (e.g. UDP, ICMP) in their IPv4 version. The way in which Nmap works is by sending up to 16 TCP, UDP and ICMP probes to known open ports of the target machines. The purpose of these probes is to exploit various ambiguities in the standard protocol RFCs. The responses given to these probes is what is then combined to form the

```
OS:SCAN(V=5.05BETA1%D=8/23%OT=22%CT=1%CU=42341%PV=N%DS=0%DC=L%G=Y%TM=4A91CB
OS:90%P=i686-pc-linux-gnu)SEQ(SP=C9%GCD=1%ISR=CF%TI=Z%CI=Z%II=I%TS=A)OPS(O1
OS:=M400CST11NW5%O2=M400CST11NW5%O3=M400CNNT11NW5%O4=M400CST11NW5%O5=M400CS
OS:T11NW5%O6=M400CST11)WIN(W1=8000%W2=8000%W3=8000%W4=8000%W5=8000%W6=8000)
OS:ECN(R=Y%DF=Y%T=40%W=8018%O=M400CNNSNW5%CC=N%Q=)T1(R=Y%DF=Y%T=40%S=O%A=S+
OS:%F=AS%RD=0%Q=)T2(R=N)T3(R=Y%DF=Y%T=40%W=8000%S=O%A=S+%F=AS%O=M400CST11NW
OS:5%RD=0%Q=)T4(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)T5(R=Y%DF=Y%T=40%W
OS:=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0%S=A%A=Z%F=R%O=%RD=0%Q=)
OS:T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S+%F=AR%O=%RD=0%Q=)U1(R=Y%DF=N%T=40%IPL=164%U
OS:N=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI=N%T=40%CD=S)
```

Figure 2.4. A typical subject fingerprint

fingerprint. All the packets are IPv4 with random IP ID value and are resent at least once if no response is provided the first time;

2. IPv6 Fingerprinting: the OS detection engine specialized for IPv6 is similar to the one mentioned above but has some differences and for this reason is separated to the one used for IPv4 OS detection. At a high level the technique is the same, that is sending probes, collecting responses and match the generated fingerprints with the one found in the database. The differences are in the specific probes used and in the way they are matched.

The world of OS detection is vast and a lot techniques are possible and Nmap supports many of them, but not all. The results provided at the end of a Nmap OS detection scan are given in association with a percentage that describes the degree of confidence of the guess that Nmap made, and it is possible using different options to specify if the guess should be more or less aggressive. In case the OS could not be detected in any way, the fingerprint that was generated during the test is printed and if the OS that is running on the target machine is known, Nmap gives the possibility to report this fingerprint with the OS version, so that the database of known operating system can be expanded.

2.4 Asset Inventory

The consequent step after having collected the information about the assets is to use this data to build an inventory which makes the work of performing

vulnerability assessment easier and more in general permits to have a clear view of the assets that compose a specific infrastructure. This practice is called *Asset Inventory* and it is now required by a lot of different standards, texts and regulation relative to IT government and Information Security, and here some examples are given[14]:

- The *ISO/IEC Standard 27001*[17] requires that "any assets associated with information and information processing facilities need to be identified and managed over the lifecycle, always up to date. A register or inventory of those assets has to be put together that shows how they are managed and controlled, based around their importance";
- The *Cobit Framework 4.1* [16] includes an inventory of components and critical systems (DS4), of hardware and software elements (DS9), of stored and archived media (DS11.3), of physical structures (DS12) that also needs to be classified following the process of risk management and finally of sensitive IT assets (DS13.4);
- The *NIST* in addition to the already cited *NIST Cybersecurity Framework*[1] mentions asset inventory also in another publication [22] where it is required that the company developes, documents and maintain an inventory of system components;
- The *PCI DSS Standard* [2] says that asset inventory is "a comprehensive list of assets that are in scope for the risk assessment, for example, software, hardware, networking and communications infrastructure and personnel."

As for asset discovery, also for the part of asset inventory a lot of new tools are being developed and this world is in continuous evolution. This field can be very important also for economic reasons, since the topic is the correct management of asset that potentially have a certain value, and so the tools are often quite expensive. In this thesis as explained in the introduction the aim is to try to favour open-source software. In Chapter 4 two of the main solutions available on the market will be presented.

Chapter 3

Perimetro di Sicurezza Nazionale Cibernetica

In this chapter a brief clarification of the context in which this thesis takes part is provided, with a focus on the *Perimetro di Sicurezza Nazionale Cibernetica*(*PSNC*), the goals that are set behind its creation and an explanation of the legislation and the various terminologies used in the constitutive law and in the subsequent *Decreti del Presidente del Consiglio dei Ministri*(*DPCM*).

3.1 Hystory

The *PSNC* is a legislation that the Italian Government has enacted to increase security towards specific entities that provide services that are considered essential for the interest of the State. The cybersecurity topic was becoming really important in the latest years and in fact the *PSNC* was not the first legislation that deals with this subject. In year 2013 the *European Commission* redacted a plan [10] in which the first strategy concerning cybersecurity are mentioned and three years later, in year 2016, following this strategy a Network and Information Security (*NIS*) Directive was published [8] and this was a first step in the direction of a more organized approach in the field of cybersecurity as European Union. This directive was later made operational in Italy in year 2018, with the issue of the decree-law 65 [7] that became effective on June 24th. These were all the preliminary steps prior to the effective definition of the *PSNC* that effectively occurred in year 2019 in the decree-law 133 [20] in which for the first time the perimeter is mentioned

and all the foundations are laid for the other legislation to come in the later years.

3.2 Goals and organization

The aim of the *PSNC* is to give a set of rules to certain entities that are considered an integral part of the system-country since they provide certain services that are fundamental and without which the State would be damaged from a civil, social or economic point of view. Concretely, as stated in chapter 1, an entity that is part of the perimeter has to keep a list of the IT assets and communicate them to the authority. In addition to this, every organization belonging to the *PSNC* has to notify every incidents to *CSIRT* authority. The list of entities belonging to the *PSNC* is updated periodically by the Italian Government trough Prime Ministerial Decree, following the indication given by the different *Comitato interministeriale per la sicurezza della Repubblica* (*CISR*) administration that work in their area of competence. The institution that is mandated to guarantee that the entities in the *PSNC* follow the rules written in the legislation is the *Centro di Valutazione e Certificazione Nazionale* (*CVCN*), that was set up by the minister of economic development and has also the task eventually conditions and hardware and software tests to be integrated also in collaboration with *CISR* administrations. If the various obligation are not followed after a certain specified time the entity can incur in various fines that are specified in the law.

It is important to note that this organization is very recent and a lot of changes have been made and will be probably made in the future, so this is just the foundation and all the future modifications will be operated on this basis.

3.3 Legislation

The body of law that regulate the *PSNC* is vast and is being updated periodically thanks to the redaction of four DPCM, that contribute to define the rules and the requirements on the various topics inserted in the PSNC constitutive law. At the time of this thesis the *PSNC* regulations are described in three DPCM, with a fourth that is yet to come, and one Presidential Decree (*DPR*). A brief description of these decrees is given in this section.

3.3.1 DPCM 1

The first of the DPCM is the number 131 enacted on July 30th, 2020[5] and it has basically the objective of defining which are the entities that are considered essential or the services that are considered essential to the State. In the DCPM a list is redacted of the sectors of activity in which the entities belonging to the perimeter are individuated, in addition to the sector of Government and these are: interior, defense, space and aerospace, energy, telecommunications, economy and finance, transportation, digital services and critical technologies. Moreover in the DPCM there is the definition of the criteria with which an entity in the perimeter has to arrange and update the list of ICT assets with indication of networks, information systems and information services that compose them, and this has to be updated at least yearly to abide by the rules. Finally, in this decree a board for the implementation of the *PSNC* is established, and its task is to support the aforementioned CISR. This is considered the first technical addition to the constitutive law, but it just added some details and the real implementation of the *PSNC* still needed other regulation to be considered complete.

3.3.2 DPR

In this second legal act, enacted by the Italian President of The Republic Sergio Mattarella on February 5th, 2021 [6], the most important topic that is covered are the methods and procedures relating to the functionality of the *CVCN*. In particular the DPR identifies the methods, procedures and terms to follow for the purpose of the evaluation by the CVCN governing the acquisition of the ICT assets identified in the previous DPCM. Therefore this DPR contains some technical details about time terms and procedures that the entities belonging to the perimeter has to follow. Moreover there is a section dedicated to how the verification procedure is performed by the CVCN and the other entities that are in charge of this task.

3.3.3 DPCM 2

The second of the four DPCM is the number 81 [3] enacted on April 14th, 2021. In this decree the rules in matter of notification of the incidents having an impact on the ICT assets are established and in particular there are 2 tables in attachment that categorize the different type of incident based on the severity and the time that an entity has to notify them to the *CSIRT*. These rules entered into force since January 1st, 2022.

3.3.4 DPCM 3

This third DPCM was enacted on June 15th, 2021 [4] and contains the definition of the ICT assets categories to be employed in the *PSNC*. These categories, must be updated at least once a year, coherently with the improvements and technology evolution. Four categories are identified, and they are defined on the basis of technical criteria specified in the constitutive law. The following are the categories included in the table attached to the DPCM:

1. Hardware and software that are involved in telecommunication networks services and activities;
2. Hardware and software that perform actions of security on telecommunication networks and data;
3. Hardware and software used for data acquisition, monitoring, control, implementation and automation of telecommunication networks;
4. Software applications for security mechanisms implementation.

3.3.5 DPCM 4

The fourth DPCM is yet to be enacted at the time of this thesis and will be the last of the Prime Minister's Decree on the topic of the *PSNC*. With this one the perimeter will finally be considered completed.

Chapter 4

State of The Art

As explained in Section 2.4 the number of tools that are designed to perform asset inventory is increasingly high and for this reason there are a lot of characteristics to evaluate when choosing the most suitable for the context in which the inventory has to be built. For this reason in this chapter a list of some important aspect that characterize an asset inventory tool is given [13].

- Control of IT asset: since one of the most important aspect of asset inventory is to have a clear view of the assets in the infrastructure to facilitate their management, it is crucial that the tool chosen to make the inventory is simple to use and designed in a way that all the elements are easily visible;
- Configurability and customization: another fundamental aspect of an asset inventory tool is the possibility to customize it according to the environment in which the tool will be used;
- Good reporting functionality: this point is very important since a tool that has the ability to report information about the collected assets permits to rapidly understand if everything is working correctly (for instance from a cybersecurity point of view a tool can report if a software is outdated, if an hardware had a malfunction, etc.).
- Cost: finally it is also important to evaluate the price requested for the solution, in relation to the context. This can include the cost for the license of the tool but also the support provided.

In this chapter two solutions will be presented, the first one is open-source and free while the second one has a trial version valid for a number of assets

lower than 100, then there is a yearly price to pay that is dependant on the number of assets. These overviews on the two tools will permit to understand the reason behind the choice of a commercial rather than a free solution or vice versa depending on the case scenario.

4.1 Snipe-IT

Snipe-IT is an free open-source (*FOSS*) project developed by *Grokability* built on *Laravel* that was made for IT asset management[28]. It is a web-based software, it must be run on a web server and accessed trough a web browser. All the code behind the project is available on *Github*[29], and the service is free if self hosted while there are different type of subscription for what concerns an hosted plan, and in this last case also complete technical support is supplied.

Snipe-IT has a very user-friendly graphical interface (4.1) and its aim is to make the work of asset inventory as easy as possible. The assets that fill the database can be added manually or imported from a csv file or via a specific API designed specifically for this reason. The idea behind this project is replacing the typical spreadsheets that are often used also in the field of IT asset inventory with something that is more organized and in which the various information can be easily accessed and modified at the time of need.

The features offered by *Snipe-IT* are numerous and this is just a brief list of what this tool provides:

- Easily see which assets are assigned, to whom, and their physical location;
- Asset Models let you group common features;
- Email alerts for expiring warranties and licenses;
- Quick and easy asset auditing;
- Add custom fields for additional asset attributes.

All of this things are designed with security in mind. For this reason a lot of security features are applied to keep the data safe, and in any case since Snipe-IT runs on a web server that could be self hosted all the correct steps to guarantee an high level of security are explained in the manual.

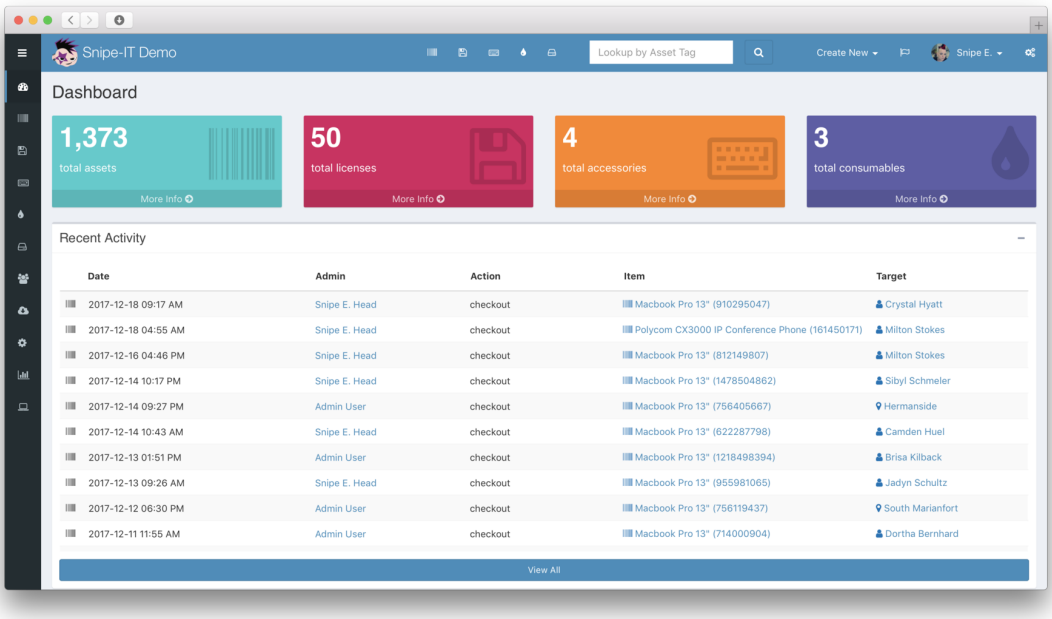

Figure 4.1. Snipe-IT graphical interface

4.2 Lansweeper

Lansweeper is a commercial tool that allows to perform asset discovery and asset inventory in the same platform, without the need of combining different tools [19]. The idea behind this powerful instrument is that all the IT asset data should be collected in just one place under one big IT asset inventory that help the organization or the company that uses this tool to keep track of all the information.

The interface is organized as in figure 4.2 and all the information about the asset are discovered by the tools, that unlike other tools like the aforementioned *Snipe-IT* has an integrated powerful feature of asset discovery that can also be agentless, and therefore does not need any installation of software on the device on which the scan is performed. It is very quick and can reveal a lot of information about the IT assets thanks to the use of a lot of different protocols as for example SNMP, of which there is a brief description in section 2.2. Another interesting feature is the ability to find out more about an asset, for example information about the hardware or the storage of a specific machine, just by giving the credential of that asset. This

characteristic is really impressive since it can be exploited to perform a complete asset management, with all the information that could be considered interesting, and collect everything in a database that can be easily consulted and automatically updated with a scan, that can be even programmed to be performed periodically.

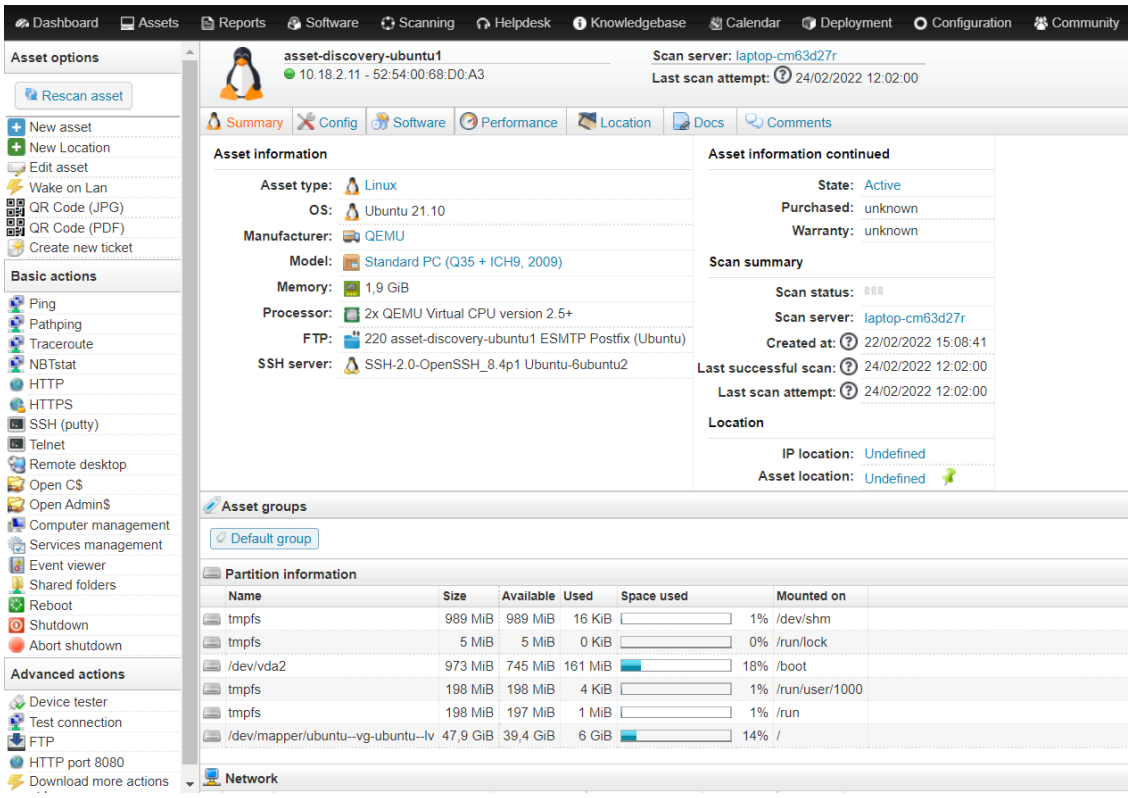

Figure 4.2. Lansweeper graphical interface

The assets discovered by the tool are stored and organized in such a way that are easible to updated and checked at any time. Also the possibility to perform various actions on the devices connected to the network is given, and this is an interesting feature since it can be exploited to control and organize a network by just one endpoint without the need of physically act on the device. Moreover if some more information, in addition to the one discovered by the tool itself, want to be added the process is quick and easy, and this allows to have an inventory that is complete of the data that the user wants to associate to a certain asset.

This solution is potentially very similar to the concept of what this thesis

proposes to do, with the important drawback of not being open-source and so the real functioning is explained just on the surface and there is not an in-depth analysis of the characteristic of the software. In addition to this the cost for the implementation of such a solution, like many other in the asset management field, can be too high for companies that do not have a lot of resources to spend to keep an asset inventory. This has to be anyway looked in perspective also of the fact that a solution of this type allows to save hours of work and this has to be taken into account when a company decide whether to use these kinds of tools or not.

Chapter 5

Tool Development

In this chapter the work done to develop the tool is described, explaining the various approaches and the reasons behind them. Before the actual illustration of the steps taken in the process of building the tool, a brief description of the test environment is given.

5.1 Environment

The idea behind the thesis is creating a testbed that is the most complete possible and that allows to recreate a scenario that eventually a company could face in the process of performing asset management. For this reason the work done in the creation of the environment was toward setting up a heterogeneous set of machines that should simulate a possible configuration of a network in a small company. To do this two solutions were exploited that shared the same common core concept: virtualization.

5.1.1 Oracle VM VirtualBox solution

The first solution was based on the use of a Windows 10 host machine with *Oracle VM VirtualBox*[1] installed and a collection of various virtual machine with different operating systems.

Oracle VM VirtualBox is a free virtualization product that gives a lot of features and possibility when it comes to build and support the execution of virtual machines. With this software it is possible to build an internal

[1] https://www.virtualbox.org/

network between the virtual machines that are being executed thanks to the use of a virtualized network adapter and the feature gave by VirtualBox of creating a virtual NAT networks, with private IP addresses given to the virtual machines connected to it, that is completely independent from the external networks but has access to the internet as shown in figure 5.1.

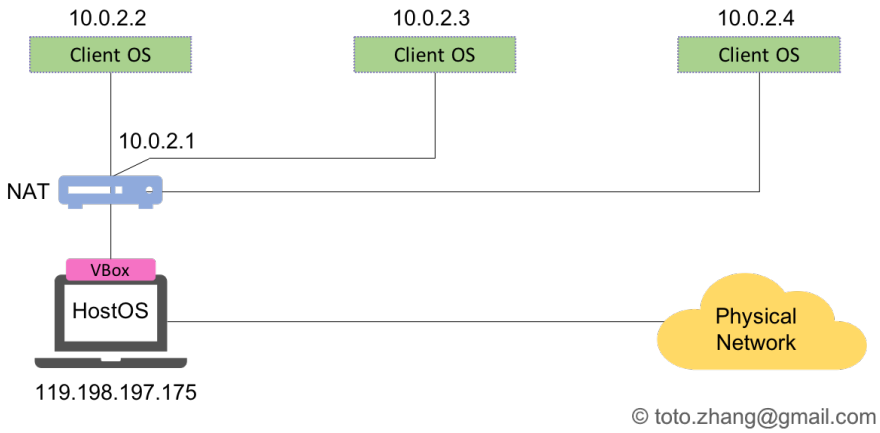

Figure 5.1. Virtual Box NAT network option representation

The working environment was composed of a main virtual machine with *Kali Linux*[2] installed, that had more resources allocated and was the source of all the testing and network asset discovering processes. Kali Linux was chosen because it is an operating system designed for security activity such as network discovering or more in general penetration testing of every kind. The features offered by an operating system as Kali Linux are many and the choice fell on it also because a great manual is offered and the network asset discovery functions are well documented. The rest of the test environment was formed by six other virtual machines with different operating systems that were chosen due to the fact that were reasonably light in terms of the use of resources, and this was important to guarantee that the whole system performance were not diminished, and had different operating systems and configuration running (e.g. open ports, applications installed, etc.) and this lead to a more complete testing environment.

[2] https://www.kali.org/

5.1.2 Remote server virtualization

The idea behind the introduction of this second solution was to create an environment that was even more representative of a possible network infrastructure of a company that needs to perform asset management. In this case the virtual machine used were set up on a server and accessed via a VPN tunnel created with *Wireguard*[3]. The advantage of this type of solution was that there were more resources allocated for the creation of the environment, fact that allowed to work with less stability trouble, and in general this kind of setup was more suitable for the work context that this thesis intended.

The virtual network was composed by six instances of *Ubuntu Server*[4], specifically in its version 21.10, based on Linux Kernel 5.13. As in the previous solution, a virtual machine with more resources allocated was included with the idea of using this main machine to run the tool developed and perform the network asset discovery on the other virtual machine of the network. All the operations were done trough SSH and on the various machines a preliminary work of installing services and application so that these machines could simulate in a realistic way the work of a server.

5.2 Reasons and overview

To better understand the implementation choices it is important to remark the objective behind the work done in this thesis. The aim is the creation of a tool that could facilitate the process of asset management, standardizing it and trying to automatize most steps so that little work and no particular knowledge are required in the use of the tool. All the procedures described in this section are designed with this concept in mind. The final result of the work is an automated script written in bash that exploits Nmap functionality and internally calls another program that is written in Python. The results of the script are then collected and represented through the use of Neo4J, a program that will be presented and described later in this chapter.

In the following sections a more detailed technical description of the work done in this thesis is provided, with a distinction between the part of asset discovery and asset inventory and for each one there is a brief explanation of the various steps taken in the development of the tool that is needed to

[3] https://www.wireguard.com

[4] https://ubuntu.com/server

better understand the workflow followed during the designing phase.

5.3 Asset discovery

The first step to be done in the research for this thesis was to use an asset discovery tool to collect the most information about the asset through a scan of the network. The tool that was chosen to perform this scan, as mentioned before, is Nmap. The choice was dictated by multiple reasons, these certainly include:

- Simplicity of the commands, along with the various option that allow to be faster, stealthier or perform more in-depth analysis depending on the context in which the tool is used;
- Good readability of the output and easy manipulation. In fact Nmap offers several output format including XML, that is very useful when it comes to performing software parsing. An example of a typical Nmap output is provided in figure 5.2;
- Lightweight and portability. Nmap is available on all the main operating system and requires very few resources to run, and this make it a very good tool for the goal pursued in this thesis.

Once that the tool to perform the asset discovery was chosen the next step was deciding which option to use to perform the network scan. Since the aim of the project is to find all the possible information, the choice fell on an aggressive scan that included all the most valuable features offered by NMap scan, that are OS detection, version detection, script scanning, and traceroute. The idea behind this choice is that the person using a tool like the one that this thesis proposes is working specifically to get information from the various machines that compose the network and has no interest in eventually getting detected, and therefore no need to perform a stealthier but also less effective scan.

At this point the network discovery tool and the option are decided and the following phase was the creation of the script that is responsible to run the scan. The bash code is very simple and in fact the only tasks that are performed by the script are:

1. Perform a Nmap scan of the network with the aforementioned option on the target IP specified as an argument when running the script. An

```
Nmap scan report for 192.168.1.172
Host is up (0.00051s latency).
Not shown: 994 closed tcp ports (reset)
PORT     STATE SERVICE
22/tcp   open  ssh
80/tcp   open  http
111/tcp  open  rpcbind
139/tcp  open  netbios-ssn
443/tcp  open  https
1024/tcp open  kdm
MAC Address: 08:00:27:08:C4:E1 (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 2.4.X
OS CPE: cpe:/o:linux:linux_kernel:2.4
OS details: Linux 2.4.9 - 2.4.18 (likely embedded)
Network Distance: 1 hop
```

Figure 5.2. A typical Nmap scan

example of the use of the script and the output prior to the effective scan can be seen in figure 5.3;

2. Store the results of the scan in a XML file;

3. Pass the obtained file to the Python written program that is responsible for the parsing of the output in a CSV file that will be needed in the next phase of the development.

```
┌──(kali㉿kali)-[~/Documenti/scriptNmap]
└─$ sudo ./ScriptBashNMapScan.sh 192.168.1.0/24
RUN THIS SCRIPT AS SUDO USER
Target IP:192.168.1.0/24
Running a quick scan
Starting Nmap 7.92 ( https://nmap.org ) at 2022-03-22 18:26 EDT
```

Figure 5.3. Use of the developed script

It is important to mention the fact that to run the tool it is necessary

to have admin privileges since the Nmap command needs them to run correctly. This was considered and judged not a problem in the designing phase since the in the context of the thesis it is taken for granted that the entity performing the analysis owns admin privileges.

After running the script three new files are obtained:

1. The output of the scan in XML format produced by Nmap, it is quite verbose and contains a lot of information that are not relevant for the aim of this thesis;

2. A second XML file containing the salient data extracted from the Nmap file. These information are: IP (version 4 and version 6 if present) and MAC addresses, open ports and the services that exploits those ports and the OS version if Nmap was able to guess it correctly. These information are divided by hosts and an example of the output can be seen in figure 5.4;

3. The CSV file containing the same information of the a aforementioned file that is going to be used in the asset inventory phase.

At the end of this phase all the work related to network asset discovery is completed and the produced output can then be used in the asset inventory phase that will be described in the next section.

5.4 Asset inventory

The second part of the project developed in this thesis consists in manipulating the data obtained during the asset discovery phase and get a graphical representation. To perform this task there are a lot of viable options. In the designing of this project the choice fell on the use of a software that permitted to obtain a nice graphical representation even if from a practical point of view other possibilities could have been also interesting to explore.

5.4.1 Neo4J

The software that was chosen as the more suitable in the context of the thesis to collect and manage the information about the asset is *Neo4J*[5]. This software is not properly an asset inventory tool. In fact the Neo4J is primary

[5] https://neo4j.com

```
<host5>
  <addresses>
    <address name="192.168.1.172" addrtype="ipv4" />
    <address name="08:00:27:08:C4:E1" addrtype="mac" />
  </addresses>
  <ports>
    <port name="22" state="open" service="ssh" />
    <port name="80" state="open" service="http" />
    <port name="111" state="open" service="rpcbind" />
    <port name="139" state="open" service="netbios-ssn" />
    <port name="443" state="open" service="https" />
    <port name="1024" state="open" service="status" />
  </ports>
  <os>
    <osmatch name="Linux 2.4.9 - 2.4.18 (likely embedded)" />
  </os>
</host5>
```

Figure 5.4. Example of the XML file produced after the parsing

Figure 5.5. *Neo4J Logo*

a Graph Data Platform and offers a lot of features concerning the storing and managing of data.

At the core of the Neo4J Graph Platform there is a database that is done in such a way that the data is connected as it is stored, allowing very powerful queries. This database has to be hosted and the Neo4J platform offers a cloud service that allows to freely use a database for small projects. In addition to this a tool named *Neo4J Desktop* is provided and that allows to create,

work and manage local database. An interesting feature of this software is the possibility to interact with the content of the database using queries written in a language called *Cypher* and obtain a graphical representation of the data retrieved with the query. This last feature is what was exploited in the development of the project for this thesis and in the next section a brief explanation of the work done is provided.

5.4.2 Implementation

In this final stage of development of the tool the starting point are the information collected in the asset discovery phase. With this data the objective is to create a map of the network and for every endpoint add the properties that were discovered in the first stage. To do so a very simple *Cypher* query has been written, as can be seen in figure 5.6 that is responsible to associate an id to every node and then set the various properties if present. The final outcome will be showed as a result in the next chapter of this thesis.

```
//Load with everything as property
LOAD CSV WITH HEADERS FROM 'file:///csvParsedOutput.csv' AS row
MERGE (e:Endpoint {endpointId: row.id, ipv4: coalesce(row.ipv4,NULL)})
SET e.ipv6 = CASE trim(row.ipv6) WHEN "" THEN null ELSE row.ipv6 END
SET e.mac = CASE trim(row.mac) WHEN "" THEN null ELSE row.mac END
SET e.ports = CASE trim(row.ports) WHEN "" THEN null ELSE row.ports END
SET e.services = CASE trim(row.services) WHEN "" THEN null ELSE row.services END
SET e.os = CASE trim(row.os) WHEN "" THEN null ELSE row.os END
WITH e
UNWIND split(row.network, ':') AS endpoint
MATCH (e1:Endpoint {endpointId: endpoint})
MERGE (e)-[r:IS_IN_THE_NETWORK_WITH]-(e1)
```

Figure 5.6. The *Cypher* query that loads the asset

In the final solution the characteristics associated to the various assets are loaded as property, and this means that for every node in the graph there will be various labels that describe it. Moreover the connection are represented as unidirectional arrows in the graph since this is a limit of an implementation done in a software as Neo4J that is not properly intended to create network map. Anyway the important positive aspect of such a solution is its quickness and immediacy, that allow a user that has just performed a scan and then inserted the obtained information in the database to see a graphical representation of the network and the connection between the devices. For

this reason even if no real asset management operations are possible this can be considered as a primitive way of performing asset inventory.

Chapter 6

Results And Discussion

This chapter focuses on the results obtained with the development of this thesis, along with the main characteristics of the solution implemented. In addition an explanation of a particular design choice is provided and a

6.1 Obtained Results

The solution implemented in this thesis has the aim of improving and making the process of asset management easier exploiting the use of already existing tools. The results achieved goes in that direction and here are listed some of the characteristics that the developed tool has and can help in the context in which this thesis took part:

- Simplicity of operation. The code developed in this thesis is designed to be used in context with many machines and devices connected in the same network, but the use of the tool is easy and do not require the reading of too much documentation;
- Automation of the process. A point that was always in mind during the development of this thesis was to try to make a tool that was as much as possible self-working and this is reflected in the final solution that does not need particular work by the user of the tool;
- Graphical representation. The use of the developed script in conjunction with the Neo4J Desktop application and the Cypher query provided in section 5.4.2 allows to have a final product that is easily readable and customizable. An example can be seen in figure 6.1.

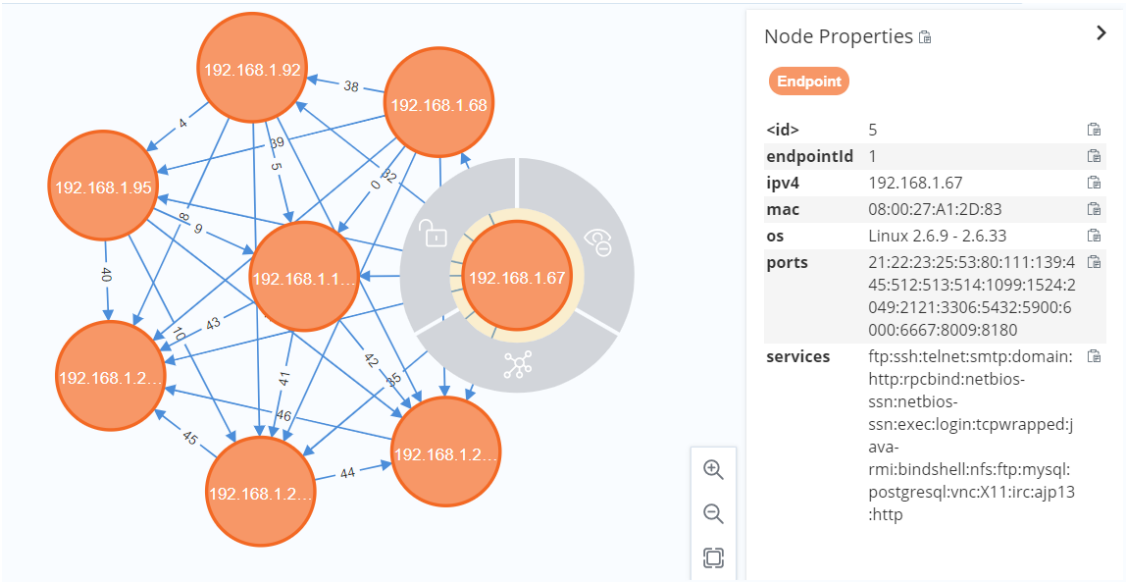

Figure 6.1. The graphical representation obtained in Neo4J

The final result is a very simple tool that gives to the user the possibility to see some of the characteristic of the asset in the network in a very simple way without the need of keeping spreadsheets or other type of documentation to list the assets, that can need extra work and above all the procedure is dangerous since it is difficult to obtain real-time information about an asset in certain contexts.

6.2 Reasons behind the design choices

This type of solution is difficult to be tested and described with a measurable parameter, because there are multiple variables that can influence the execution of the script and the tool as a whole. One parameter which can be taken in consideration for a quantitative measurement of the performance is the time needed to perform the Nmap scan, that is certainly the most time-consuming phase in the implementation of the tool proposed in this thesis. More than a real evaluation of the performance, that as explained before is difficult to obtain even with a certain degree of approximation, in this section a consideration on the type of scan in relation to the time needed to perform it is provided.

The Nmap scan that is performed, as described in section 5.3, is an aggressive and complete network scan. This mean that even for a small network it can possibly become quite time-consuming. In the environment of this thesis, anyway, this is never been a problem since the number of machines that have to be discovered and analyzed in the network was quite small in comparison to the one that a real context could require. But this could reveal to be a drawback in certain environments, so a possible solution is to use a lighter type of scan that may be less precise but faster. In any case the appropriate solution has to be based on the real context, but for this reason is important to know that is possible to still use the tool and just apply some small changes to come to a compromise between performance and reliability.

Chapter 7

Conclusions

To summarize, in this thesis the problem of using asset discovery tools to support a cybersecurity inventory was faced and large part of the contribution was oriented towards a study of the various techniques and tools that are available in this field. The work done showed how this world is vast and there are many commercial solution that try to answer to this problem, but there is not something that can be considered standard and that can be universally used. In fact in most cases every company develops or uses software or tools that can differ in many way from one to another. For this reason it is very difficult to imagine a standard way of performing asset inventory and this becomes a problem in a context like the one of the *PSNC*, in which is requested to the entity to periodically collect and send the list of the assets in their possession.

The work done in this thesis tries to fill this gap with a solution that aspire to be an example of what the research could try to focus on. The developed tool is very basic and is not ready to be used in a more complex context but the crucial part of this work is that it lays the foundation to a more complex study that has to be done in this field that still has a lot to offer and to be discovered. In any case the solution that this thesis proposes is useful to understand the workflow that should be followed in developing an asset management tool and which are the advantages of building a strong and secure cybersecurity inventory.

In conclusion the initial objective of the work can be considered just partially solved since the research and the results that this document offers can be the basis to more projects oriented in this vast field. It is interesting to notice that a initial use of resources aimed at finding a good solution to this problem would then translate in an important reduction of future costs and

above all a more secure final result. For this reason in the next section a list of future improvements and possible new solutions is presented.

7.1 Future work

With the solution proposes by this thesis just the surface of the problem posed in origin was analyzed and there is still plenty room for improvement in this regard. The idea of a tool that allows a company to perform asset management and asset discovery at the same time, exploiting the use of open-source software and protocol is certainly open to new research and projects, and this thesis proposes to be a foundation of the forthcoming work that is still viable on this topic.

Therefore a list of possible improvements and ideas that should be pursued in future research is provided.

- The use of multiple different network asset discovery tools. In this document is explained in detail which are the strengths of Nmap and one above all is the fact that is easy to use and was perfect in the context in which this thesis took place, but there are plenty of other tools that can be used to perform the phase of network asset discovery. For this reason a possible improvement could be trying to make a tool that exploits more than one of them and combine the obtained results to have a final outcome that is the most complete possible.

- Add to the proposed solution a part of asset discovery performed directly on the endpoint. There are a lot of tools that can be used in the different nodes of the network to extract more information from the machine to have a more complete asset inventory that contains information that are impossible to extract just by scanning the network. Another idea could be to exploit the use of the SNMP protocol (see section 2.2 for reference) or in any case an architecture developed in that way, with manager and agents that communicate to extract the data that needs to be collected.

- Improvement on the asset inventory phase. The solution explored in this thesis proposes a small part of cybersecurity inventory, but there are not many functionalities and this can not be seen as a definitive solution. So a possible addition in a future version of the tool is a more complete and rich instrument to perform manage the asset discovered. This tool could be chosen between the one available, or in alternative be developed if the resources allocated permit to do so;

- Integration with the *PSNC* ontology. An aspect that was not covered in this thesis was the part of using the assets discovered with the tool to fill an ontology that could be used by the entities of the *PSNC* to create the list of assets that need to be periodically updated and sent to the authority. This is a characteristic that would be very interesting to implement and could potentially ensure that the appeal of the tool is increased and it can become very important in the *PSNC* dynamics.

Bibliography

[1] Matthew P Barrett et al. «Framework for improving critical infrastructure cybersecurity version 1.1». In: (2018).

[2] PCI Security Standards Council. *PCI DSS Risk Assessment Guidelines 1.0a*. https://www.pcisecuritystandards.org/documents/PCI_DSS_Risk_Assmt_Guidelines_v1.pdf. 2012.

[3] *Decreto del Presidente del consiglio dei ministri 14 aprile 2021, n. 81*. https://www.gazzettaufficiale.it/eli/id/2021/06/11/21G00089/sg. 2021.

[4] *Decreto del Presidente del consiglio dei ministri 15 giugno 2021*. https://www.gazzettaufficiale.it/eli/id/2021/08/19/21A05087/sg. 2021.

[5] *Decreto del Presidente del consiglio dei ministri 30 luglio 2020, n.131*. https://www.gazzettaufficiale.it/eli/id/2020/10/21/20G00150/sg. 2020.

[6] *Decreto del Presidente della Repubblica 5 febbraio 2021, n. 54*. https://www.gazzettaufficiale.it/eli/id/2021/04/23/21G00060/sg. 2021.

[7] *Decreto legislativo 18 maggio 2018, n. 65*. https://www.gazzettaufficiale.it/eli/id/2018/06/09/18G00092/sg. 2018.

[8] *Directive (EU) 2016/1148 of the european parliament and of the council*. https://eur-lex.europa.eu/eli/dir/2016/1148/oj, year=2016.

[9] Jon Erikson. *HACKING the art of exploitation (2nd ed.)* NoStarch Press, 1977, p. 264. ISBN: 1593271441.

[10] *EU Cybersecurity plan to protect open internet and online freedom and opportunity*. https://ec.europa.eu/commission/presscorner/detail/en/IP_13_94, year=2013.

[11] Fyodor. *The Art of Port Scanning*. http://phrack.org/issues/51/11.html. 1997.

[12] Gordon Fyodor Lyon. *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*. Insecure.com LLC, 2009. ISBN: 9780979958717.

[13] Gartner. *IT Asset Management: It's all about process*. https://www.gartner.com/imagesrv/media-products/pdf/provance/provance_issue1.pdf.

[14] Guarnaccia. *Il valore aggiunto dell'asset inventory come elemento centrale di un'azienda*. https://www.ettoreguarnaccia.com/archives/3796. 2016.

[15] Hardaker. *Transport Layer Security (TLS) Transport Model for the Simple Network Management Protocol (SNMP)*. RFC 6353. RFC Editor, July 2011. URL: https://www.rfc-editor.org/rfc/rfc6353.txt.

[16] IT Governance Insitute. *Cobit 4.1 Framework*. https://www.bauer.uh.edu/parks/cobit_4.1.pdf. 2007.

[17] ISO/IEC. *ISO/IEC 27001:2013 Information technology — Security techniques — Information security management systems — Requirements*. https://www.iso.org/isoiec-27001-information-security.html. 2013.

[18] Jones. *An Introduction to Factor Analysis of Information Risk (FAIR)*. http://www.riskmanagementinsight.com/media/docs/FAIR_introduction.pdf. 2006.

[19] *Lansweeper website*. https://www.lansweeper.com.

[20] *Legge 18 novembre 2019, n.133 "Conversione in legge, con modificazioni, del decreto-legge 21 settembre 2019, n. 105, recante disposizioni urgenti in materia di perimetro di sicurezza nazionale cibernetica."* https://www.gazzettaufficiale.it/eli/gu/2019/11/20/272/sg/pdf. 2019.

[21] Schmidt Mauro. *Essential SNMP 1st edition*. O'Reilly, 2001.

[22] NIST. *Security and Privacy Controls for Information Systems and Organizations*. https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r5.pdf. 2020.

[23] *Nmap Host Discovery*. https://nmap.org/book/man-host-discovery.html.

[24] *Nmap OS Discovery*. https://nmap.org/book/osdetect.html.

[25] *Nmap Other Platforms*. https://nmap.org/book/inst-other-platforms.html.

[26] *Nmap Port Scanning*. https://nmap.org/book/man-port-scanning-techniques.html.

[27] *Nmap Scripting Engine*. https://nmap.org/book/nse.html.

[28] *Snipe-IT documentation*. https://snipe-it.readme.io/docs.

[29] *Snipe-IT github*. https://github.com/snipe/snipe-it.

[30] SACHIN Umrao, MANDEEP Kaur, and GOVIND KUMAR Gupta. «Vulnerability assessment and penetration testing». In: *International Journal of Computer & Communication Technology* 3.6-8 (2012), pp. 71–74.