

FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

An OSINT Approach to Automated Asset Discovery and Monitoring

Pedro Daniel Carvalho de Sousa Rodrigues



Master in Informatics and Computing Engineering

Supervisor: Professor João Neves

February 20, 2019

An OSINT Approach to Automated Asset Discovery and Monitoring

Pedro Daniel Carvalho de Sousa Rodrigues

Master in Informatics and Computing Engineering

February 20, 2019

Resumo

O objetivo principal desta tese é apresentar uma proposta para melhorar a eficácia dos *Security Operation Centers* (SOC) através da articulação de diferentes fontes de informação de segurança públicas. O objetivo é desafiante devido aos diferentes modelos de abstração de dados das fontes em questão (como Shodan ou Censys). Estas fontes necessitam de ser compatíveis universalmente por forma a criar impacto nos eventos gerados e na escalabilidade da solução, tendo em conta os dados, recursos computacionais de processamento e de rede na recolha de tais eventos.

Seguindo as normas da indústria propostos na literatura atual - *Offensive Security Certified Professional* (OSCP) guide, *The Penetration Testing Execution Standard* (PTES) e *Open Web Application Security Project* (OWASP) -, a deteção de ativos de rede e subdomínios através da articulação de diferentes fontes é visto como a primeira interação numa avaliação. Esta interação falha frequentemente algumas fontes que poderiam descobrir mais ativos de rede. Isto torna-se pertinente visto que as redes escalaram a sua dimensão para a *Cloud* onde os endereços IP da rede não são pertencentes à organização mas sim à empresa que fornece o serviço de *Cloud* onde vários serviços podem ser partilhados por outras entidades, como é por exemplo o conceito de *Virtual Hosts* para armazenar diversas aplicações no mesmo servidor.

O foco desta dissertação será esse mesmo primeiro paço da avaliação, a enumeração da rede alvo. Atacantes, por regra, usam diversas técnicas para enumerar o alvo de forma a descobrir serviços vulneráveis. Esta enumeração pode ser melhorada pela adição de novas fontes de informação e técnicas que, de forma geral, são deixadas de fora de alguma literatura. Adicionalmente, ao criar um processo automático é possível dotar os SOC na descoberta destes ativos e mapear as aplicações de forma a manter a par de tais vulnerabilidades usando recursos de *Open Source Intelligence* (OSINT) e soluções publicamente disponíveis, antes de tais atacantes terem a oportunidade de as explorar. Isto dota de uma visão sobre a exposição à Internet dos serviços que são vistos pelos atacantes sem questionar diretamente os serviços, evitando assim a deteção de intrusão. Esta investigação está enquadrada num plano de análise completo e deve ser integrada em soluções atualmente já criadas, logo os resultados devem ter a capacidade de alimentarem ferramentas adicionais para explorar o processo de análise usualmente feito.

Ao trabalhar esta problemática esperamos vir ao encontro de administradores de sistema e equipas de segurança, ajudando-as com a tarefa de proteger os seus ativos e assegurar uma "higiene de segurança" transversal a toda a organização resultando assim em melhores políticas de *compliance* sem nunca se contactar o serviço final.

Keywords: *Security and privacy; Vulnerability management; OSINT*

Abstract

The main objective of this thesis is to present a proposal to improve the efficiency of Security Operation Centers (SOC) through the articulation of different publicly open sources of security related feeds. This objective is challenging because of the different abstraction models of the feeds in question (such as Shodan and Censys) that need to be made compatible, with intent to create impact the security events, and the scalability of computational and networking resources that are required to collect those security events.

Following the industry standards proposed by the current literature - Offensive Security Certified Professional (OSCP) guide, The Penetration Testing Execution Standard (PTES), and Open Web Application Security Project (OWASP) -, the detection of hosts and sub-domains using an articulation of several sources is regarded as the first interaction in an engagement. This first interaction often misses some sources that could allow the disclosure of more assets. This became important since networks have scaled up to the cloud, where IP address range is not owned by the company, and important applications are often shared within the same IP, like the example of Virtual Hosts, to host several application in the same server.

The focus of this dissertation is on the first step of any engagement, the enumeration of the target network. Attackers, by norm, use several techniques to enumerate the target to discover vulnerable services. This enumeration could be improved by the addition of several other sources and techniques that are often left aside from the literature. Also, by creating an automated process it is possible for SOCs to discover these assets and map the applications in use to keep track of said vulnerabilities using Open Source Intelligence (OSINT) resources and publicly available solutions, before the attackers have the opportunity to exploit the service. This gives a vision of the Internet facing services often seen by attackers without querying the service directly therefore evading the intrusion detection. This research is in frame with the complete engagement process and should be integrated into already built solutions, therefore the results should be able to connect to additional applications in order to reach forward in the engagement process.

By addressing these challenges we expect to come in great aid of system administrators and security teams, helping them with the task of securing their assets and ensuring security cleanliness of the enterprise resulting in better policy compliance without ever connecting to the destination service.

Keywords: Security and privacy; Vulnerability management; OSINT

Acknowledgements

As with any engineering project, there is no progress without the collaboration and support of other people. Throughout the entire thesis, I should be grateful for my mother support who endure all my "crisis" and bitterness whenever I hit a roadblock.

I also give my greatest gratitude to my friends and colleagues that supported me and insisted on me to finish this project, especially to Diogo Freitas who was there for whenever I needed; Luis Catarino for the technical support and management/testing of the solution giving me a deep insight of the conundrums of managing event logs on a workplace and techniques that I could leverage to get more information about the hosts; António Meireles and Daniel Teixeira with his expertise in advance tactics of defending (and attacking) an enterprise. Furthermore, I like to emphasize the thank you to João Pedro Dias and Tiago Dias for the document revision.

And finally to my advisor professor João Neves, who came to my help in my time of need and believed in me when nobody would. A truly astonishing person who deserves every credit from this document.

Without the help from everyone, I couldn't possibly achieve such a milestone in my life. My biggest thank you.

Pedro Rodrigues

"The world is a dangerous place, (...) not because of those who do evil, but because of those who look on and do nothing."

Elliot Alderson (Rami Malek)

Contents

List of acronyms and abbreviations	xv
1 Introduction	1
1.1 Context	1
1.2 Motivation	2
1.3 Methodology Overview	3
1.4 Research Questions	3
1.5 Goals	4
1.6 Structure of this dissertation	4
2 Literature Review	7
2.1 OSINT feeds	7
2.1.1 Network Discovery	7
2.1.2 Service Discovery	9
2.1.3 Vulnerability Databases	10
2.1.4 SCAP	11
2.1.5 Parsing Feed Data	12
2.2 Flow of Asset Discovery Testing	13
2.3 Incident Response	14
2.3.1 OODA - Observe, Orient, Decide and Act	14
2.3.2 NIST SP 800-61	15
2.3.3 ENISA Incident Management guide	16
2.3.4 ISO/IEC 29147 (Vulnerability Disclosure)	18
2.4 Supporting Application	19
2.4.1 Alienvault USM	20
2.4.2 IntelMQ	21
2.4.3 Choosing the Supporting Platform	22
2.5 Risk Assessment	23
2.5.1 CVSS	23
2.5.2 Network Risk Graphs	26
2.5.3 VULCON	26
2.5.4 Choosing the risk metric	27
3 Solution Proposal	29
3.1 Problem Description	29
3.1.1 Lack of Security awareness	30
3.1.2 Difficulty in detection of Hosts	30
3.1.3 Difficulty in detection of Services	30

3.1.4	Lack of automation of the process	31
3.2	Possible Solutions	31
3.2.1	Tackling the Lack of Security Awareness Problem	33
3.2.2	Tackling the Difficulty in Host Detection	33
3.2.3	Host Port Search APIs	34
3.2.4	SubDomain Search APIs	35
3.2.5	Tackling the Difficulty in the detection of Services	41
3.2.6	Tackling the Lack of Automation of the process	42
3.2.7	Developed application	44
3.2.8	Limitations of the application	52
3.2.9	Integration with IntelMQ	55
4	Use cases and results	59
4.1	Analysis of a University Network	59
4.2	Analysis of a Government Network	63
4.3	Analysis of a Network of Health Institutions	69
5	Conclusions and Future Work	75
5.1	Future Work	76
5.2	Legal	77
A	Raw data results	79
A.1	University Network	79
A.2	Government Network	81
A.3	Health Institutions	84

List of Figures

2.1	Proposed Workflow to discover and evaluate risk	14
2.2	ENISA Incident Resolution	17
2.3	Vulnerability disclosure process summary	19
2.4	USM AlienVault architecture	20
2.5	Bot Architecture IntelMQ	21
2.6	CVSS Metrics and Equations	24
3.1	Model of the data to be stored in the Database	33
3.2	Application developed - Terminal	45
3.3	Application developed - Graphical	48
3.4	Application developed - Graphical - Scan	48
3.5	Application developed - Graphical - Domain Tree	49
3.6	Application developed - Graphical - CPE/CVE correlation	49
3.7	Application developed - Graphical - Names Database	50
3.8	Application developed - Graphical - Services Database	50
3.9	Process Flow of the proposed solution	53
3.10	IntelMQ - Bot configuration	56
3.11	IntelMQ - Dashboard	56
3.12	IntelMQ - Bot Output	57
4.1	DNS Misconfiguration on an University Network	60
4.2	Exposed Development environment on an University Network	61
4.3	Exposed FTP Services on an University Network	61
4.4	Exposed MySQL/MariaDB Services on an University Network	62
4.5	DHT on an University Network	62
4.6	Revoked Certificate on a Government Network	65
4.7	Exposed FTP on a Government Network	65
4.8	DNS Recursion on a Government Network	66
4.9	Exposition of Portmapper on a Government Network	66
4.10	Exposition of MySQL/MariaDB on a Government Network	67
4.11	Apache version in Government Network	68
4.12	Apache CVEs in Government Network	68
4.13	Exposition of Telnet on a network of Health Institutions	70
4.14	Exposition of FTP on a network of Health Institutions	71
4.15	Exposition of PPTP on a network of Health Institutions	72
4.16	IOC on a network of Health Institutions	72

List of Tables

2.1	OODA in incident response	15
2.2	CVSS Qualitative Score	26
4.1	Result Statistics of a University Network	63
4.2	Result Statistics of a Government Network	69
4.3	Detected software on a network of Health Institutions	70
4.4	Result Statistics of a network of Health Institutions	73
A.1	Detected software in a University Network	81
A.2	Detected software in a Government Network	84
A.3	Detected software in a network of Health Institutions	85

List of acronyms and abbreviations

AAA	Authentication Authorization and Accounting
API	Application Programming Interface
C&C	Command and Control
CCE	Common Configuration Enumeration
CERT	Computer Emergency Response Team
CNCS	Centro Nacional de Cibersegurança
COTS	Commercial Off-The-Shelf
CPE	Common Platform Enumeration
CSIRT	Computer Security Incident Response Team
CVE	Common Vulnerabilities and Exposures
CVSS	Common Vulnerability Scoring System
CWE	Common Weakness Enumeration
DHT	Distributed Hash Table
DNS	Domain Name System
DoS	Denial of Service
ENISA	European Union Agency for Network and Information Security Agency
EoL	End of Life
EU	European Union
FISMA	Federal Information Security Modernization Act
FQDN	Fully Qualified Domain Name
FTP	File Transfer Protocol
GDPR	General Data Protection Regulation
HTML	Hypertext Markup Language

HTTP	HyperText Transfer Protocol
IDS	Intrusion Detection System
IIS	Internet Information Service
IOC	Indicator of Compromise
IPS	Intrusion Prevention System
IPv4	Internet Protocol version 4
IPv6	Internet Protocol version 6
ISO	International Organization for Standardization
ISP	Internet Service Provider
IT	Information Technology
JSON	JavaScript Object Notation
LDAP	Lightweight Directory Access Protocol
NIDS	Network Intrusion Detection Systems
NIST	National Institute for Standards and Technology
NSA	National Security Agency
NVD	National Vulnerability Database
NVT	Network Vulnerability Tests
OODA	Observe Orient Decide Act
OSCP	Offensive Security Certified Professional
OSINT	Open Source Intelligence
OTX	Open Threat Exchange
OU	Organizational Unit
OVAL	Open Vulnerability and Assessment Language
OWASP	Open Web Application Security Project
PPTP	Point-To-Point Tunnel Protocol
PTES	Penetration Testing Execution Standard
RCE	Remote Code Execution
RDP	Remote Desktop Protocol
RFC	Request For Comments
RIPE	Réseaux IP Européens

RPC Remote Procedure Call

SCADA Supervisory Control and Data Acquisition

SCAP Security Content Automation Protocol

SIEM Security Information and Event Management

SMB Server Message Block

SNMP Simple Network Management Protocol

SOC Security Operations Center

SP Special Publication

SQL Structured Query Language

SSH Secure Shell

SSL Secure Socket Layer

TLS Transport Layer Security

TSP Trusted Service Provider

TVE Total Vulnerability Exposure

TVR Time-to-Vulnerability Remediation

URL Uniform Resource Locator

USD United States Dollars

USM Unified Security Management

VPN Virtual Private Network

XCCDF Extensible Configuration Checklist Description Format

XML eXtensible Markup Language

XSS Cross-Site Scripting

Chapter 1

Introduction

This dissertation intends to understand how Internet-facing assets are discovered by attackers and build an automated system to detect those assets before they have the opportunity to do so. The normal architecture regarding building a Security Operations Center (SOC) is to have a central logging system where all the event logs will converge, considering a network monitor probe is in place, where it can trigger an alarm on the Network Monitoring System (NMS). This is somewhat useless due to the impossibility to look at the content and the lack of knowledge of what information to look for. Our approach to this problem is to create a system that will retrieve online publicly available information known as Open Source Intelligence, (OSINT) to better detect a breach (as proposed in 8.2 of [1]), reduce the gap between the breach and the detection, and create a risk map where it is possible to analyze the potential for security breaches in the future. As an example it is often seen dumps of passwords online (password reuse on other sites than the corporate one by using the enterprise email) if detected that is indeed a breach and it could give access to the attacker to our systems it is possible to immediately disable the account and notify the user of the action that had been taken. This could help reduce the time to patch the system and increase the security of the enterprise. Furthermore, if a vulnerability could be checked live it is possible to notify the administrator that there could be an exposed asset, possibly at risk, thus helping the enterprise to mitigate the risk. Another enhancement proposed is the risk assessment of the Internet exposed assets of an enterprise. This metric will help to understand what actions should the enterprise take to reduce exposure and mitigate risks and to understand if the enterprise takes security into consideration or if it just exposes services on a demand basis where every service created is immediately put online.

1.1 Context

This dissertation was brought to light in need for better communication within several enterprises. These enterprises had several branches, each one with their services exposed to the Web where the management of those services was problematic (who is responsible, why did not the problem was reported earlier to upper management, and where it was publicly disclosed by

any means). To try to solve this problem a network monitoring probe was put in place to try to catch communications with Command and Control (C&C). However several issues arose such as tunneled communications, either with Secure Sockets Layer/Transport Layer Security (SSL/TLS) or through a Virtual Private Network (VPN). These types of communication are considered challenging since they can not be examined. Since the traffic is encrypted end-to-end, several attacks bypassed the Intrusion Detection System (IDS). If no interaction is made with the service, there is nothing to analyze in scenarios where a compromise was made, but no further interaction is being done, or even where the compromise activity of the compromise is so low that it is hidden by other alarms. Due to this insufficiency, it is needed to monitor the network using different techniques to reduce the time for detection and to improve the detection rate. The problem of monitoring organizations that are out of our control proves to be difficult (providing right to alert administrators but not the right to intercept communications), in the case of the universities, where different Organizational Units (OU) have control of their assets and exposure but the security team is located on the main building and receives no alerts on exposed assets of said OUs. The proposed methodology will provide additional information about these assets and provide the security team with a surface of the whole organization to check for problems along the road such as forgotten assets where patching has stopped, or misconfigured permissions.

1.2 Motivation

It is essential to undertake this problem since security compliance concerns, such as the General Data Protection Regulation (GDPR)[2], are on the rise (often with heavy fines [3]). Besides, the sensitivity of data stored on applications could lead to identity theft and other criminal use of said information.

Criminals like to discredit organizations by releasing this information demanding a ransom to keep them from publishing these findings. Since organizations store a lot of personal information and have a considerable amount of exposed hosts, it is imperative they keep the data of its users protected.

Security compliance is important. Shortcuts for solutions to accelerate the development are often taken but with serious ramifications that are unknown to the developers. There are also times when critical security vulnerabilities are discovered, and the security administrators can not get a quick overlook of their complex network to determine what websites they should patch. As proposed in "An Implementation Model for Open Sources Evaluation" [4] in section 1.1.5 OSINT "Means obtaining intelligence from publicly available sources that are legally and ethically accessed and are available at low cost. (...) exploitable open sources has increased after the Cold War, and the intelligence products started to be based mostly on open sources.". Thus reducing the time that assets remain vulnerable and exposed is an utmost necessity and the main goal that this dissertation tries to achieve. In the mentioned article [4] it was also proposed an architecture of an OSINT platform where a product that gathers information from several sources is prone to help the operator/client to make better decisions towards their objectives.

Data dumps are becoming more common, and the base of the problem is leaving unprotected systems exposed to the Internet. Regarding security practices, if a system is patched and has the basics of security practices, the experience required to hack the system is higher. Therefore a simple attacker who only launches malicious tools and has no knowledge to circumvent necessary mitigations (also known as *Script Kiddie*) would not be able to break it easily.

"Shamming" of enterprises is also gaining some terrain, where security breaches require compensation for the users whom latter know that the enterprise dealing with their data is misusing it. The more recent example is Facebook where a piece of overly excessive information was being given to application developers that will later be used to create a profile and target those people about political campaigns. Although a little out of the scope of this dissertation, this behavior leads some people to drop Facebook regarding their data privacy [5].

1.3 Methodology Overview

The roadmap for this project is as follows: there is a need to understand what procedures are in place to handle incidents. This procedure will be useful to understand at what part of the process can be undertaken to resolve the issue in question; after understanding the security best practices, it will be analyzed what software already exists to perform network monitoring. This will be an overview of different systems to understand what information could better address the problem; next, different OSINT sources and how to use them in our system are going to be analyzed; foremost an attempt to correlate information to join security vulnerability feeds to our map of exposed hosts/domains/sub-domains/net ranges together with open ports/services with the objective to trigger alerts of risk and alert the administrator is going to be made.

The base of this experimentation will have an engineering method such as described in [6]. This method foresees that "Engineers develop and test a solution to a hypothesis", therefore "Based upon the results of the test, they improve the solution until it requires no further improvement."

This method describes that, for each iteration of the solution, tests are going to be designed to verify if the results are expected and reflect on the desired progress.

1.4 Research Questions

The main question is how the final prototype reduces the time that takes to mitigate a possible risk. Assuming the reduction of time, often it is forgotten that by the end of the day people still need to access the systems and patch them. The patch can be simple or complex depending on the architecture of the solutions or the risk that the enterprise assumes, since patching some systems may be economically not viable.

Another concern is that the feeds will not provide an immediate clarification of a point of entry of the attackers, meaning that when a compromise occurs, the attack vector is often unspecified. This lack of information will trigger a further investigation of different logs and artifacts left by the attacker to understand what was the vector exploited.

Some feeds will also not give a clear indication of compromise but indicate that the service is unpatched and poses a risk to the organization. These situations demonstrate when the preventive approach should be adopted, where it should trigger a procedure to notify the organization in order to expedite a patch quickly.

1.5 Goals

This dissertation proposes a new approach to passive security monitoring. With the addition of new feeds to detect hidden and forgotten hosts, it is expected to discover hosts that lack security updates or are running outdated, unsupported and End of Life (EoL) applications that could be an entry point in the organization.

Often attackers, in the first phase of the engagement, look for "low-hanging fruit" which means that they look for systems they know vulnerabilities to try to penetrate the network perimeter. This action, however, is not always performed by defenders and SOC operators and in a big enterprise with dynamic creation of services could lead to confusion. This lack of inventory will cause some services to be forgotten and kept outdated. This approach, where a continuous search for new hosts is made and then correlating with known vulnerabilities, could be a step forward in a preventive action against an attacker. By detecting vulnerable hosts in the target companies without interaction, it is possible to have a visibility of what the enterprise exposes to the Internet, and infer the risk of it. This ability could be valuable to a SOC since active scans are noisy and often are left out in detriment of attack patterns, meaning that only when people attack the service, the SOC team can detect the incident which, at that point, could be too late.

The main objectives of this dissertation are:

1. Reduce the time taken to handle an incident;
2. Proactively protect the assets of an enterprise by discovering forgotten resources;
3. Create a database of exposed assets to have an historical view of the evolution of the exposure;
4. Provide information to other assessment tools to better determine the risk of the exposure.

1.6 Structure of this dissertation

In addition to this introduction, this dissertation contains four chapters describing the progress done.

In the *Introduction* a brief introduction to the problem and its context is given, how it came to light and what is the target problem.

In chapter 2, *Literature Review*, a presentation of the already established research that could aid us in the approach of the problem such as already established solutions for Intrusion Detection System/Intrusion Prevention System (IDS/IPS) and analysis of their architecture. It

will be analyzed some of OSINT feeds and incident response procedures from National Institute of Standards and Technology (NIST) and European Union Agency for Network and Information Security (ENISA). Then an overview of some established solutions for incident and event management is going to be made. Finally, a look at the Risk Assessment Metrics is made to possibly classify the risk for the enterprise.

In chapter 3, *Solution Proposal*, the problem is presented with a possible solution to solve or at least mitigate it. A possible implementation as well as some key components of its dependencies are going to be discussed in this chapter.

In chapter 4, *Use cases and results*, will be presented using three different networks. These networks will be analyzed by vulnerabilities and misconfigurations. This evaluation will help to understand how much the developed solution can contribute to mitigate the problem.

In the last chapter, Chapter 5 *Conclusions and Future Work*, an observation on the developed work will be made as well as pointing some constraints and the viability of the solution. Future improvements to the solution are also considered in this chapter.

Chapter 2

Literature Review

In this chapter it will be discussed different topics regarding OSINT feeds, supporting database, application, and security reports.

As stated in the "ShoVAT: Shodan-based vulnerability assessment tool for Internet-facing services" [7], it is possible to use Shodan [8] to detect vulnerable hosts exposed to the Internet and enumerate some of the services and versions using the Common Platform Enumeration (CPE) [9] enumeration standard. Here an introduction of the concept where all the feeds will be interlaced and a risk map of the enterprises will be presented. Although the authors of ShoVAT only use Shodan, it is possible to use other sources to enumerate virtual hosts [10] that could be behind an IP (Internet Protocol) address, for example a vulnerable application can not be easily accessed without knowing the Fully Qualified Domain Name (FQDN). Furthermore, this method is used in an attack perspective, but it could also be used in a defensive stand in hope to detect services that are not in the service inventory of the enterprise, or it could be used to monitor the growth of the network of some companies.

2.1 OSINT feeds

To support this project, the collaboration of several types of feeds are going to be needed. The feeds will correspond to the following categories:

- Network Discovery - Discovery of exposed assets;
- Service Enumeration - Discovery of the services running on exposed assets;
- Vulnerability Databases - Database of security vulnerabilities regarding running service version.

2.1.1 Network Discovery

In this section several feeds are going to be presented that might provide useful information of the exposed surface of the enterprise:

Shodan - Shodan is the leading online scanner, it scans the entire IPv4 and IPv6 address range and provides an Application Programming Interface (API) that indicates if the host already has a service with a critical flaw. It also stores information regarding the hostname if available.

Censys [11] - This is another online scanner that provides a well define API. Although it scans the same Internet Protocol version 4 and Internet Protocol version 6 (IPv4 and IPv6) address range, it does not include services vulnerability check. This is required to extract information from the header if possible.

RIPE database [12] - Réseaux IP Européens database. This database allows to obtain registered IP address ranges. This feed is important since companies often register different IP address ranges that are not usually associated with hostnames but hold critical services.

HackerTarget HostDiscovery [13] - This online tool helps to discover hostnames from the domain intended to search. Considering that an enterprise registers a domain (or several), this feed keeps a collection of hosts they have on that domain. In the example: University of Porto registers the *up.pt* domain, consequently each OU registers a sub-domain (i.e.: *fe.up.pt* and finally the OU registers an hostname (i.e.: *www.fe.up.pt*). If this information is reachable, the feed will store it.

Bing API [14] - Another great source of exposed hosts are search engines. Bing provides an API where a search for hosts within a domain can be performed. It crawls Web pages that can later be analysed in search for new hosts.

Google API [15] - Google also has an API that can be leveraged to discover even more sub-domains and hosts. Using the power of Google crawling it is possible to detect obscure exposed assets.

ThreatCrowd [16] - ThreatCrowd is a search engine for threats actors, compromised or malicious hosts. However, it provides access to a database with domain names that can be harvested to complete the already obtained results.

Netcraft [17] - This feed is a similar search engine for assets, although it fell behind time. It is referred here for completeness.

Certificate Transparency [18] - To mitigate flaws in the SSL/TLS protocol, this project was created to ensure that Certificate Authorities (CA) do not issue certificates wrongfully. In its essence, a public audit of certificates is done to ensure that the certificates are not misused. However, since these certificates and the records are public, it is possible to consult them and inspect all the certificates with a domain associated, therefore discovering new assets.

ssltools [19] - This website provides an API where it explores the SSL Subject Alternative Name (SAN) where a certificate is issued containing several names. By looping through those records, it is possible to discover other assets.

These feeds will be online but could have a paid subscription. To note that these feeds are considered a third party and therefore no direct interaction with the target host is made at this point. This means that no active check for vulnerabilities is made, but rely on services that do that.

2.1.2 Service Discovery

The discovery of new services comes with a price. Usually, a query to the port via banner grabbing is done, and possibly some strings are sent to get the headers of the protocol to extract more information. For example, the Secure SHell (SSH) [20] service when connected announces its version without interaction. However, some services need an initial string or action to announce themselves. These services could prove challenging to discover if a policy of "no interaction" with the service is going to be maintained. The services can either be found by performing an additional check on the port and invalidate the passive OSINT approach in exchange to better improve the results or use the ShoVAT approach and analyze the already discovered information by Shodan and others to determine what service and version are running on the port.

"Towards Automated Vulnerability Scanning of Network Servers" [21] proposes a technique called fuzzing to try to get a response from the specific port. Fuzzing is a technique where "garbage" (pseudo-random characters) is sent to the port in hope to detect either a security vulnerability or to trigger a response from the service, even an error response that could lead to the disclosure of the service. Some of these scans are done with the ZMap project [22], a project that aims to map the entire Internet. These datasets are later available to download or to consult using the Censys [11] project. These scans are done using the default ports, meaning that if an HyperText Transfer Protocol (HTTP) over TLS service is running on port 80, it is probable that the enumeration will not work correctly.

With this in mind, a consideration of some services to be overlooked should be taken into account. This behavior will be observed in case the feed does not query that specific port. An example would be Information Technology (IT) administrators that expose service in non-standard ports to try to hide it from unknown attacks and expose different applications to the Web. This procedure is a known risk of service discovery through OSINT, and this is not a replacement for penetration testing.

For this step of the process the following feeds were considered:

Shodan - Shodan was already described in this dissertation. It searches for a subset of open ports and is actively developed to include more and improve detection of services. It provides accurate results and a compliant CPE identification of the detected services. Its API allows for one request per second to enumerate a host but allows for subnets to be passed as arguments to improve the efficiency of the search.

ZoomEye [23] - ZoomEye was one of the public accessible known rivals of Shodan. It is an Asian service that performs the same tasks over a higher number of ports. However, it lacks on the service detection site and does not provide a standard CPE for the service detected. It is also limited as stated in this dissertation. Since it is Asia based, it takes several seconds

to retrieve the results, delaying the pipeline. Its Web interface is in Chinese, but automatic translators can process the page. API is in English.

FOFA [24] - This feed is a new competitor of ZoomEye, and it is very similar to it. It is also Asia based however the API is in English. Due to time limits, it was not implemented but should be referred for future integration.

In summary, the service discovery will be made by the feeds that perform banner grabbing and make available that information to correlate with the CPE database (discussed in the next sub-chapter 2.1.3). With this, contacting the target is avoided and therefore there is no chance to create a denial of service or IDS alert.

2.1.3 Vulnerability Databases

Complementing the process, several databases are needed where information about vulnerable software versions can be obtained with the objective of creating alerts to system administrators and warn them about the danger. There are several databases that store the needed information. These databases are public, although some are subscription-based. There are also some vulnerability markets (exploit market). These exploit markets offer several vulnerabilities that are not publicly known and are essential to understanding if the software is vulnerable. For this dissertation, public available vulnerability databases are going to be rely on, and these exploit markets are going to be disregarded. Next are some databases that could be used to retrieve this information:

NVT [25] - Network Vulnerability Tests provided by Greenbone (company that maintains the OpenVAS solution). It contains several networks vulnerabilities signatures to test and assess if the service is vulnerable. This, however, could require to send information to the service and could result in unexpected behavior. This also breaks the passive OSINT approach.

SCAP - Security Content Automation Protocol is a protocol intended to "specify standards to enable automated vulnerability management, measurement, and policy compliance evaluation" [26]. SCAP is used to evaluate security compliance with the help of the following security feeds/standards to understand if the software is outdated or vulnerable (CVE; CCE; CPE; CVSS; XCCDF; OVAL).

CERT [27] - In this part of the dissertation and referring to the security feed, CERT refers to the US-CERT, United States-Computer Emergency Readiness Team. They provide a feed of security alerts. However, they are, and without text mining, it would be difficult to correlate with a piece of more technical information to automate the process.

CVE [28] - Common Vulnerabilities and Exposures, provided by MITRE CVEs are a collection of vulnerabilities publicly disclosed, although it could lack the ability to be automated since it is text-based, similar to the CERT feed.

CCE [29] - Common Configuration Enumeration is a database provided by the NIST that ties the configuration to the service running, so it is easy to enumerate all the configuration files associated with a service. Since no access to the target machine is provided, it would not be used in this dissertation.

CPE [30] - Common Platform Enumeration is a feed that helps to enumerate the underlining technologies. This feed is helpful to understand if the technology is outdated or even end-of-life. It could help find CVEs according to the versions and technology of the solution found, thus helping determine the risk.

XCCDF [31] - Extensible Configuration Checklist Description Format is a database where it is possible to enumerate configurations within a service in hope to detect misconfigured services. Due to the necessity for the configuration file, it would not be possible to parse it using OSINT methods.

OVAL - Open Vulnerability and Assessment Language describes as: "The OVAL Language standardizes the three main steps of the assessment process: representing configuration information of systems for testing; analyzing the system for the presence of the specified machine state (i.e., vulnerability, configuration, patch state); and reporting the results of this assessment." [26]. It is associated with active scanning thus not being considered.

NVD - National Vulnerability Database is a database of vulnerabilities available online and supported by NIST. It contains a list of vulnerable services/versions following a certain structure. [32].

CWE [33] - Common Weakness Enumeration database is a database that tries to list the weakness in software, mainly configuration. Considering the unability to access configurations this standard will not be used.

vFeed [34] - VFeed is a list compatible with the CVE, CWE and OVAL standards. It correlates those three and more databases to create a risk map of the vulnerability in question. However, it does not contains a CPE reference.

Other exploit databases are available, but they are not standardized and depend on the submitter of the vulnerability.

2.1.4 Security Content Automation Protocol

"The Security Content Automation Protocol" (SCAP) [35] is a "method for using specific standards to enable automated vulnerability management, measurement, and policy compliance evaluation" (e.g., Federal Information Security Management Act - FISMA - compliance [36]). More specifically, SCAP is a suite of selected open standards that enumerate software flaws, security-related configuration issues, and product names; measure systems to determine the presence of vulnerabilities and provide mechanisms to rank (score) the results of these

measurements to evaluate the impact of the discovered security issues. SCAP defines how these standards are combined. The NVD provides a repository and data feeds of content that utilize the SCAP standards.

SCAP could be used to gather information about all the vulnerabilities and try to classify them. This could help later to understand the degree of risk an asset imposes to the enterprise. This standard is comprised with the following standards [37]:

CVE, CCE, CPE, XCCDF, OVAL - Already covered in the previous section

CVSS - Common Vulnerability Scoring System [38] is a standard to classify vulnerabilities according to the risk it poses to the system. It is a numeric score from 0.0 to 10.0. This standard will be analyzed in 2.5.1.

2.1.5 Parsing Feed Data

Parsing feeds can be difficult on some applications. The deturgency of technologies has a severe impact on how to extract data from a website.

Commonly two ways are going to be used through this dissertation: Web scraping and API access.

2.1.5.1 Web Scraping

This technique is used in several applications where an API is not present. The central concept is to look at the structure of the Web page and find elements with identification values (ID) to select the desired information to extract. An ID in this context can either be the attribute ID of the Hypertext Markup Language (HTML) syntax or any other attribute that can uniquely identify the data (some frameworks assign IDs differently). The data can be retrieved by combining several techniques, for example, the use of Cascading Style Sheets (CSS) tags to find a specific table that contains the required information and then iterate over it to obtain each cell.

In Python some libraries help the iteration of such data structures. One of the most known is BeautifulSoup [39]. This library contains procedures to obtain tags, attributes, and values effortlessly.

Another known library is lxml [40]. This library is used in the known Recon-ng [41] application that use this library to retrieve information from the resource. However, developers often opt for BeautifulSoup since it is more versatile to develop the application. However, lxml is lighter resource wise.

For this concept, BeautifulSoup was selected for its simplicity to develop the concept quickly.

2.1.5.2 API Access

API is a generic term adopted in this section. In this context, API refers to an interface that was specifically developed to allow other applications to access the data without style or visual

effects on the page. These APIs usually tend to have some structure: JavaScript Object Notation (JSON) or eXtensible Markup Language (XML), even though others may exist.

JSON is the most common one observed, and it is supported natively by Python. In the Python core, parsing a JSON structure is effortless. It is similar to the same way an array is accessed. In a computer science way, this kind of structure is less resource intensive to parse and allow for huge data structures.

XML is an "old" structure. In some sense it is related with HTML, meaning the structure relies on the concept of tags and each tag may have attributes or content. This structure is a lot more resource intensive to parse than JSON, and it is not natively supported by Python.

For this concept, JSON was selected if there is an option due to its simplicity and ease of use in the Python programming language.

2.2 Flow of Asset Discovery Testing

During an engagement, an auditor typically uses search engines (similar to Google search) and brute-force on domain names to try to uncover hosts and services on the targeted enterprise. Other information sources could improve this search, but it is often a manual process where the auditor needs to select each tool for the process: (i.e., browser for Google, scripts for Domain Name System (DNS) brute-force). This process could be improved by automation and using OSINT sources. It can also be useful in a defensive strategy where monitoring third-party networks is required. By consulting those OSINT services, a distance from the network is maintained (no interaction with the target assets) and therefore avoiding brute-force of domain names that could cause network or service instability.

In figure 2.1 a description of the proposed process for asset discovery is shown. In essence, threads are started for each of the OSINT sources described earlier. Then, information of the domain name is stored. Due to some errors that could arise at the end (wrong parsing of results as an example), checking with a regular expression (Regex [42]) if the domain name follows the specification is necessary. IP addresses associated with the domain (in case of a DNS load balancing is in place and to get an IP range of assets) are also stored. It is crucial that both of this data is stored since an IP address could have several domain names pointed to it and a domain name could have a rotating IP address to offer load balancing for users.

With the data already described, entry points for the organization are now available. However, services listening on these addresses need to be enumerated in order to understand if they pose a risk to the organization. Therefore, the next step is to query the subset of feeds providing the service enumeration. After a list of open ports is provided and, if detected, a list of services running is made available, it is possible to detect the running service by analyzing the banner of said service. Some protocols are easy to enumerate since, when contacted, they state the service running and the version of said service (like SSH or Supervisory Control and Data Acquisition (SCADA) [43] services). If this is the case, this information should be stored on the database. Other services do not offer a response and need some interaction. Luckily the feeds try to detect it

without further interaction and store said information conveniently (through the use of CPE fields in their results). Otherwise, the port will be marked "Open" without any information, for further analyze.

With the dataset now populated with domain names, IP addresses, and services associated with those names, a map where the services are, and what services are running, can be made. This information will be helpful further down the pipeline.

Meanwhile, a separated thread will spawn a worker that will create a process for the vulnerability feed. Then a match is made with the vulnerability feed to try to discover vulnerable services. If an occurrence is found (service running and vulnerability on the database) an alert should be issued to the administrator.

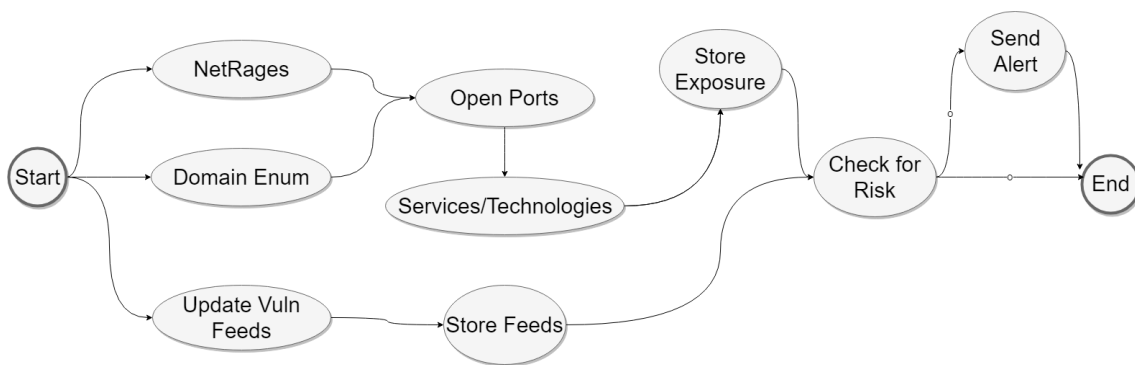


Figure 2.1 Proposed Workflow to discover and evaluate risk

2.3 Incident Response

In a security operation center, operators follow a determined procedure to assess and escalate the situation. Most procedures use three high-level stages: Detection, Analysis, Remediation. The most used incident response guides on the industry are defined by NIST, Alienvault and ENISA Incident Response (CERT). Since Alienvault is based on NIST, it will not be discussed in this dissertation in detail. NIST is based on American legislation while ENISA/CERT is based on European guidelines.

2.3.1 OODA - Observe, Orient, Decide and Act

Observe, Orient, Decide and Act (OODA) is a military acronym designed to impose a procedure of problem approach. It is based on the concept of state machines, where a state can move to a subset of other states from the beginning to the end. When the last state finishes, lessons are learned to be used on the next problem or in similar situations. This concept is useful in incident response since it can transpose the same concepts on how to process with the new information obtained.

Assuming receiving a new piece of information: The first step is to observe if the information is perceivable (it follows a certain pattern, for example, a domain name should have letters and

some characters by a certain order to be understood as such) and not a random piece of data (integrity check), also it should verify what source provided that information (trustiness).

On the deciding stage, a decision on if the information already exists or it was already reported and if the information is critical or merely informative, is made. If the need arises, an alert should be issued on the system to the operator.

On the "Blue Team Handbook: Incident Response Edition" [44], Don Murdoch states each phase as follows: "When engaging an enemy, try to ensure that you are not always reacting. Pause, analyze, incorporate information from the battlefield, and then integrate new knowledge into the next course of action". This procedure is the one that prevails across this dissertation. Enumerating the steps on that reference:

State	Description
Observe	Gather data and raw information from relevant sources, because decisions are made as a situation evolves.
Orient	Separate low value or useless data from valuable data. Organize data into information following rules, presets, and filters.
Decide	Taking action based on the current situation.
Action (Act)	Following through on the decision, choosing the best tools in question to mitigate or resolve the situation.

Table 2.1 OODA in incident response

2.3.2 NIST SP 800-61

According to NIST Special Publication (SP) 800-61 [45], when an incident occurs the response team follows six steps:

1. Preparation - Be ready for an incident, have the right tools to monitor connections, check logs and correlate information. Identify responsibilities (who is responsible for each system on the enterprise) to quickly escalate the situation if the need arises.
2. Detection and Analysis - Detect an incident through the ingest of stream data (network logs, application logs, system events similar to Windows Event Viewer or Syslog). The procedure states that an operator should be allocated to the incident in order to understand the point of entry and what attack vector was exploited.
3. Containment - After the incident has been identified the operator should escalate the situation to contain it. This should be done according to the incident in the way that, for example, if it is a Denial of Service (DoS) attack the originating IP addresses should be blocked by the Internet Service Provider (ISP). This is the first reaction to take and only mitigates the risk for the organization.

4. Eradication - This stage is where the full patch is applied and the problem is fixed, if possible. The system should be checked for Advanced Persistent Threats (APT) or backdoors that could give back access to attackers. If the system has sub-systems (if part of a domain group or connects to databases) they also should be investigated to see if they were targeted and had backdoors installed.
5. Recovery - In this stage, a recheck of all systems is made to assess if the system continues to function properly and if the patch was done correctly. All operations should resume normal operation status.
6. Lessons Learned - Create documents reporting what went wrong (often known as Postmortem [46]).

This philosophy should be intrinsic in every operator and taken for each incident. However, this procedure could not always be met if other problems arise (i.e., patching an EoL firewall with proprietary code). If it could not be fixed, then the risk should be mitigated.

2.3.3 ENISA Incident Management guide

ENISA technical guideline [47] justifies their procedure by sustaining the role of core values. Due to the increase in e-Governance [48] within the European Union (EU), the technical guidelines were issued with the intention to create "Guidelines for trust services providers" (TSP). This guideline is composed of three areas:

Security framework : describing the framework surrounding Trust Service Providers (TSPs), focusing on EU standards, but taking into account others where relevant.

Risk assessment : discussing the principles and concepts of managing the risks applicable to TSPs by defining and controlling threats and vulnerabilities.

Mitigating the impact of security incidents : recommending measures to mitigate the impact of security incidents on TSP by proposing suitable technical and organizational means to handle the security risks posed to the TSP.

Regarding the **risk assessment**, the procedure is subdivided into the following items:

- Assets: identification, classification, and evaluation
- Threats to assets: classification and evaluation
- Vulnerabilities present in the environment
- Probability or frequency of the threat
- The impact that the exposure can have on the organization

- Countermeasures that can reduce the impact
- The residual risk, risk acceptance, risk treatment plan

These guidelines are divided into four parts, as stated in [47]: "The incident handling process has many phases. It describes the sequence of steps that begin when an incident reaches your team.". The stages that the process could take are as follows:

Incident report - There is a report of an incident received by CSIRT or similar.

Registration - The incident is registered on the management system

Triage - The incident then proceeds to triage to classify the threat and verify the trustworthiness of the incident.

Incident resolution - After classifying the incident and prioritizing it according to the risk posing to the infrastructure of the enterprise, it is time to resolve it.

Reaching the last phase, a loop similar to other procedures already discussed in this dissertation can be observed. The loop can be easily observed in figure 2.2. It is similar to the already seen OODA standard.

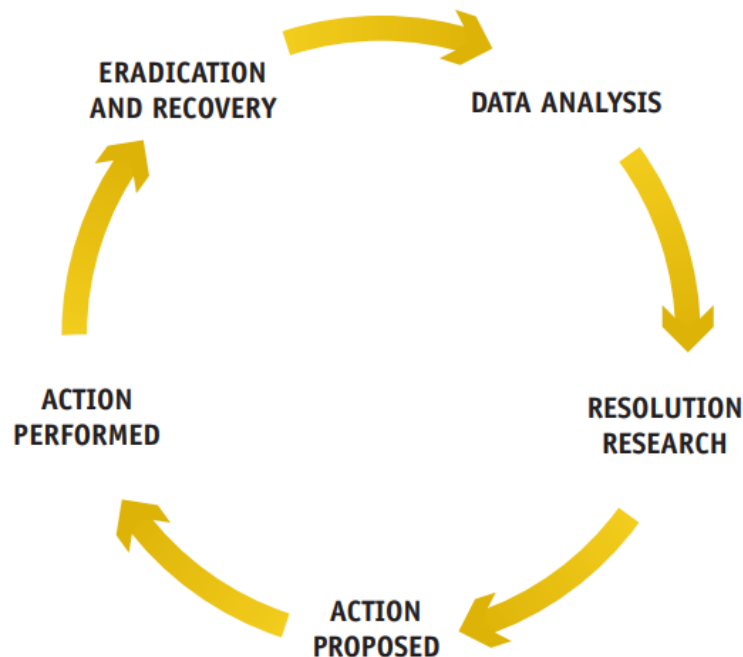


Figure 2.2 ENISA Incident Resolution

Data Analysis is the phase where the incident is analyzed and processed to set its priority. In this phase, the affected parties need to be notified to exchange information. Several entities could be analyzed, for example: incident reporter (if available and willing to participate); Monitoring systems (i.e., systems that log activity, such as, syslog, data flows, related IP source);

In **Resolution Research** an operator first searches a database of incidents (check if the report was already in the management database or is a similar *modus operandi* to an already known attack), and other relevant sources that will contribute valuable information to the operator and aid him in understanding the incident. This phase is equivalent to the "Observe" phase in OODA.

Action Proposed is the phase where solutions, if available, are presented. The solutions come from the resolution research done. It is similar to the "**Decide**" phase in OODA.

The next phase is the implementation of the proposed solution, also called "**Action performed**". In this phase, the solution is put into practice. The solution could be to mitigate/fix the risk or to collect additional metrics to understand how the attack is being carried out, inform others CSIRTs to protect in case of similar incidents. It is similar to the "Act" phase in OODA.

The last phase will be the "**Eradication and recovery**". In this phase the incident needs to be fixed and eradicated, the risk should be no more, and the normal operation of the service should be restored. A final analysis should be done to ensure that no other service is affected or has malware installed that could potentially trigger in the future.

In retrospective, this standard is very similar to the already described NIST SP 800-61 standard.

2.3.4 ISO/IEC 29147 (Vulnerability Disclosure)

This International Organization for Standardization (ISO) [49] document is not targeting incident response, but it defines how the interaction between a researcher (finder) and a vendor should occur with and without coordinators, such as CSIRTs or equivalent. It was adequate to include the reference to this ISO since it would be used when reporting any vulnerability that would be found during the development of this dissertation.

The ISO indicates, on the 5.5 section, how an incident should be handled (considering the incident as a vulnerability disclosure).

It starts with the reception of the disclosure. In this stage a finder tries to communicate with the vendor or the enterprise in question (in case of custom software) by any means that he specifies (usually email, however other means may include telephone and or fax or even a properly located form for security purposes). Secure communications are recommended since the vulnerability could be used for damaging the enterprise in question. The source should also have a cryptographic key to prove that it is indeed the person that claims to be and not an attacker impersonating someone. A coordinator can be contacted to help with the incident. After the initial contact has been made the process changes to the next stage, Verification.

In the Verification stage, the vendor verifies the identity of the finder and his claims. If there are grounds that it is indeed a vulnerability the vendor acknowledges the submission and gives a status on the resolution to the finder. If it does not consider to be a vulnerability, the finder could agree and desist, or it can query other parties like a CSIRT to verify and coordinate a disclosure. If the vulnerability is still not considered as such the finder can publish the finding on their own will and risk.

The following stage (Resolution Development), the vendor, takes the recommendations given by the finder and checks for their practicality and effect. If the vulnerability is being exploited at the moment of disclosure and no immediate resolution is in question, the vendor should create an advisory to their clients in order to mitigate the risk for their infrastructure.

After testing the fix and deploy to the clients the vendor should check if the patch does not break functionality and passes all the tests of the application.

In the Post Release phase, if there are still problems to be addressed, it can fall back to the Resolution Deployment phase or create an advisory, update the CVE (if there is one) and close the incident.

Figure 2.3 shows the flow of the process.

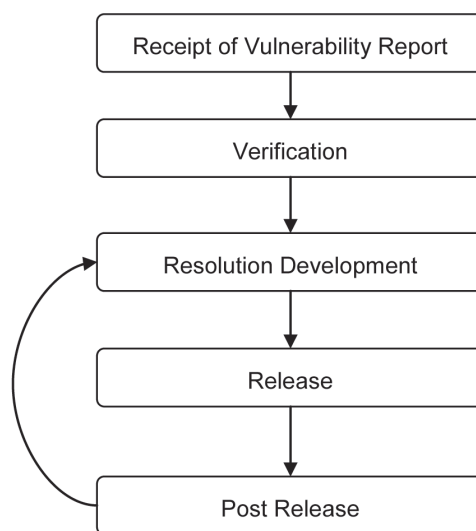


Figure 2.3 Vulnerability disclosure process summary

Using this process, the finder can state his intentions to publish the incident to the vendor. If it is product-related such as a Commercial Of The Shelf (COTS) product, a CVE ID can be issued. In this dissertation, every vulnerability that was not addressed in 60 calendar days will be considered to be unpatched. Less than that it will be considered "On Hold" and the details would be private to the finder and the vendor/enterprise only.

2.4 Supporting Application

Creating an application from scratch to address these issues will be useless. Therefore an application that would be easy to deploy, already in use (to bring additional features to existing SOC) and flexible should be chosen.

2.4.1 Alienvault USM

Alienvault provides an "all-in-one" solution to gather logs and correlate information about the network [50]. In their words: "AlienVault Unified Security Management (USM) delivers powerful threat detection, incident response, and compliance management across cloud, on-premises, and hybrid environments". This means that this platform combines network monitoring (either with Snort [51] or Suricata [52]) with a vulnerability scanner (in this case OpenVAS [53]) and an external feed. Confirmed incidents are reported to the Open Threat Exchange (OTX) [54]. OTX is a feed, a threat incident advisory, with shared resources in the sense that, where an Indicator of Compromise (IOC) is detected in one instance of the world, it will be replicated for the rest of the world. Having a reactive effect in the detection of newer incidents and trying to reduce the response time. IOCs include IP addresses, domains, subdomains, emails, Uniform Resource Locators (URL), file hashes, Mutual Exclusive object (MUTEX) name, CVE ID number.

Although having access to the source code for these solutions several problems arise with this platform:

1. Having access to the source code, there is no permission to modify it freely due to license issues;
2. There is only access to the OTX feed; there are not options to have multiple feeds;
3. It is heavy on hardware requirements (at least 8Gb of RAM and sufficient space for the logging of network packets);
4. Is based on network captures and direct scans of services to trigger events. Therefore is a reactive approach; even the vulnerability scans need to be set with a range or domain names to run and in the enterprise.

An abstract architecture of this solution is easily understood in figure 2.4[50]. It can be observed how the OTX feed and the Public/Private Cloud sends information to USM for correlation.

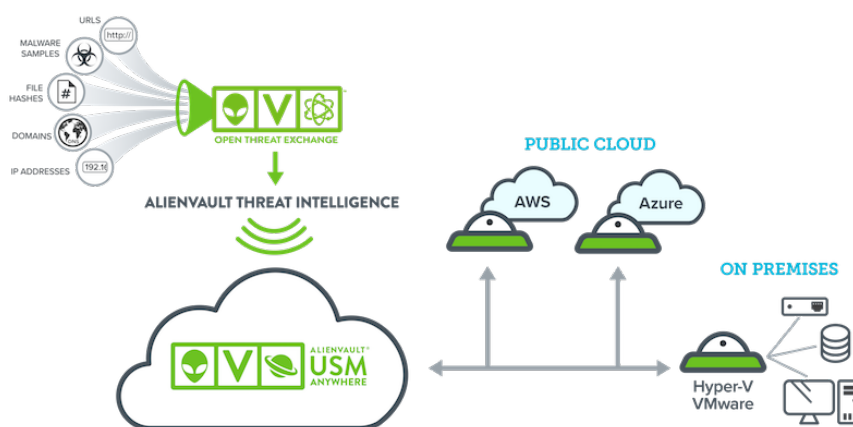


Figure 2.4 USM AlienVault architecture

2.4.2 IntelMQ

This application is "a solution for IT security teams for collecting and processing security feeds using a message queuing protocol." [55] This application is used by ENISA Incident Handling Automation [56].

This application is written in Python and uses "bots" (short for robots) to retrieve information periodically. Also, the pipeline building process is already established, only needing to create a manager to initialize and launch bots. This "already defined" base structure of the application helps developers build on top of the already existing methods. Bots are the "workers" of the application. This means that when they need to retrieve information a bot is created to manage communications with the manager. There are four types of bots in this application:

Collectors Bots - Collect information from sources;

Parser Bots - Parse the information from sources;

Expert Bots - Analyze the information from sources and enrich the information retrieved by the parser bots;

Output Bots - Outputs all the operations results.

In figure 2.5 a diagram of how IntelMQ Bots are organized can be observed.

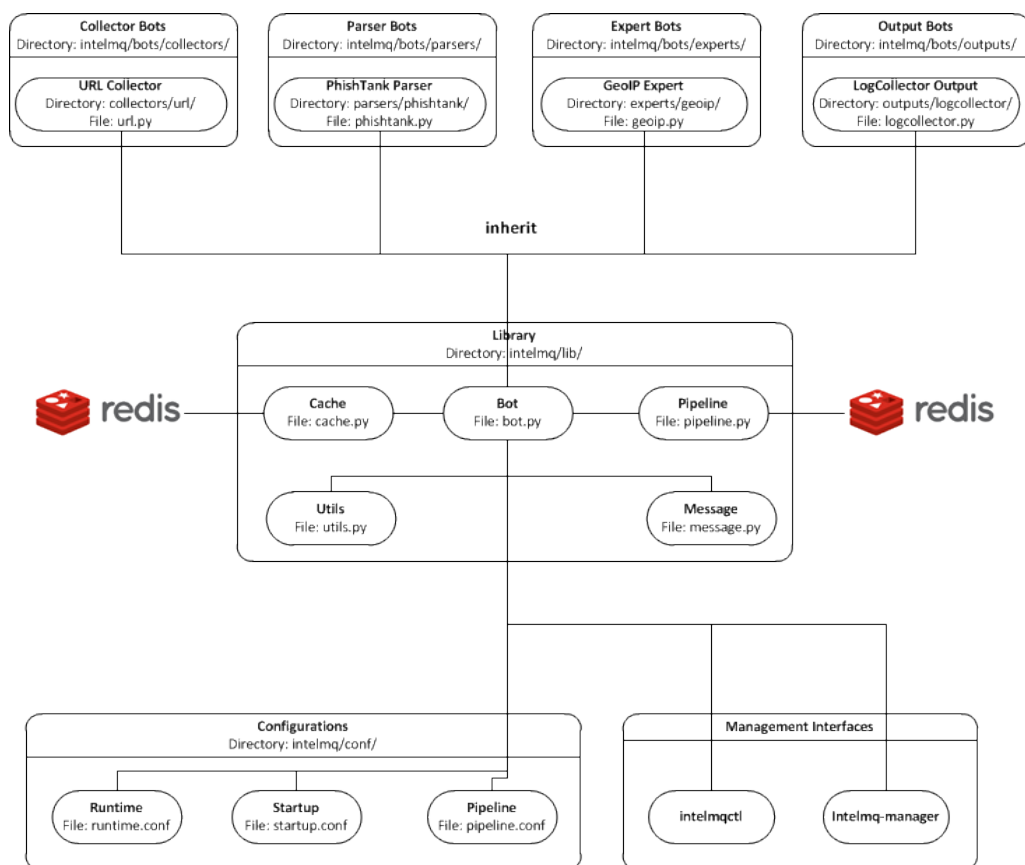


Figure 2.5 Bot Architecture IntelMQ

As seen from the figure 2.5 [57] a Redis database [58] to store the values of the system is needed. Redis is an open-source, in-memory data structure database which means that data is stored in memory. This storage capability helps keep the access to processing data fast, and handle even more transactions per second. Redis is often seen as a cache server which means that stores the more relevant data to be quickly accessible. The handicap of this technology is that data is not often written to disk as a relational database, and if the server is unexpectedly shut down data has a higher probability of being lost. However, since several events are being processed in a given time, it needs a database that could handle the requests.

2.4.3 Choosing the Supporting Platform

Both solutions are compared and, as discussed, both offered benefits and constraints. For guidance the "Benchmarking vulnerability Scanners: An experiment on SCADA devices and Scientific Instruments" [59] is used since, it brings strong points to the discussion to assess a vulnerability scanner such as: relevance of findings, understandable, good metrics, scalable, coverage and acceptable details. The relevance is tied with the direct results of the operation, and it should provide relevant data that are not duplicated. Understandability is related to the information given to the SOC operator. Good metrics are based on the identification of problems in the scanning and giving a critic score to them. Coverage is being accounted for the different feeds that are going to be possible to correlate. Moreover, the acceptability is providing results with the context of the industry standards.

Considering the objective of this dissertation, it would be acceptable to use the Alienvault USM since it correlates information with network analysis and vulnerability scanner and has the foundations to receive a security feed. However, edition of the source code cannot be made since it is a commercial product. To aggravate the situation, it is a significant solution (specification and hardware wise) to deploy on a CSIRT since it needs network probes on multiple locations to obtain information and access to network services directly to test for security vulnerabilities. In a SOC perspective, this will not be a problem but since one of the objectives of this dissertation is to reduce direct interaction with vulnerable services, it is an obstacle. Also if a CERT entity is actively scanning an asset outside their domain, it could potentially create a DoS. This DoS may happen while testing for the vulnerability on the service.

IntelMQ, on the other hand, does not do a direct scanner. It is a system where the primary source of information are security feeds, thus avoiding direct interaction with the systems. The interaction could be configured, but it is not required. The architecture is easy to understand, and it is possible to build upon the existing code, to build bots to achieve this dissertation objective. Since the application provides an easy API, it could be easily reached to identify possible false-positives later.

With this easy to develop environment, IntelMQ was chosen to support the initial hypothesis.

2.5 Risk Assessment

As stated in the previous sections one of the core contributions of this dissertation is to give an overview of exposed hosts/services of an enterprise. This contribute could be used to evaluate the risk and exposure of some enterprises. Several risk metrics could be used. "CVSS" [38]; "NIST SP 800-55" [60]; "Security Risk Analysis of Enterprise Networks Using Attack Graphs" [61]. In comparison, it will be analyzed only three metrics (CVSS, Security Risk Analysis using Graphs and VULCON).

2.5.1 CVSS

CVSS, or Common Vulnerability Scoring System, is a scoring system based on the risk imposed by the vulnerability itself, meaning that a particular vulnerability has a CVSS and it alone dictates the risk imposed. [62] CVSS does not characterize the risk by combining vulnerabilities. This behavior imposes a limitation on the scoring capabilities but simplifies the process of risk rating. Some solutions aggregate the combined risk and choose the highest value to show the risk index for the whole enterprise (similar to OpenVAS). This score is also typically used to set priorities (i.e., the higher the risk, the more priority the incident has). The metric is attributed by the following method: There are three main metrics groups: The Base Metric group; Temporal Metric Group; and Environmental Metric Group. Each group consists of a set of metrics that will help characterize the risk.

The Base Metric Group is subdivided into two subgroups: Exploitability metrics, composed by "Attack Vector Attack Complexity Privileges Required" and "User Interaction", and "Impact Metrics" composed by Confidentiality Impact, Integrity Impact, and Availability Impact. There is also a common metric between those two groups which is the scope of the vulnerable component.

The Temporal Metric Group is composed of three metrics: Exploit Code Maturity; Remediation Level; Report Confidence.

Finally, the Environmental Metric Group composed of Modified Base Metrics; Confidentiality Requirement; Integrity Requirement; and Availability Requirement. This representation can be observed in figure 2.6.

As stated in the specification: "The Base metric group represents the intrinsic characteristics of a vulnerability that are constant over time and across user environments." which means that the base metric is given regarding of context and any vulnerability chaining.

Analyzing the metric, two main sub-metrics stand out, the exploitability metric, and the impact metric. This is due to the fact that an exploit could not be consistent (exploiting success rate, every time the exploit is run there is a percentage of success associated in race conditions or memory corruption for example) but provides a significant impact on the organization or the reverse applies, and an exploit is consistent, but the impact could be null. These two factors are required to assess risk. When an exploit is not stable, it could mean two things: if it is a race condition or time-based attack it does not affect availability, but if there is a code injection and the system becomes unstable, it could lead to DoS and impact the organization.

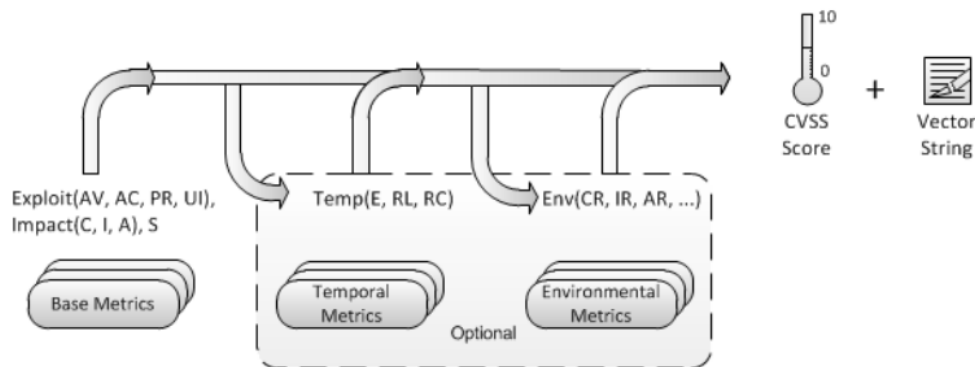


Figure 2.6 CVSS Metrics and Equations

The temporal metric tries to give an index of the environment of the vulnerability itself, for example, on several occasions researchers publish exploits on various websites but with wrong *shellcode* (injected code that will render remote access to the attacker) so the exploit will fail, but when a researcher ports the exploit to Metasploit [63] (attack framework) it would be easy to use, and therefore the metric will increase. This situation happened when the Equation group published the National Security Agency (NSA) exploits: the MS17-10 [64] exploit lied dormant until people started porting it to Metasploit; soon after the attack WannaCry happened.

The environmental metric group tries to represent the context of the vulnerability, what users are affected (administrators or simple users) and other characteristics that could help define the scope of the risk.

Scoring is the final index given by all the parameters previously discussed. It is a decimal value that ranges from 0.0 to 10.0 from low severity to critical severity.

The Exploitability metric reflects the "point of entry" that is vulnerable, often understood as "vulnerable component". Inside this metric several characteristics are presented: Attack Vector (**AV**), Attack Complexity (**AC**), Privilege Required (**PR**), User Interaction (**UI**), Scope (**S**).

AV - Reflects, in part, the context which the vulnerability is exploited. It increases with the increase of remote (physical < local machine < adjacent (local network) < Network (remote network)). The assumption is that the more potential users that could attack the system, the greater the risk.

AC - This metric indicates how much the information should an attacker have to exploit this vector. This information could be a memory address or a file path. Either way with the more information required, the more this values diminishes since with low complexity the easiest it is to be exploited. This metric excludes user interaction.

PR - This metric describes the level of access the attacker must have. The attacker could have no privilege that will result in a higher value of this metric or Low which requires a basic user session, or High which requires administrator access.

UI - This describes if the user needs to interact to exploit the vulnerability, for instance clicking on a button.

S - Available on the CVSS v3.0 this metric refers if the exploit could affect the system beyond the authority of it, for instance, if it is a cloud system, exploiting the system on a vulnerable service could allow the attacker access to other systems on the same cloud server.

In the impact metrics the following metrics are presented: Confidentiality (**C**); Integrity (**I**); and Availability (**C**). These are the pillars of information security. A threat can impact one or more of these issues.

C - An attacker, by exploiting this vulnerability could access confidential information.

I - The attacker could change information on the system.

A - The attacker could disrupt the system in a way that other users could not use it.

The temporal metrics group have the following metrics: Exploit Code Maturity (**E**); Remediation Level (**RL**); Report Confidence (**RC**).

E - Evaluates if the exploit code is public or not and if it is stable to exploit the vulnerability successfully.

RL - Evaluates if there is a patch already released for the vulnerability or if it is a 0-day or even a product that could not (or will be) patched.

RC - Checks if the source and the content of the vulnerability are feasible and from a trustworthy source. This check happens since often CVE entries could be allocated but not proven because the exploit is private.

The environmental metrics permit to adjust the score based on the affected IT assets within an users organization. They are based on: Confidentiality Requirement (**CR**); Integrity Requirement (**IR**); and Availability Requirement (**AR**). Since this is to fine tune the already determined score based on the assets within the enterprise, it will not be considered due to fine detail that would not have a significant impact on the final metric.

The same applies to the modified base metrics described in the specification. In this dissertation, only pre-established values are going to be consider according to the feeds, not with the value of the assets within the enterprise.

It is possible to classify vulnerability whitin the asset using a qualitative value accordingly with table 2.2 [38]. This table shows the classification values into five classes. This value helps management to decide what vulnerability they should prioritize:

Rating	CVSS Score
None	0.0

Table 2.2 continued from previous page

Rating	CVSS Score
Low	0.1-3.9
Medium	4.0-6.9
High	7.0-8.9
Critical	9.0-10.0

Table 2.2 CVSS Qualitative Score

In summary terms, the scoring is given by the base metric and optimally adjusted with the temporal and environmental metrics, but these values are optional.

2.5.2 Network Risk Graphs

This metric is based on user behavior. As described on Network Security Metrics [65] typically an attacker follows a specific path to attack a system. For example, it is common for an attacker to trigger a scan on the server for open ports, then connect to services and try to obtain the running version as well as vulnerable endpoints if necessary, and finally to attack using some known vectors. However, the vectors could vary, so the procedure maintains. The actions have weights to them, and for each suspicious action, it is evaluated if could be an ongoing attack. At the limit, the connection will be terminated, and the user blocked. In essence, the risk will be higher for each action taken by the user. It is a procedure to asses risk based on user interactions. Of course this could be seen as a complement of the typical vector base (analyzing payloads in each request) since it is easy to bypass it by using several sources, for example, a list of proxies that would be changed request after request or even a The Onion Router (TOR) [66] script that will rebuild the circuit upon usage of a random/determined number of requests, effectively assuming a different identity.

This risk assessment scheme is built on machine learning and often written in List Processing Programming Language (LISP). The system has a Knowledge Database (KB) that will be built upon new attack vectors.

2.5.3 VULCON

VULCON is a metric proposed in [67]. This metric is based on the CVSS score as an input to produce a final score and is aimed to prioritize the correction of detected vulnerabilities. The process is straightforward. The metric revolves in two main concepts the Total Vulnerability Exposure (TVE) and the Time-to-Vulnerability Remediation (TVR).

TVE scores the assets by un-remediation vulnerabilities per month. The total score is aggregated between all the hosts in the enterprise and from newly detected vulnerabilities to residual/unpatched vulnerabilities that were left unresolved.

TVR defines the requirement to correct a vulnerability. It is calculated based on the difference between the detection and the selection for the vulnerability to be mitigated or solved.

The objective of this dissertation consists in lowering the **TVE**, since **TVR** is the difference between the detection and the remediation and it is not in the scope of this dissertation.

Abstractly, it is stated in the paper that the computation is done in two steps. The first is a sum of weighted values between the severity score, the time (by months) that a vulnerability has been known in the performed scans and finally the time since a vulnerability has been known in the community. From each of these indexes a weight is added to fine-tune each of the parameters to the desired for the SOC. Next is an optimization process where it is being used integer programming to optimize the resolution of incidents or to reduce the attack surface as much as possible. The process is described in the after-mentioned paper.

2.5.4 Choosing the risk metric

Since Graph-Based metric involves using a more intrusive analysis, by having access to network logs and monitoring, it is considered out of the scope of this dissertation. This dissertation aims to be the less intrusive as possible to the client and gather information from the most outside sources without direct interaction.

VULCON also provides useful input for the risk indication and could be implemented as an alternative to better handling the SOC incidents. However, in this context, the lack of some input constraints such as the scans that the target is doing to itself could prove troublesome in accomplishing this step. It could also cause an increase in complexity however, it is an excellent alternative to the simpler CVSS score.

The CVSS database provides a good source of information for classifying the risk of an enterprise. This metric will be used for the rest of the dissertation.

Throughout this chapter, it was described the groundwork on this subject that could aid with the problem resolution from the concepts of enumeration, both at the network level and application level (service level) to the feeds parsing and data extraction. It is also approached some norms to understand how the incidents are managed, and vulnerabilities are disclosed. This information is essential since real-life examples are going to be handled and an understanding of how these incidents are processed is critical to avoid unnecessary delays in their resolution.

Chapter 3

Solution Proposal

In this chapter, it will be discussed the problem in question and the possible solutions that can be used to address it.

The solution proposed is based on the correlation of several feeds to enumerate addresses allocated to the enterprise, being IP addresses or DNS records. This information should be extracted using different techniques. After the initial entry point is being identified, a subset of feeds is used to check for services. Then, all the information is analyzed to detect misconfiguration and vulnerabilities in order to alert administrators or to send the information to other tools for a more in-depth analyze.

3.1 Problem Description

In recent events, an uprising of data breaches [68] due to the lack of asset management have been witnessed. Some of these assets, as previously stated, remain unpatched. This risk will increase in case of a data breach or even system compromise given enough time. Government network show a high number of web services that are still provided with Microsoft Internet Information Services (IIS) 6.0, or another outdated version. The problem is that CSIRTs can only monitor assets that they know they exist and leaving critical platforms forgotten can leave an open window to attack opportunities. Since every citizen support is being migrated to an online service to reduce waiting times, this monitoring should be constant to reduce the risk of compromise.

Another critical world event that took place was the attack with the ransomware WannaCry [69]. This attack used a known vulnerability published some months prior the event and leaked by the Equation Group regarding tools that the NSA allegedly developed and used. This vulnerability exploits the Server Message Block (SMB) service which is often open with any windows installation. The problem was that many servers remained unpatched and exposed to the Internet. This lead many companies to lose their files and were demanded a ransom to decrypt and retrieve them. This behavior led several ISPs and companies to shut down their whole infrastructure afraid of losing data [70]. This situation had a nefarious impact on the availability

of those companies. This risk could be avoided by having a security policy to patch those systems as soon as the patch was made available or block Internet-facing assets to be unnecessarily exposed.

3.1.1 Lack of Security awareness

Enterprises nowadays disregard their security policy, the excuse that is given is often concerning financial conflicts. Since security is not often related to economic gain, it is often left aside.

Today, hefty fines are applied to those who cannot keep up their client data secure, and some companies are even legally bound to disclose any compromise that occurs. However, keeping these standards requires infrastructure and experts on the field to lead the enterprise to a better security policy.

Also, it is seen that outsourcing contractors often leave backdoors on infrastructure to ensure that their access is maintained for maintenance operations. As an example, several Enterprise Resource Planning (ERP) software require to expose the Remote Desktop Protocol (RDP) to the Internet to be able to update the systems instead of choosing to connect over VPN.

This exposure is difficult to detect from a client perspective, that often do not have the expertise to detect the exposure in equipment configuration.

3.1.2 Difficulty in detection of Hosts

Due to the globalization of the Internet and services, such as shared hosting where clients can rent hosts to publish their enterprise or personal web page, changed the way how host discovery takes place. In the distant days of the Internet, people leased IP address space. This registry was easy to find by consulting a WHOIS (similar to the already described RIPE database) database and querying it for the name of the enterprise itself. Nowadays those shared hosting services share the same IP although with different hostnames pointing to it. For example, a Web server now has what is called as Virtual Host. HTTP protocol states that when a client tries to access a Web page, it should include the Host Header. This header refers to the FQDN that was entered in the URL bar. This FQDN helps differentiate from several Web applications using the same server and guiding the Web server to the correct application. Thus it is necessary to adopt further techniques to allow us to discover these new hosts and applications.

3.1.3 Difficulty in detection of Services

Following the previous section, it is possible to infer that services can as well be challenging to locate. Although other services than HTTP(S) are not as easy to be shared, they still need address space. When searching for some service, people refer to Web application (something that appears on the browser) however, there are other services capable of being reached like RDP, File Transfer Protocol (FTP) or even databases. These services are typically considered critical for the core business of companies. These services can be hard to detect if there is no direct link to them or

does not show on search pages. There are ways of discovering them but are typically very noisy such as port scanning.

3.1.4 Lack of automation of the process

In penetration testing, a portion of this discovery process is done to help define the scope of the project. However, this discovery is typically a one time process. Solutions like AlienVault OSSIM performs network scans the check for new hosts and services. Although this process is good at enumeration, it is very noisy and could impact the network. In networks external to the AlienVault sensor it will consume the link and probably trigger IDS alerts if an exclusion has not been made for that test. Besides, it will also only perform network scans on the selected scope, and everything that was not previously defined in the scope will not be scanned. This methodology is excellent to ensure that other hosts will not be scanned but in ever growing networks with cloud solutions it will not detect some applications, leading to the asset not be in the inventory.

3.2 Possible Solutions

This dissertation proposes to implement a pipeline that follows the logic described in figure 2.1. It starts with each bot collecting as much information from the initial points of entry (domain name, IP address). In the discovery section, an attempt to discover new sub-domains, new addresses ranges, and services associated with said IPs is going to be made, using sources like Shodan and Censys. It is possible to determine if the service is vulnerable to a subset of vulnerabilities (those that are automatically searched by the feed service); also with the banner grabbing provided by that service is possible to try to enumerate some platforms like the server running the application, the version and some technology associated. This technique is not new, and Kern [71] already discussed it. It was stated that "open source components are integrated into their products. However, one aspect that is often neglected is watching for known security vulnerabilities of the used components", this leads us to believe that enumeration of some of the components could lead to indicators of vulnerabilities. It is possible, for example, to further this investigation by using services like "BuiltWith" to identify possible Content Management System (CMS) and dependencies. This identification is helpful since by using this enumeration it is possible to quickly check for already known vulnerabilities associated with those versions. With all this information it is possible to scan for databases of vulnerabilities to find a match, and if a match occurs, an alert can be sent to notify the underlining operator.

Some notes are necessary: the process so far is OSINT based, and almost no interaction was made to the targets. However, it is possible to improve the accuracy rate by testing the server directly. For example, having the name of the server, the port and the service as well as the underlining technologies, it is possible to test for known critical vulnerabilities: Joomla Remote Code Execution (RCE) [72] or similar. However, this could impact availability, and inevitably it will trigger the IDS on the monitoring side.

Figure 3.1 shows the data model of the developed application. This model is not complex since the application does not require a complex model. In this data model, vulnerabilities detected are not stored since vulnerabilities can be disclosed and it would give a wrong idea of the data. With this model, a correlation needs to be made to check for vulnerabilities. Since that data is relatively dynamic, it had no positive effect in storing it. CPE values are directly correlated by the NIST database.

On the DNSNAME table, four attributes can be found:

Name - The FQDN of the asset discovered. This tuple is the primary key on the table since it is unique on identifying different assets using this value.

ip - The IPv4 address of the FQDN. This field is a required value. Different FQDN can point to the same IP address, meaning that different applications are running on that server or that the server has different names.

firstSeen - This record saves the date when the entry was discovered. This column helped debug the application

lastSeen - This is a debug value while developing the application to check if the application was getting new values with the addition of new feeds while keeping the previous results. This field was abandoned whilst the solution matured since it now creates a new file to keep historical data.

On the Services table the following attributes can be found:

name - This column is a reference to the "name" column of the DNSNAME table. In case the host did not have a name (i.e.: RIPE search) an IP address will be presented.

port - This is the port number running the service detected

banner - This column stores the value of the banner upon connection or, since the API tries to enumerate the service, the banner of the detected service

cpe - This column stores the CPE detected by Shodan if there is any. If no CPE is found a local database can be checked to search for specific strings to identify the service

Following the work-flow design, it was aimed to implement it in the intelMQ platform. This workflow will ensure that the tasks are performed repeatedly and it will be easy to integrate with the already used frameworks. The information will be stored in the corresponding database (Redis database) for statistical analysis in the future. Information like date, name, IPs, ports, services running and technologies associated will be stored for later analysis if the need arises.

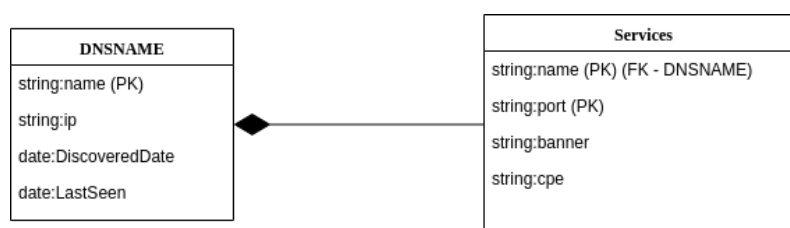


Figure 3.1 Model of the data to be stored in the Database

3.2.1 Tackling the Lack of Security Awareness Problem

One way to manage this problem is to give visibility of the network to the IT team. This visibility will help them understand if a system is exposed to the Internet and, by evolving the solution, if there are known CVEs for the CPE of the service. This leads to a better security cleanness of the infrastructure where deprecated systems will have a higher risk. This overview can also show to upper management if there are actions need to be taken to the infrastructure to mitigate risk. This map will help determine what can be a threat to the enterprise with the intention to fix or mitigate that threat.

3.2.2 Tackling the Difficulty in Host Detection

Host detection, as stated before is difficult since networks are spread and, in some cases, the ever-changing state of the network (network services are created and destroyed in a dynamic way (i.e.: a city council that created platforms for different events every year) is a challenge.

To undertake this challenge, the first step is to enumerate the IP address allocated to the said enterprise. If an enterprise bought said space, it would be easy to make a query search to find it. For example, RIPE full database search could be used to search for "inetnums". "inetnums" are addresses ranges associated with an entity/object. If in the description of an entity/object there is a match with a given string it is possible to locate the "inetnum" entry to know the address space.

This information gathering is essential since if the same entity buys another range or IP classes, it will show a list of the new (current) IP address space allocated to that specific query keeping it as up to date as possible.

The following discovery will be subdivided into two categories: Host Port Search and SubDomain search.

The first one are APIs that have port discovery capabilities and enumeration meaning that it does not only focus on subdomain search but port enumeration.

The second one are APIs that directly or indirectly returns subdomains and hosts providing a subdomain or IP. Indubitably some searches in the previous category help retrieve said domains. However, a distinction should be placed since one is different in its core.

A small note should be placed. APIs refers as resources since some pages do not offer APIs in the true meaning of the acronym and a bit of page scrapping is needed. Page scrapping is the term that, elements of a request are retrieved to extract information.

3.2.3 Host Port Search APIs

In the following section, it will be enumerated the chosen APIs to try to solve the Host Service/Port enumeration. These APIs scan the network for some default common ports in a cyclic way, meaning from time to time they pick a random address and port and perform banner grabbing (connect to the port and store the result). Some of them also try to enumerate the service if there is some protocol initiation needed.

Saying this, of course, these APIs do direct access to the target Host. However, since it is the API, the IP address of the requesting device will not get recorded in the target host, keeping the discovery passive.

3.2.3.1 Censys

This API was created following the ZMap project [22]. The ZMap project was a project performed at the University of Michigan that would map the entire IPv4 range in search of some vulnerabilities like HeartBleed [73], Freak [74], and botnets such as MIRAI [75].

It quickly expanded to perform several other port scanning and is now an excellent source of information from services running on the host to handshake information such as exported certificates.

It offers free and paid access. The free access is limited and can retrieve at most 1 000 results per query, meaning that if the API found more than 1 000 ports, it could only show 1 000. In addition to the result per query limitation, there is also a result per month limitation where the API is still very limited. Paid versions offer more access and more results. Prices range from 99 United States Dollars (USD) per month to "Custom" (for enterprises).

In this particular API, an User ID (UID) and an API Key is required to perform queries to the service. The service returns information in the JSON format which is easily parsed by python JSON package.

3.2.3.2 Shodan

Shodan is the *de facto* Internet scanner. It is known for their consistent results and an API that is mature enough to support several applications. It scans for the most commonly used ports as well as ports that are associated with C&C hosts. It can also identify service running on the port and perform some enumeration. For example, it can detect if a DNS server is an open resolver (open resolver is a DNS Server that accepts the resolution of any request to any domain and could be abused to perform Distributed Denial of Service - DDoS - attacks and evade detection) [76].

It is a paid service although it is only 5 USD on discount rather than the usual 49 USD. The only reason this API was chosen is that of the versatility, the adoption and the proven results among the community that this service provides make it unique. This feed is the only one to provide a reliable CPE. Moreover, since it is not subscription based its price is considered justifiable.

With the paid subscription limits are removed, meaning that it is possible to retrieve as much information as needed by the platform. This one is essential for us to be able to get as much information as possible about the target.

This API only needs an API key to be able to query the service.

3.2.3.3 ZoomEye

ZoomEye is a direct competitor of Shodan. It tries to perform the same tasks. It is a new engine that scans the Internet in search for open ports that later enumerates.

It can perform a CPE enumeration although it does not comply with the standard thus being difficult to correlate to other databases.

The API is limited to only 30% of results to free users with an upper limit of 1000 results. This limit degrades the quality of the results severely. Unrestricted results need a special account that is priced by directly contacting the service through email. As stated in the API documentation: For avoiding some malicious purpose on making use of ZoomEye resources, some limitations are in place on results from the endpoints. The register users could only get 30% of it, contains 10,000 results upper limit.

In this service, an API key and API user is needed to query the service. A special header does the authentication in the HTTP Protocol. Results are returned in a JSON format.

3.2.3.4 FOFA

FOFA is a service to enumerate services in a host. Similar to ZoomEye in allows for some subdomain enumeration by retrieving information in the reverse lookup of the IP address. As stated before, this feed was not implemented due to time constraints. It allows for data retrieving using an API with results being retrieved in JSON format.

3.2.4 SubDomain Search APIs

In the following section, it will be described all the APIs that were included to gather subdomains and hosts. Some APIs could give a little more information, but it will be focused only on the domains and subdomains that are possible to enumerate.

3.2.4.1 Bing

Bing is a product of Microsoft. It uses Microsoft search engine to crawl the Internet in search of references to websites. Using some *dorking* (specific queries to improve search, the term is related to Google Dorking [77]), it is possible to gather some samples of domains.

Previously Microsoft provided an API to query the Bing search engine. However, this has been deprecated and replaced by a payed API subscription. A trial account can be created, but it will not last long (about 30 days).

The solution is to mimic a typical user in the Bing search engine and return the results. If the API is abused, the connection could be detected as a bot and blocked. Therefore, the number of requests should be limited.

3.2.4.2 Google API

Google Search Engine provides an API to query for results. This interface helps programmers since the results are structured in JSON and easy to implement in their application.

Google is also limited by the number of requests that the application can perform, but the free developer account provides more than enough to be able to automate the process almost freely.

If several requests need to be made a subscription should be paid to perform them without any problem.

3.2.4.3 Baidu

Another Web search engine known in the Asian region is Baidu [78]. Leveraging the same technique as Google and Bing, it is possible to discover additional domains using this source. It does not provide an API, but if a Web scrapping is performed, it is possible to extract the URLs the search engine gives. By extracting the domain information from the results, it is possible to enumerate several subdomains.

The connection may be throttled if an attack is detected. Also, due to the remote location of the service, it could take a time to contact it.

3.2.4.4 DNSDumpster

DNSDumpster [79] is a product developed by Hacker Target helping Penetration Testers perform a first assessment of the network. It can perform other tasks like retrieve information from HTTP headers that could be beneficial in enumerating additional technologies.

It was created to be used by humans and not automated tools. Therefore some protections were implemented like the Cross-Site Request Forgery (CSRF) tokens applied, meaning that the application should handle this tokens in the first contact with the page to be able to pass it in the domain information request. Therefore it has used a page scrapping technique.

It is somewhat restricted and is limited to only 500 results per request.

3.2.4.5 Virus Total

This API [80] is usually used to detect malware either in files or Web pages. However, it stores known information about subdomains of a given domain.

It is a Google service and provides an API to query for results. It has four requests/minute limitation. However, since the search is done by domain entry, and all data is retrieved without paging, there should be no issue.

It uses an API-key to query the service and results are provided in JSON format.

3.2.4.6 Yahoo

Another known Web search engine is Yahoo [81]. Keeping in mind that is powered by Microsoft Bing, it was found to mangle some results thus creating a possibility to obtain different results than their engine.

By querying this service in complement to Bing helps to avoid a lockout by Microsoft engine.

Using this search engine, it is impossible to query the API since it has been deprecated for years. With that, a different approach needs to be taken such as Web scraping to collect the information needed.

3.2.4.7 Yandex

This search engine [82] is known in the Asia continent. It is a Russian search engine similar to others search engines. Being an independent search engine, the results given are almost entirely different from other search engines thus giving a great perspective of what sites are more known in other regions of the world.

It provides an API to query for results. It requires a phone confirmation to create an account and, in some cases, solve a Russian "Completely Automated Public Turing test to tell Computers and humans Apart" (CAPTCHA) which can prove difficult in western keyboard where Cyrillic is not available. The service comes without cost.

3.2.4.8 Threat Crowd

Threat Crowd is another threat discovery engine. It analyses DNS sinkholes to keep track of infected hosts and domains. It can also be used in this context to query for subdomains of a domain since it stores that information.

It provides an API to retrieve results, and it does not seem to block any results. The service is free to use. Results are gathered in JSON format.

3.2.4.9 Hacker Target

Hacker Target is a service that provides a collection of tools to perform OSINT and even vulnerability analysis. There are both paid services through a subscription or a one time fee to perform such analysis. However, the service used in this dissertation was free of charge, provided the API was not abused.

The used service does not provide an API however Web scraping can help us get the results needed. The service provides APIs for certain operations, for example, an API that retrieves the HTTP headers of a Web service. However, this will increase the usage severely, and requests would be blocked. For that reason, the application will not use those APIs and opt for another that would not block it such as Shodan or SecurityHeaders.io.

3.2.4.10 SecurityHeaders.io

SecurityHeaders.io [83] is a web application that evaluates the headers of a Web application to check for misuse or lack of security. This is not used in this dissertation to check for vulnerabilities but to retrieve the headers of the application since the other feeds do not support virtual hosts. With this, it is possible to check if the service is up and retrieve partially the information it contains.

3.2.4.11 Exalead

Exalead [84] is known by OSINT specialist for document search. Exalead is a search engine that looks for exposed documents. However, it aggregates results from different search engines thus being possible to extract information from it. It provides some unique results but tests shown that several of the results are cached and the subdomain no longer exists, so an additional check is necessary to understand if the domain exists.

It does not provide a web API, so the extraction method used is web scraping. It is severely limited if an attack might be undergoing. If an attack is detected it is necessary to resolve a captcha to unblock it again. It will block in a per IP base, so only from the same IP it is possible to unblock it.

3.2.4.12 Passive Total

Passive Total [85] is a complete OSINT platform where several indicators can be monitored such as compromised hosts domains and respectively subdomains.

Usually, this service costs a subscription but a limited account can be obtained to access a subset of the available information.

It provides an API to query the services with results being retrieved in JSON format.

3.2.4.13 Qwant

This service [86] is an anonymous French search engine. Since it is a French company, the results might be similar to others on that region.

Although free and anonymous it monitors to avoid abuse. It does not provide API, so Web scraping is used.

3.2.4.14 SearchX

SearchX [87] is a Web search engine of Web search engines, meaning that it connects to Google and Bing (as well others) to retrieve its results. It aims to perform anonymous searches since Google analytics can track people based on several aspects like IP address, search methods and preferences and cookies.

It does provide an API, so Web scraping is not needed. It is very limited since in case of abuse it will throttle the connection.

It helps complement the Web searches and avoid being block by redistributing the load.

3.2.4.15 BuiltWith

This service [88] is intended to enumerate technologies associated with a particular website. This service could come in handy to enumerate technologies with vulnerabilities. However, the API is severely limited to non-paying customers. Prices range from 295 to 995 USD per month. For big SOCs this could be a great aid to further detect vulnerabilities using non-invasive methods.

It provides an API to query their database however its limited and only a small number of requests can be made. The results are also limited due to the free nature of the API. However, it is possible to enumerate some subdomains using it.

3.2.4.16 Certificate Transparency

Certificate Transparency is not a service in its true meaning. It is an improvement to SSL/TLS technologies. In the description of the project it has stated the following that best describes its nature [89]:

Googles Certificate Transparency project fixes several structural flaws in the SSL certificate system, which is the main cryptography system that underlies all HTTPS connections. These flaws weakened the reliability and effectiveness of encrypted Internet connections and can compromise critical TLS/SSL mechanisms, including domain validation, end-to-end encryption, and the "chains of trust" set up by certificate authorities. If left unchecked, these flaws can facilitate a wide range of security attacks, such as website spoofing, server impersonation, and man-in-the-middle attacks.

Focusing on the domain verification, domains need to be recorded, and since the framework is public (to verify authority), it is possible to store all the domains that use SSL/TLS with this protection thus having a window of opportunity to enumerate subdomains. Luckily there are search engines to search for these records. By consulting this feed [18] it is possible to search for them.

This search engine provides the results in JSON format (as well others but since it is easier to deal with the format it was the one it was chosen). It is free, and it does not seem to implement any restrictions.

3.2.4.17 Alienvault OTX

Alienvault OTX [54] is another threat engine, more precisely Open Threat eXchange (OTX). It is an engine that gathers information from several probes installed in different parts of the world and different companies and tries to exchange information about a known attacker. This feed aids SOC operators to block incoming threats before harm can be done.

Due to this collaboration and information gathered from several sources, it is possible to query the engine for information about subdomains. This particularity can be leveraged to investigate further subdomains stored by this engine.

It provides an API using JSON objects, and it is free to access provided an API key freely available.

3.2.4.18 Threat Miner

Another threat intelligence engine is Threat Miner [90]. This engine is similar to other described in this section and act in the same way.

It is free to use and does not seem to impose any restriction on its API. Its results are gathered in JSON format using the API.

3.2.4.19 findsubdomains

This feed [91] proves to be one of the best found. It allowed to retrieved a considerable amount of information with one request. Unfortunately, this feed had become accessible with a paid subscription and only allow for the enumeration of 50 records in the free subscription. This is a severe limitation. However, it was included since it allows for the possibility of enumerating additional hosts.

3.2.4.20 Shodan

As stated in the earlier Shodan is capable of discovering hosts. It performs reverse DNS lookups as well as SSL/TLS certificate inspection. DNS reverse lookup is the procedure that instead of resolving a DNS name it is given an IP address and a name is returned. In the certificate procedure, when a certificate is emitted, it is emitted to a domain name since IP addresses can be changed, the domain name, even if changed the host can still be used. Connecting to the port and presented the certificate it is possible to determine to whom it was issued.

As stated, Shodan uses an API and has a small cost to the solution.

3.2.4.21 ZoomEye

ZoomEye uses the same techniques as Shodan to enumerate Domains. It uses both reverse DNS lookup and certificates. However, their API is limited to 1 000 results, as stated before.

3.2.4.22 Censys

This service can be used to enumerate domains by searching the SSL/TLS certificates. The platform does not perform reverse DNS, it can only check for known certificates that the service provides.

One particularity is that, due to its nature, it can only store the certificate in the default configuration of the server. If the server has multiple applications (i.e.: HTTP), it will not be possible to enumerate the others using this service.

3.2.4.23 Binary Edge

Binary Edge [92] is a Platform as a Service (PaaS) that allows similar behavior as Shodan and ZoomEye but with some extensions such as support for data leakage detection and penetration

testing aid of the target networks. Whoever due to the limit to 50 requests per month on the free tier, although possible to use in production, it is hard to test and implement correctly. Paid tiers allow for more requests, but since it is paid monthly, the cost to test this would be high.

It provides both an API and a Web interface.

3.2.4.24 Security Trails

Security trails[93], similar to Binary Edge is a platform that harvests data on many domains such as subdomains, certificate transparency logs, Domain-based Message Authentication, Reporting & Conformance” (DMARC) [94] logs and IP address & port data. Then they report them on a Web Interface. It also provides an API to access historical and present data. The problem with this API is the limit of 50 queries per month that would be a setback developing and testing the application. Therefore it was not implemented. However it could provide great help in the future.

3.2.4.25 FOFA

FOFA is a rather new PaaS similar to ZoomEye. It is a Chinese platform that provides a Web Interface, an API, and a Command Line Interface (CLI) platform. In the sense of discovery, it does not stand out from the rest of them. It specializes in service discovery with a great collection of ports scanned. However, it does not provide a standard CPE. It provides a good human interface to analyze the results. It was not implemented but deserves a reference due to the potential it could give in the future.

3.2.5 Tackling the Difficulty in the detection of Services

Another set of problems are the detection of services running on the found hosts. From an attackers perspective, the next step is to enumerate the services and running version of the hosts. This step is challenging due to several reasons.

Some application may divulge their software name and version soon after establishing a connection to the port. Other requires further interaction.

This interaction breaks the requirement of this dissertation which states that no interaction cannot be made. Enumeration of remote systems is, according to the current Portugal legislation, an intention of attack and could be prosecuted.

CPE enumeration can be done using several programs like Nmap, Amap [95] or even the suite OpenVAS (that calls Nmap to enumerate the service) but that requires interaction with the hosts.

Fortunately, Shodan performs this enumeration. Resolving the IP and looking it up on the Shodan API it is possible to enumerate some services that are running on the target machine.

The same applies to ZoomEye, Censys and FOFA. However, it does not follow the standard CPE nomenclature thus proving hard to work on.

For that reason, it is only possible to get reliable CPE information from Shodan and it is where the information is going to be obtained in order to issue the alerts.

Other scanners, particularly Web application scanners could be implemented in the pipeline to improve better detection, but the costs of having access to that API will be significant and impossible to undertake in this dissertation.

Regardless of constraints, it can also be possible to connect the pipeline to a second phase of the engagement performing an active analysis and a network scan. Furthermore, the pipe can be connected to OpenVAS suite to perform extensive analysis given the new wider engagement perimeter.

The OpenVAS suite can also be extended to connect to applications in order to perform active scans like Arachni [96], Acunetix [97] or others. There are also implementations of the Metasploit framework to automate this processes using the "check" command to try to obtain vulnerabilities in the different found services.

However, this is not the core of this dissertation. A complete pipeline that can be improved in the automated detection of vulnerabilities should be considered for future work.

3.2.6 Tackling the Lack of Automation of the process

In the previous section, it was presented some tools prone to automation, OpenVAS, Arachni, Accunetix, and Metasploit.

Accunetix and Arachni are Web Application scanners that given an entry point crawl the application and test for the vulnerabilities in their database. It performs scans for Open Web Application Security Project (OWASP) Top 10 [98] and more. They can perform authentication before the application given the correct inputs. They are used by several people such as penetration testers and software developers (to test their code in the Software Development Life Cycle -SDLC).

It was also presented OpenVAS and Metasploit. These applications, as well as Tenable (former Nessus), perform network vulnerability scanners not restricting to web applications. Needless to say that these applications interact directly with the service in order to check for the vulnerability in question. Several try to get command execution in order to check for the vulnerability. This action is further down the engagement testing, and it is not the focus of this dissertation, but it is helpful to understand the scope of the work.

In the state of the art section of this dissertation, it was addressed some technologies like IntelMQ and OpenVAS. OpenVAS being a network scanner does not include the process of network domain and subdomain discovery. That process should be performed beforehand. OpenVAS has connectors and an API that is possible to connect and perform scans. For example, Metasploit can connect to OpenVAS [99] and issue scans as well as consult the results of scans. However, Metasploit cannot perform OSINT scans.

IntelMQ is a feed aggregator. It consists on consulting feeds and check for compromised domains. This tool has been discussed in the state of the art of this dissertation. IntelMQ is probably the most indicated application since it is already used in the industry for the CSIRTs of the EU as stated in their documentation [100].

It is possible to use already established tools, like Recon-ng to handle these scans. Due to the framework, already established in the major offensive distribution (Kali Linux) it is possible to add this feeds to it and with a wrapper define a pipeline using only tools from that distribution. Some of the scripts are being ported to that framework so it will aid penetration testers on their task [101].

Recon-ng is not just a subdomain finder. It can also gather other information as reverse DNS, contacts, geo-location and other. The big advantage it is that it is already defined in the offensive community and the Remote Procedure Call (RPC) interface, as well as the modularity of the framework, is a big advantage to it.

However, due to some constraints in the development (framework dependencies), it is not too easy to develop, mainly due to some constraints in the pipeline (redirecting output to other feeds) for example, it would need a wrapper to that and command the RPC to issue the commands. So, in the case of brevity and proof of concept, the program written for this dissertation was written with scripts somewhat compatible with the code could be later port to that tool.

Concluding, automation can be implemented in a configuration similar to "cron" [102] running the program and piping the output to another tool or using the concept of bots, already described in the state of the art section to monitor changes and alert SOC operators about them. Other programs can also be used in complement to this solution, and the code is being ported to other tools, so the concept is used in other scenarios.

There are several concerns left of this approach. As stated in the previous section, some APIs performance throttling or even block in case of suspicion of an attack. This limitation can prevent the pipeline of getting new results, so the load applied to those APIs need to be taught beforehand risking the blockage of the service until a captcha needs to be resolved or a timeout is exceeded in order to query the service again.

The main blocks that were found were: captcha resolve; IP address block in given time, API reach limit and IP block in a black hole.

The first one it presents a captcha to be resolved by the user. The captchas complexity can change depending on the reputation of the IP address source, like in case of the Google Search Engine, or can be a simple captcha that even a small python script can resolve it [103] this bypass was not implemented since it could be considered abuse to the system.

Other blocks like the IP address block (time) is given a timeout by IP address. The connection is then throttled, and the service refuses to answer the requests in the, whoever since the connection is accepted and keep-alives are sent, no real request is made. This behavior was observed in DuckDuckGo and blocks the script to continue with any other request, stopping the pipeline. Mitigation was introduced to implement a timeout to continue the application, but was removed due to continuous block.

API limit reached is when the number of requests or the results given by the API reached its maximum. In this case, there are some alternatives. Either upgrade the plan to withstand the search or, in the cases that a human interface is present (i.e.: Google search), page scrapping could

be done to try to extract the information. Often this kind of services will block if continuous page scrapping is performed.

Blackhole is a technique where the browser is redirected to a blank page where it is not possible to obtain any information and often will take time to respond to block the offensive requests.

All these issues are relevant and should be addressed to deliver a product that will give results continuously. For the latter concerns, it should consider keeping the requests low so it will not trigger an alert and do not impact the core service consulted since that would be an attack. It is possible to throttle the connection as well as limit the requests per API, however, fewer results could be gathered. That is why other redundant search engines were added, not only because it could give other results, but also not to stress one API.

Therefore it is assumed that the application only get results once a day for a single domain, and it should not trigger any warning. Some services were also left aside in the pipeline since it would exhaust the API and then give partial results. This behavior was the case of Censys. Their API is limited, and quickly it would be exhausted.

3.2.7 Developed application

In order to test the proposed methodology, an application was developed. The core of the application was developed to be console based, whoever a Graphic User Interface (GUI) extension was built upon that layer to help demonstrate how the application can work and help users to use it outside of an automated context, for example in an engagement to discover exposed hosts.

This application queries all the feeds already described in this dissertation and extracts information about the target network. This information is saved on a database to be later used in the rest of the process. Several checks are done to avoid false positives in this stage. After getting the results, a new set of requests are made to determine the software running on the detected targets. This enumeration serves as an intermediate step to later correlate with another feed, in this case, the NVD CPE database, to detect vulnerabilities associated with the detected service versions and present them to the user or to new tools to further investigation.

The terminal application serves as a way to automate the process to other tools and interfaces where the graphical interface helps an operator to single-use it and process the results in a user-friendly way.

3.2.7.1 Terminal

The terminal application (console application) developed started to test the main concept where several feeds would be contacted and print out the discovered hosts. However, it was discovered that a lot more functionality could be added to it in hope to improve the results gathered and to be a little more flexible in what searches are done.

Keeping in conscious that the application should not stop when a feed fails (times out, unreachable, wrong data provided like a change of schema) the application should be able to

handle those errors and log them into a file to better understand what is happening behind the scenes.

```
usage: osintmap.py [-h] -t TARGET [-r RIPE] [-o OUTPUT] [-d] [-g GRAPH] [-R]
                  [-v] [-c CORRELATE]

Discover subdomains and map an attack surface of company

optional arguments:
  -h, --help            show this help message and exit
  -t TARGET, --target TARGET
                        Target Domain
  -r RIPE, --ripe RIPE  RIPE String search
  -o OUTPUT, --output OUTPUT
                        Output format (Text/SQLite)
  -d, --domains          Print Domains
  -g GRAPH, --graph GRAPH
                        Print Graph
  -R, --reverse          Perform Reverse DNS Lookup on IPs gathered
  -v, --vulnerabilities
                        Get Vulnerabilities from CPE
  -c CORRELATE, --correlate CORRELATE
                        Use file to correlate vulnerabilities
```

Figure 3.2 Application developed - Terminal

The application relies on a configuration file where several options can be chosen, for example:

- Number of pages to retrieve from feeds
- Log location
- API keys
- Usernames and passwords for some feeds

With focus on the number of pages to retrieve, this is used to avoid triggering the bot detection in most of the sites (i.e: Bing and Baidu).

Another improvement done was the RIPE database search. Since this method is aimed at considerable networks, several of them rely on DNS. However, it is possible to reserve IP address space. Considerable enterprises buy IP address spaces in order to keep their services and avoid a constant update of their DNS records that could lead to temporary downtime or unnecessary complexity of management. Therefore it is advised to search for this registered address spaces.

By using the RIPE search query page, its possible to search for numerous parameters for instance:

person and maintainer pair - Entity responsible for maintaining the block of information presented;

as-set - Collection os Autonomous System Identifiers;

domain - Domain Name. If registered, there may be information about the subnets allocated;

inetnum - "Specifies a range of IPv4 addresses that the inetnum object presents. The range may be one or more addresses." [104];

inet6num - Same as inetnum, but in IPv6;

inet-rtr - "This attribute is a valid DNS name of the router described by this object. It cannot end with a dot." [105]

irt - Abbreviation for "Incident Response Team" responsible for any security related incident to that object [105];

mntner - Maintainer of the object, being an entity;

organisation - Also known as "org", it is the organization that holds the object.

This information can be found in more detail at the RIPE database Documentation [106].

With inetnum, it is possible to search for a string in the registry of said allocation. For instance, a search for "Universidade do Porto" is performed, the IP address space allocated for the University of Porto will be provided.

This search will not show IP addresses that are on shared hosting or even in some cloud providers.

Another improvement done is the "alive check". Some domains retrieved by APIs are "historical" data. This check means that although they were registered, they do not point to an IP address. By checking if they resolve to an IP address it is possible to remove some false positives and improve performance. To keep the search passive, an external API will be used to resolve the name. There are several options:

Google - Google offers an API to resolve DNS names. This API returns a JSON object with all the information of a simple query using the dig or nslookup tools. It allows the use of different records such as MX and AAAA. This API, however, offers a limit on the queries per second made and may even block if continuous abuse is detected.

DNS Checker - This API tries to resolve the address in several DNS servers. This method is beneficial for those that want to see if there is any block or difference between countries to resolve that specific IP address. It also supports different records. It also comes with some limitations such as the limit of queries and the need to send a CSRF token to avoid abuse.

Whois (Vodien Group) - This website provides a simple DNS resolver. This website is not stable but if the requests are not made uncontrollably (several requests per second made to the API) it provides good resolution of address without any block.

Other options are available but it was chosen to analyze these three since it is not the focus of this dissertation to determine the best API for resolution but one that provides accurate results to address the problem. Since almost all of them employ limits, either per second or query performed the Whois (Vodien Group) was selected since it did not employ any lock. Sometimes, even without

stressing, the API will lock and return an error but retrying the request will go through without a problem.

Another improvement done was the correlation of the services discovered (CPE) with the already existent CPEs for the service running.

CPE enumeration is performed by (typically) identifying the service and then the version running. Since different services require different requests to enumerate their version, it is hard to do it passively. Active tools such as Nmap or Amap allow enumerating services by using their database of known applications to enumerate the service. Luckily Shodan provides a CPE for each of the entries retrieved in case it already enumerated the service.

It is possible to extract and save that CPE for later processing from Shodan. ZoomEye and FOFA is not consistent with that enumeration. Therefore, it was not used.

Since CVEs do not usually contain a direct relationship with a CPE, it is difficult to do this offline. However, NIST provides a platform where it correlates the CPEs with CVEs.

During the development of this dissertation, NIST had undergone a small change in the schema and on how it stores the data. Usually, there was a file that contains all the data in the NIST database and, after retrieving it, it was possible to search it locally. Since the schema changed, the file was deleted, and now there are several files, each for each year of entries. Downloading everything and parsing everything would be very resource consuming (downloading, extracting and parsing every year from 2002 to the current year and so on. Fortunately, NIST also provides a search page where a CPE can be searched and then gather all the related CVEs associated with it. Page scraping the results will give us a list of possible vulnerabilities.

Reiterating that this is a possible vulnerability. It can only be confirmed by directly contacting the service which would break the "no interaction rule" as well as it can be considered an attack and therefore illegal.

Since not always Shodan could interpret the CPE (in case of virtual hostings) a small optimization can be done using an option of the interface. It is possible to write a dictionary that searches the banner and check for the occurrences of a string and then presents the corresponding CPE. This correlation can be done in a file on the root of the program directory "correlate.txt".

Threading is also used to improve the process of IP resolution. Sub-Process are created to improve the resolution of IP address since, in big networks, this process can take a considerable amount of time. Threads are lighter but due to the application architecture, saving on the database will cause the thread to exit and do not process anything else, therefore processes were used.

On the service check, this will not be possible since Shodan limits the API to 1 request per second. In order to improve results if one second has not passed since the last try, the application will halt and proceed after one second so it will not stress the API, improving the detection rate.

3.2.7.2 Graphic

The Graphic program is just an interface to the terminal application. The GUI application was developed in Python QT5 (PyQT5) which is the implementation of Qt5 for Python [107].

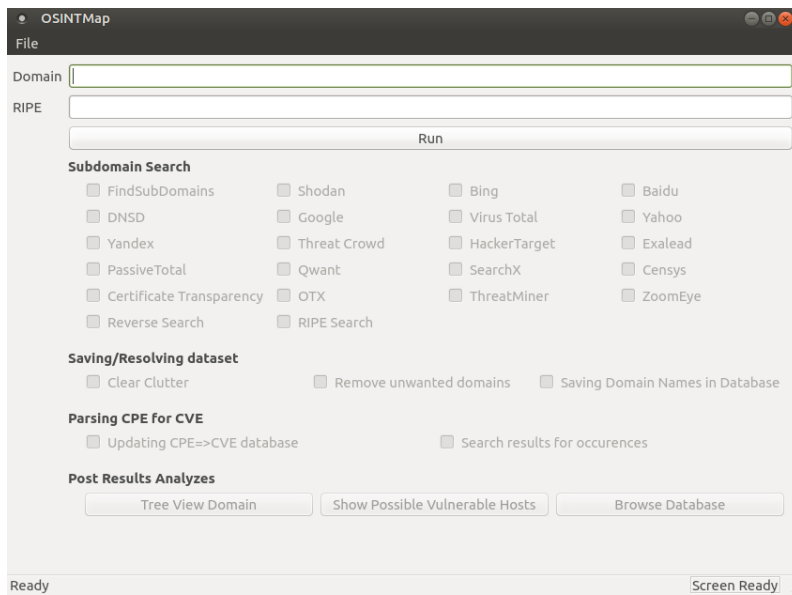


Figure 3.3 Application developed - Graphical

In the main screen, it is observed the search progress of the Application: what feeds were contacted and what time (in seconds) elapsed during the search of said feed. It is also possible to perform a RIPE search for allocated IP Address space.

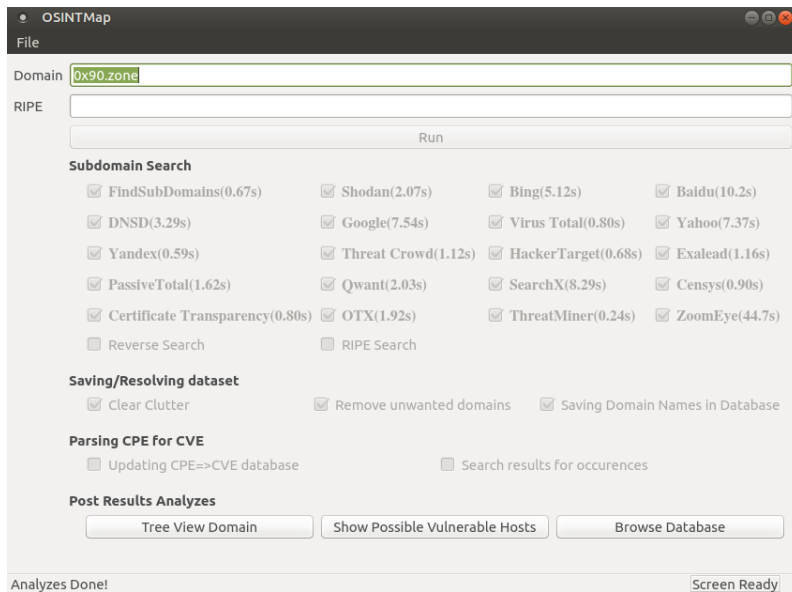


Figure 3.4 Application developed - Graphical - Scan

After the search is finished, post scan options appear on the screen, as seen in figure 3.4. In the following order, a tree view of the domain is shown as in figure 3.5.

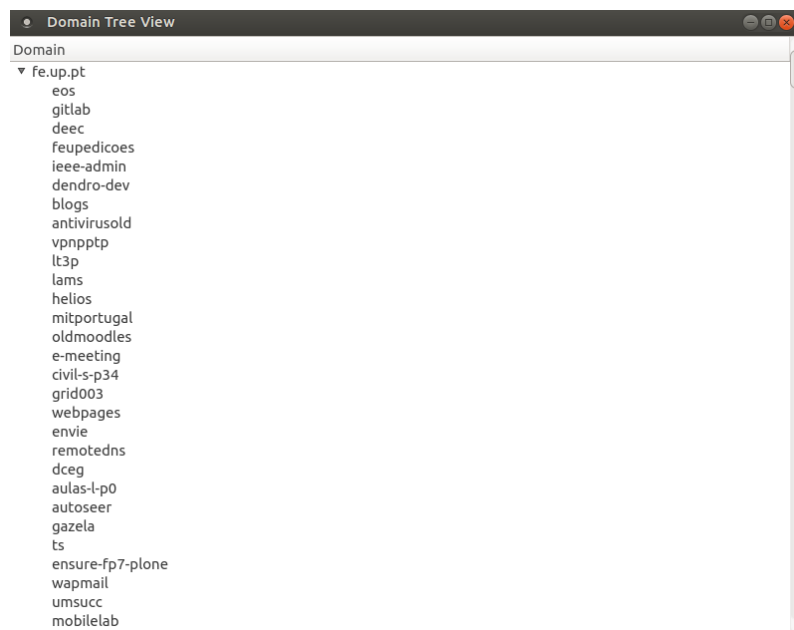


Figure 3.5 Application developed - Graphical - Domain Tree

Next Button is the CPE-CVE correlation. By pressing this button a search of all CPEs will be done with the online feed and presented, by product and increasing risk. The risk is assessed in a similar way to OpenVAS processing. A risk factor is determinate by the highest of the Base CVE Score of the vulnerability to each IP address. This metric makes sense since, by comparison, "a chain is only as strong as it is the weakest link". The order helps determine what should be prioritized to be checked.

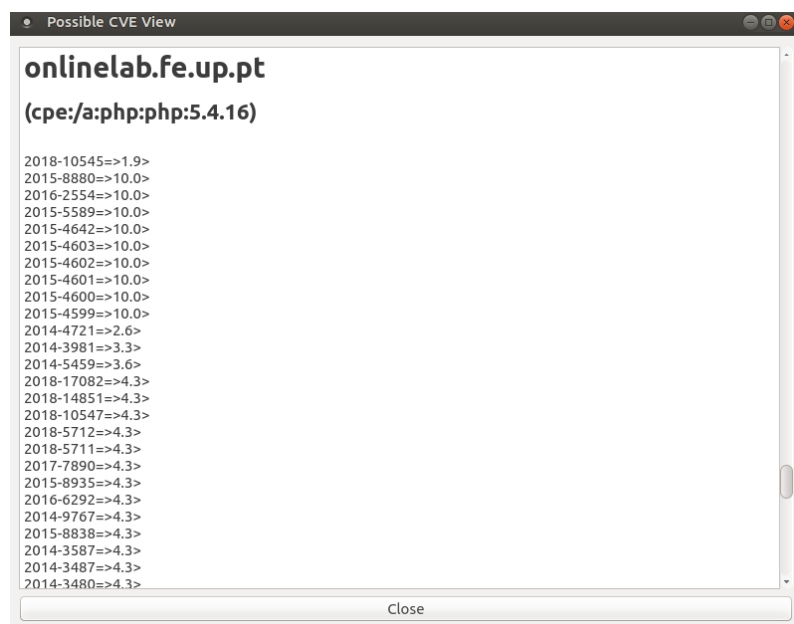
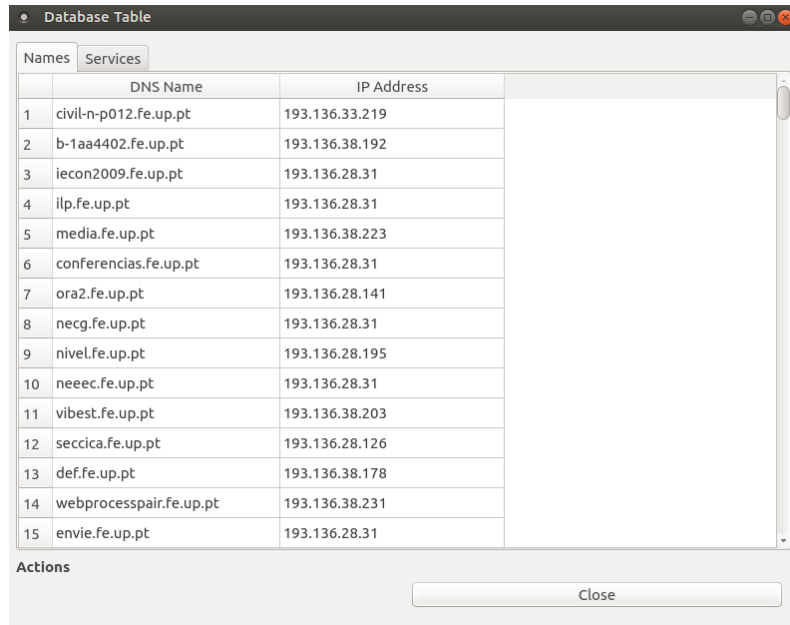


Figure 3.6 Application developed - Graphical - CPE/CVE correlation

The last button will display the information on the SQLite database, the services and the names discovered. This database usage is only to be a quick way to look into the database if the user opt not to install SQLite Browser.

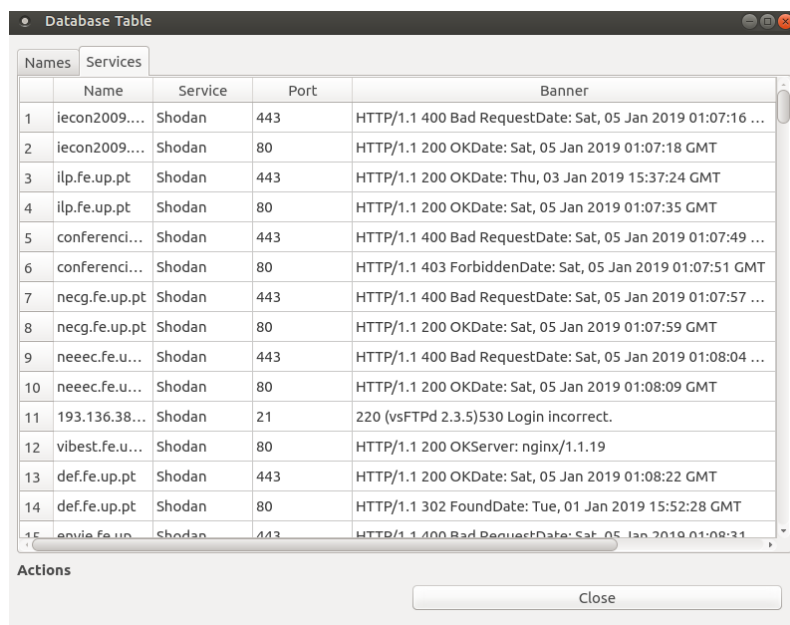


The screenshot shows a window titled "Database Table" with two tabs: "Names" and "Services". The "Names" tab is active, displaying a table with the following data:

	DNS Name	IP Address
1	civil-n-p012.fe.up.pt	193.136.33.219
2	b-1aa4402.fe.up.pt	193.136.38.192
3	iecon2009.fe.up.pt	193.136.28.31
4	ilp.fe.up.pt	193.136.28.31
5	media.fe.up.pt	193.136.38.223
6	conferencias.fe.up.pt	193.136.28.31
7	ora2.fe.up.pt	193.136.28.141
8	necg.fe.up.pt	193.136.28.31
9	nivel.fe.up.pt	193.136.28.195
10	neec.fe.up.pt	193.136.28.31
11	vibest.fe.up.pt	193.136.38.203
12	seccica.fe.up.pt	193.136.28.126
13	def.fe.up.pt	193.136.38.178
14	webprocesspair.fe.up.pt	193.136.38.231
15	envie.fe.up.pt	193.136.28.31

At the bottom of the window, there is an "Actions" section with a "Close" button.

Figure 3.7 Application developed - Graphical - Names Database



The screenshot shows the same "Database Table" window, but with the "Services" tab active. The table displays the following data:

	Name	Service	Port	Banner
1	iecon2009....	Shodan	443	HTTP/1.1 400 Bad RequestDate: Sat, 05 Jan 2019 01:07:16 ...
2	iecon2009....	Shodan	80	HTTP/1.1 200 OKDate: Sat, 05 Jan 2019 01:07:18 GMT
3	ilp.fe.up.pt	Shodan	443	HTTP/1.1 200 OKDate: Thu, 03 Jan 2019 15:37:24 GMT
4	ilp.fe.up.pt	Shodan	80	HTTP/1.1 200 OKDate: Sat, 05 Jan 2019 01:07:35 GMT
5	conferenci...	Shodan	443	HTTP/1.1 400 Bad RequestDate: Sat, 05 Jan 2019 01:07:49 ...
6	conferenci...	Shodan	80	HTTP/1.1 403 ForbiddenDate: Sat, 05 Jan 2019 01:07:51 GMT
7	necg.fe.up.pt	Shodan	443	HTTP/1.1 400 Bad RequestDate: Sat, 05 Jan 2019 01:07:57 ...
8	necg.fe.up.pt	Shodan	80	HTTP/1.1 200 OKDate: Sat, 05 Jan 2019 01:07:59 GMT
9	neec.fe.u...	Shodan	443	HTTP/1.1 400 Bad RequestDate: Sat, 05 Jan 2019 01:08:04 ...
10	neec.fe.u...	Shodan	80	HTTP/1.1 200 OKDate: Sat, 05 Jan 2019 01:08:09 GMT
11	193.136.38...	Shodan	21	220 (vsFTPD 2.3.5)530 Login incorrect.
12	vibest.fe.u...	Shodan	80	HTTP/1.1 200 OKServer: nginx/1.1.19
13	def.fe.up.pt	Shodan	443	HTTP/1.1 200 OKDate: Sat, 05 Jan 2019 01:08:22 GMT
14	def.fe.up.pt	Shodan	80	HTTP/1.1 302 FoundDate: Tue, 01 Jan 2019 15:52:28 GMT
15	envie.fe.up...	Shodan	443	HTTP/1.1 400 Bad RequestDate: Sat, 05 Jan 2019 01:08:31 ...

At the bottom of the window, there is an "Actions" section with a "Close" button.

Figure 3.8 Application developed - Graphical - Services Database

The main inspiration for this program was the Sparta Scanner [108]. This small program is essentially a wrapper of popular tools such as Nmap and Nikto [109]. This tool helps in the automation of the scanning process with simple tasks for example: perform a Nmap network

scanning and given the enumeration of the services launch tools against those services such as the Simple Network Management Protocol (SNMP) [110], HTTP(S), X11 [111] and others. This application is also considered an active scanner. Therefore, it contacts the servers directly and can even brute-force some usernames and passwords, therefore, it might help on a later stage of the engagement, but it is not the focus of this dissertation.

Sparta manages the scans in a similar way to the proposed pipeline. Although the pipeline requests information from several sources and then requires information about the services running from such DNS and IP records correspondingly. Then it requires the information about the services gathered, either with CPE provided from Shodan or with a local database that will search for strings in a given banner and identify that service, with the NIST database to enumerate potential vulnerabilities. Finally, it can present the operator with the information gathered.

To be able to parse the information a structure was designed. Since the platform was composed by bots that are specialized in one task, a bot was designed that will invoke the process and wait for results.

For each line of the results started by '#', it will decompose the string using a pre-defined character. In this case, it was chosen the pipe character ('|') since it was not allowed in the results. It will decompose the string in 4 structures. The first value of the tuple is the address of the host running the service. The second value is a list of CVE-IDs and their score such as (CVE-ID, CVE Score). The third value is the Max CVE Base Score value of the previous list. This value is used to classify the risk of the asset. Finally, the fourth value is the CPE that generated the initial search so the running service can be pinpoint at the host and dealt accordingly.

With this structure, the values are sorted in ascending order of severity to be easily manageable and prioritized.

Other options are available, for example, performing a reverse lookup to try to detect other domains associated with the IP. However, this is an option to aid a human to analyze since an enterprise may have servers on the cloud that could flag, for example, "azure.com" part of the scope. Therefore an automatic scanning will not be possible. An improvement may have a blacklist of domains to skip while doing the scan.

The way to print information is the standard for the connection to other tools, the information is printed on a per-line basis. For example, Nmap with the use of the "-iL" flag can accept a file with the IPs and hosts to scan for, due to the way Nmap was built and the way bash (Bourne again shell) allows for piping commands, it can accept a redirect of the standard output to Nmap by using '-' as the argument for the flag '-iL', thus '-iL -'.

Other tools were requiring additional parsing such as OpenVAS. In the web console, the user can select a file of hosts. However within the command line, using the OMP client the user needs to specify an XML entity [112]. Since the base output is simple it is easy to create a connector.

In the Metasploit framework it is also possible to set the "RHOSTS" option to point to a file and run the scan or exploit. Metasploit also can execute scans as a per command issued (not using the framework directly) [113] by using the Metasploit Command Line Interface (msfcli) interface. This option is beneficial since every result will be stored in the Metasploit framework

local database and allows for the interconnectivity of other tools. This interoperability could help create a System as a Service scanner to help maintain the security of large enterprises, performing automatic scans of hosts, services, and vulnerabilities. Metasploit Pro even has a REpresentational State Transfer (REST) [114] API to be able to accept those hosts and then scan them.

In figure 3.9, it is observed a perspective of what was intended to achieve for the rest of the pipeline.

Due to the simplicity of the output, its possible to, even with a simple command, pipe the information to other tools and therefore continue the pipeline. If for instance, an application needs to access historical data (to create a dashboard of the evolution of the exposure) it is also possible since the application stores all the results in the \$HOME/.osintmap folder of the running user. For the dashboard to check for progress or errors, a log file is also created.

3.2.8 Limitations of the application

As with all the developed applications, limitations are sure to be recognized. After some preliminary tests some limitations of this approach should be assumed. The biggest one is the dependency of external APIs. These APIs tend to change over time and require constant maintenance for that case. For instance, the findsubdomains API suddenly turned a paid subscription that would not be affordable. Without the subscription, the limit of 50 results is applied. That is the reason to have multiple APIs for the possibility of one being locked out. Others can try to compensate for the failure of one API. Other APIs appear on the Internet that could help in the discovery process such as Security Trails and Binary Edge, limited to 50 queries per month in the free plan but should be referred. These are examples that this application should be continuously maintained to adapt to new trends, but that should be expected to improve results.

Another limitation is the appearance of new online port scanners that should be added. In this dissertation, there are shown a few, but they only were implemented two. They are quite similar but not all show the CPE information needed to perform the vulnerability assessment, but it could help scan different ports and a list of available services. Since one of the expected results were to determine what kind of services the host is running the only way is to determine the service using only banner grabbing, but that is not always possible. Implementing a local correlation in the application to use a local database and determine the CPE could improve results, but some services versions only show if a particular command is issued, therefore a more direct contact with the service is needed. That would be out of the scope of this dissertation but should be considered as an improvement and future work.

As stated before, not all APIs return and might not display all the available host records to the client. These results are due to several reasons such as DNS records not being accessible to the Internet, passive analyze impossible (DNS request and response not available). Therefore although this helps systems administrators to discover unwanted exposed information, it does not guarantee that there are not other exposed services and hosts, and a more thorough analysis should be performed.

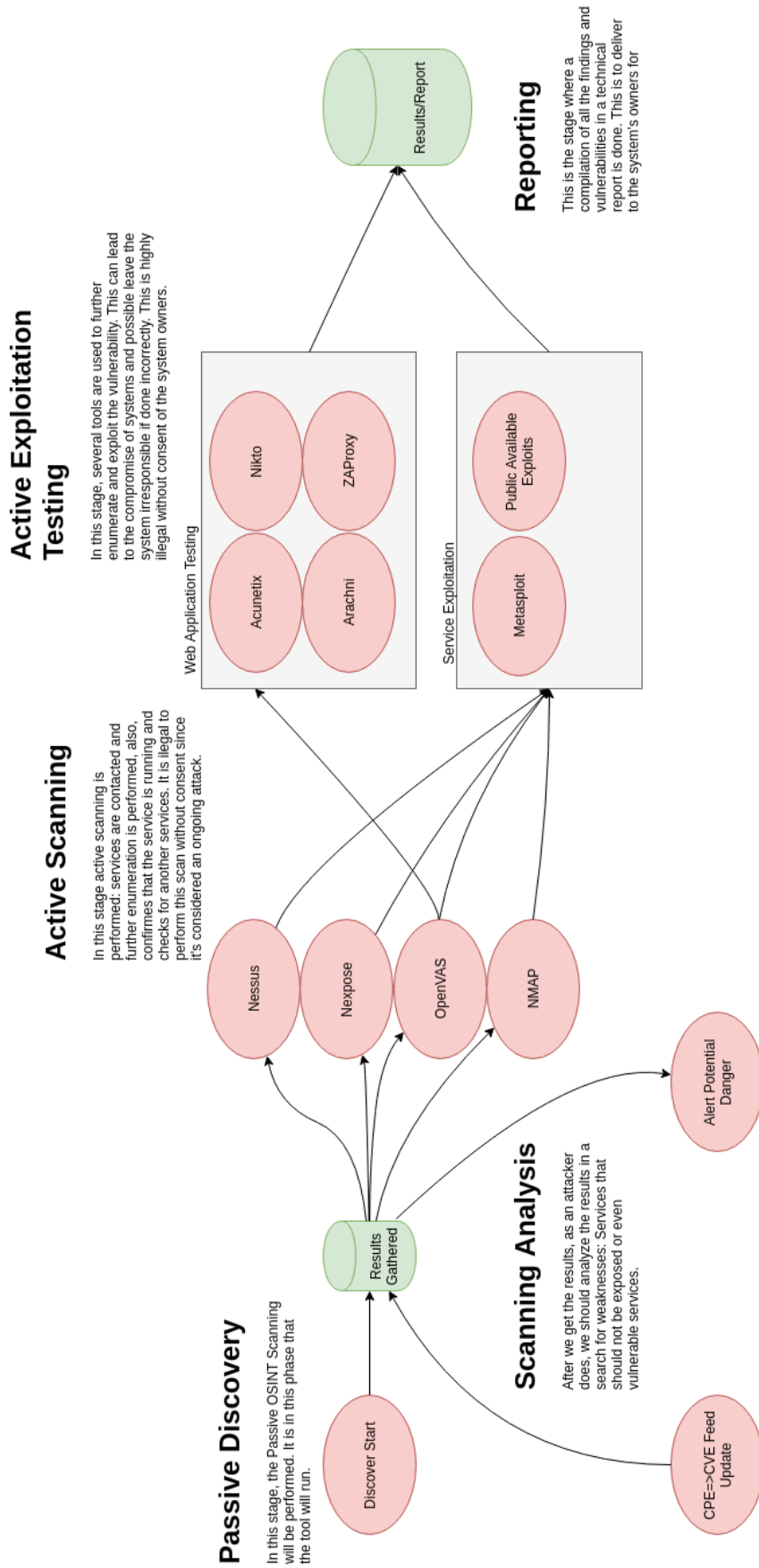


Figure 3.9 Process Flow of the proposed solution

Another significant limitation is the lack of support for IPv6. IPv6 is being deployed widely in several countries due to the shortage of IPv4 addresses. The entire IP addresses of IPv6 will suffice for several years (or decades) allowing the use of even more devices such as Internet of Things (IoT) devices to have a unique IP address. The developed application does not check for IPv6 addresses since it is still challenging to assure IPv6 connectivity on the testing bench of this application. There are alternatives to achieve connectivity such as VPNs, Teredo Tunneling (IPv6 encapsulated in IPv4 packets for exchange routers) [115] and some proxies, but they were not tested due to time constraints.

As stated before, the identification of vulnerabilities is also a concern and a limitation of this tool. Even though it is possible to detect the service and the version of some services, others will remain undiscovered either by using a different port or because it needs a specific interaction with the service. This limitation should be mitigated using in the next step of the process where indeed interaction with the service and all the 65536 ports should be scanned and check for service and vulnerabilities associated with it. For instance, some ports are not shown in Shodan, and some unusual ports are not detected as HTTP service since they are not on the standard expected ports. This limitation could be addressed either by improvements of the Shodan platform or in the next phase of the engagement. However, it is a concern that every user should be aware of it.

The attribution of the metric to each vulnerability can also be seen as a limitation of the tool in the sense that this process is still using the CVSS V2 Base Scoring System. The CVSS V3 Scoring System takes into account the context of the flaw. For example, a score of a vulnerability changes with the forest of services available in a more practical example is a vulnerable FTP service where files can be uploaded, but there is no way to execute them vs. an FTP service that hosts files to a Web Application, and on the Web application the malicious files are executed. This risk classification changes the perspective of an attacker and a defensor in the way that priority changes from one to another. This could also be mitigated at some extent by using the available services information in the specific machine, but understanding the logic relationship without a more extensive test such as, if the machines have some dependencies between another and if there are mitigations for the available exploits is difficult without prior knowledge of the network. For example, an Structured Query Language (SQL) injection may be available, but there are Web Application Firewalls (WAF) that would block the desired attack, they may be circumvented, but they lower the risk. In an additional note, some older vulnerabilities do not have a CVSS V3 score assigned to them, that is the reason CVSS V2 was chosen since it is more consistent. The same methodology OpenVAS uses to classify hosts was also chosen: The value of the danger a host has is the highest score of the vulnerability. Following the saying: "The chain is as strong as the weakest link". Although taking this literally may be exaggerated since, as described later on the results, the use of vulnerable functions may not be presented in the developed application, and thus the risk as is may not exist, but indicates a lack of security update policy.

During the development of this application, several tests were designed to test the approach continuously. The first test was designed to ensure that the information extracted from the feeds was accurate and every feed is parsable in a way that contributes results to the resulting set.

This set was then matched to check if the values obtained belong to the same domain. This is to avoid the check of out of scope targets that, although legal, takes time and slows the desired search.

This set was then checked for resolution. This means that for each result an IP address should be gathered to avoid outdated hosts to be contacted. This helps on the later phase of the search when the host is passed to the services check feeds where the host will be check for services running and if the resolution fails, it will cost time and will not produce the desired results.

For each of the data sources added these checks were run. It was also checked for their consistency, for example, if the ending result of this phase if a host is marked with a port tcp/80 open, a manual check was made as well as every CPE detected. The results showed that almost one host for every 1000 tested had a service closed instead of opened. Since these are public databases and the scan is not on demand. It is possible that some services that changed recently are still shown as vulnerable. It is also possible that the enterprise could badly block some IP addresses to accessing it, for instance, Portugal IP addresses are blocked, however, Russian IP addresses are allowed.

In the last step, where CPE was correlated, it was found that the CPE was correctly assigned. This means that the service when it was contacted, assigned a CPE accordingly.

In conclusion, the results are reliable even with the use of public databases although considering that some host could have outdated information the information retrieved proves accurate in real-world tests.

3.2.9 Integration with IntelMQ

Following the developed application it is proposed a connector to deliver alerts to operators monitoring the security of companies. The adequate application to do this is IntelMQ since it is already deployed in several CSIRTs and, since it works with several feeds to aggregate information, a direct integration with this work could be seen.

To be precise, there are two ways to achieve this integration: integrate the already established application; or create a new workflow entirely on the IntelMQ platform to work with the same results.

The first option will rely on the produced code to output the results to IntelMQ in a way that it will generate alerts with all the information required.

The second option is to port every feed code to a bot in IntelMQ that will send the information to a queue that will then enumerate the services and their CPEs, and then send to a queue to lookup for CVEs associated with the CPEs found. Finally, the result operation will send the results to an output bot to trigger the alerts. In essence, implementing what was proposed in 2.1.

Due to time constraints and the already established workflow it was chosen to directly parse the results in a Parser bot in IntelMQ interface. This scheme does not only allow for IntelMQ call the developed application, but it can also demonstrate how a bot can call external applications to create alerts on the application. Porting the feeds to the application also work since the developed

application works with the same engine (Python 3) of the IntelMQ, however, this would take time to test.

As stated before, a parser bot was developed that calls the developed terminal application and waits for results. For this bot to work properly, some options were defined as shown in 3.10. The two main options are: the target (the domain for the search); and the retry value (time to wait since one result is obtained and a new search is started).

The screenshot shows the 'Edit Node' configuration interface for the 'OSINTMAP-Domain-Parser' bot. The configuration is organized into sections: 'generic', 'runtime', and 'default'. The 'generic' section includes fields for 'id' (OSINTMAP-Domain-Parser), 'name' (OSINTMAP Domain), 'group' (Parser), 'module' (intelmq.bots.parsers.osintmap.parse_vulns), and 'description' (OSINTMAP Get Vulnerabilities obtained by OSINT). The 'runtime' section includes 'enabled' (true), 'run_mode' (continuous), 'name' (up.pt), and 'rate_limit' (10800). The 'default' section is partially visible at the bottom.

Figure 3.10 IntelMQ - Bot configuration

The logger developed in the application will still work as expected to debug results, but it also works with the logger of IntelMQ to output status information about the process.

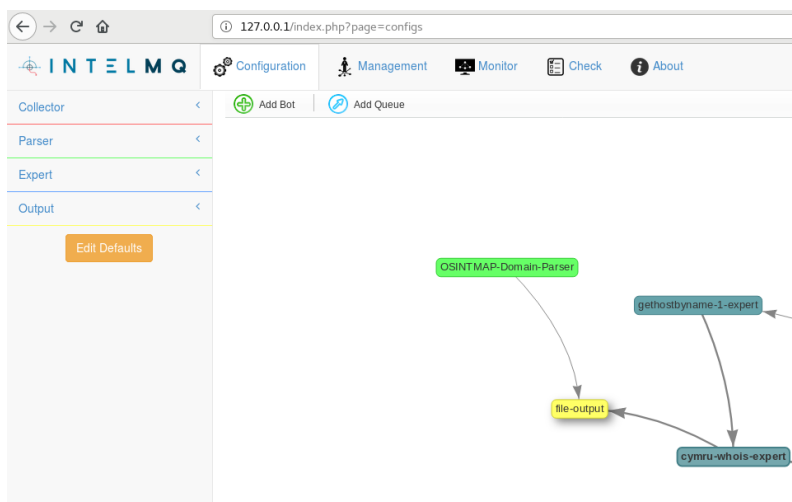


Figure 3.11 IntelMQ - Dashboard

To test the bot, the results are sent to a queue that will output to a file as 3.11 shows.

```

6.4'), ('2017-9788', '6.4'), ('2018-1312', '6.8'), ('2017-15715', '6.8')][6.8][cpe:/a:apache:ht
p_server"]
{"classification.taxonomy": "information gathering", "classification.type": "scanner", "extra.dat
a": "fado", "extra.data": [{"CVE": "2018-1283", "CVSS": "3.5"}, {"CVE": "2018-1302", "CVSS": "4.3"}, {"CVE": "2018-1301", "CVSS": "4.3"}, {"CVE": "2018-1333", "CVSS": "5.0"}, {"CVE": "2018-1303", "CVSS": "5.0"}, {"CVE": "2017-15710", "CVSS": "5.0"}, {"CVE": "2018-1312", "CVSS": "6.8"}, {"CVE": "2017-15715", "CVSS": "6.8"}][6.8][cpe:/a:apache:http_server:2.4.29"]
{"classification.taxonomy": "information gathering", "classification.type": "scanner", "extra.dat
a": "fado", "extra.data": [{"CVE": "2016-8612", "CVSS": "3.3"}, {"CVE": "2018-1283", "CVSS": "3.5"}, {"CVE": "2016-4975", "CVSS": "4.3"}, {"CVE": "2018-1302", "CVSS": "4.3"}, {"CVE": "2018-1301", "CVSS": "4.3"}, {"CVE": "2015-3185", "CVSS": "4.3"}, {"CVE": "2018-1303", "CVSS": "5.0"}, {"CVE": "2017-15710", "CVSS": "5.0"}, {"CVE": "2017-9798", "CVSS": "5.0"}, {"CVE": "2016-8743", "CVSS": "5.0"}, {"CVE": "2016-2161", "CVSS": "5.0"}, {"CVE": "2016-0736", "CVSS": "5.0"}, {"CVE": "2015-3184", "CVSS": "5.0"}, {"CVE": "2015-3183", "CVSS": "5.0"}, {"CVE": "2015-0228", "CVSS": "5.0"}, {"CVE": "2016-5387", "CVSS": "5.1"}, {"CVE": "2017-9788", "CVSS": "6.4"}, {"CVE": "2018-1312", "CVSS": "6.8"}, {"CVE": "2017-15715", "CVSS": "6.8"}, {"CVE": "2017-7679", "CVSS": "7.5"}][6.8][cpe:/a:apache:http_server:2.4.6"]
{"classification.taxonomy": "information gathering", "classification.type": "scanner", "extra.dat
a": "www", "extra.data": [{"CVE": "2016-8612", "CVSS": "3.3"}, {"CVE": "2018-1283", "CVSS": "3.5"}, {"CVE": "2014-3250", "CVSS": "4.0"}, {"CVE": "2016-4975", "CVSS": "4.3"}, {"CVE": "2018-1302", "CVSS": "4.3"}, {"CVE": "2018-1301", "CVSS": "4.3"}, {"CVE": "2018-8011", "CVSS": "5.0"}, {"CVE": "2018-1333", "CVSS": "5.0"}, {"CVE": "2018-1303", "CVSS": "5.0"}, {"CVE": "2017-15710", "CVSS": "5.0"}, {"CVE": "2017-9798", "CVSS": "5.0"}, {"CVE": "2016-8743", "CVSS": "5.0"}, {"CVE": "2016-2161", "CVSS": "5.0"}, {"CVE": "2016-0736", "CVSS": "5.0"}, {"CVE": "2017-7659", "CVSS": "5.0"}, {"CVE": "2017-9789", "CVSS": "5.0"}, {"CVE": "2017-12171", "CVSS": "6.4"}, {"CVE": "2017-9788", "CVSS": "6.4"}, {"CVE": "2018-1312", "CVSS": "6.8"}, {"CVE": "2017-15715", "CVSS": "6.8"}][6.8][cpe:/a:apache:http_server"]
{"classification.taxonomy": "information gathering", "classification.type": "scanner", "extra.dat
a": "www", "extra.data": [{"CVE": "2016-8612", "CVSS": "3.3"}, {"CVE": "2018-1283", "CVSS": "3.5"}, {"CVE": "2014-3250", "CVSS": "4.0"}, {"CVE": "2016-4975", "CVSS": "4.3"}, {"CVE": "2018-1302", "CVSS": "4.3"}, {"CVE": "2018-1301", "CVSS": "4.3"}, {"CVE": "2018-8011", "CVSS": "5.0"}, {"CVE": "2018-1333", "CVSS": "5.0"}, {"CVE": "2018-1303", "CVSS": "5.0"}, {"CVE": "2017-15710", "CVSS": "5.0"}, {"CVE": "2017-9798", "CVSS": "5.0"}, {"CVE": "2016-8743", "CVSS": "5.0"}, {"CVE": "2016-2161", "CVSS": "5.0"}, {"CVE": "2016-0736", "CVSS": "5.0"}, {"CVE": "2017-7659", "CVSS": "5.0"}, {"CVE": "2017-9789", "CVSS": "5.0"}, {"CVE": "2017-12171", "CVSS": "6.4"}, {"CVE": "2017-9788", "CVSS": "6.4"}, {"CVE": "2018-1312", "CVSS": "6.8"}, {"CVE": "2017-15715", "CVSS": "6.8"}][6.8][cpe:/a:apache:http_server"]

```

Figure 3.12 IntelMQ - Bot Output

The bot will only alert for items that have CVEs associated. It is possible to see the bot output in IntelMQ Web interface in figure 3.12.

Figure 3.12 requires some additional explanation regarding the expressed data. The value "classification.taxonomy" is the classification of the information according to [116] and [117]. The value "information gathering" is used because that is the nature of the information, to collect information. The following field "classification.type" is set accordingly with the referred sources. The field "extra.data" is where all the information with the format already described in 3.2.7.2.

For a more in-depth analysis, the direct usage of the tool is recommended since only assets with a positive CPE enumeration are parsed to reduce false positives.

In conclusion, the tool helps automate some procedures, but it is not an all in one solution that will help detect and prevent every situation possible to the target network. However, it helps to understand what should be analyzed and pass it to the next phase of the engagement.

Chapter 4

Use cases and results

In this chapter, it will be presented the real world results of the application on some networks. These networks are considered critical and should be monitored to ensure that an attack is not undergoing. For each network, it will be exposed the services and versions that were detected, reported and fixed. The names and addresses will be obfuscated in order to avoid damage to the institution in question.

In the development of the procedure, several networks were analyzed. In this chapter three networks were chosen to demonstrate real-world scenarios of the usability of the procedure. These networks were chosen as an example, since they all host critical services in their underlining infrastructures and it is reasonable to consider them critical to the infrastructure of the country.

4.1 Analysis of a University Network

The first network that is going to be analyzed is a typical university network. Universities networks are considered critical infrastructure since they are the foundation of science and progress of the country. Relevant research is done in these institutions. If research is leaked, profit could be lost and hurt not only the institution but every public system that rely on that profit to keep the academic world accessible to every student.

A privacy issue arises in these networks. Not all the infrastructure is critical due to the fact of the intellectual property, but since all the private and confidential data of every student, professor, and assistant is store in these networks and if leaked poses a risk to all members.

This university has a security management team that could handle vulnerability discloses, meaning that they have a CSIRT "indoors" that can handle and process all incidents.

Looking at the results, it is observed that the running versions of the services do not stray too away as seen on some networks. The reason is mainly that this particular university, with their centralized management, tries to consolidate all services in a manageable way.

There are however some discrepancies that allow for a network breach to occur. It appears that some of the faculties in the University do not upgrade their systems too often. Some obsolete services should not be active and represent a considerable danger to the institution. Some of these

services are known and may have been reported in the past. However, due to the complexity of the institution organization, the mitigations may take time to be implemented. For example, sites like zone-H.org [118] may already have reported compromises in the past. However, they may appear again in the future since the fix may not be applied correctly. Another example is the usage of old Outlook Web Access (2007). Although this version is still widely used, people are encouraged to update it to a more modern (and secure) version.

In this institution, there where also evidence of DNS misconfigurations where public facing IP addresses are resolved to a private one accordingly with the Request For Comments (RFC) 1918 [119]. The external DNS server or the DNS view to that service should not allow this since it leaks internal IP addresses that are not Internet routable. However one can understand why this is the case. In the internal networks if a computer uses a DNS server resolver other than the internal DNS server it cannot resolve such IPs internally and will lead to connectivity issues, thus need to resolve external requests to internal IPs. This service organization is considered bad practice since every computer within the institutions should use the internal infrastructure and not have multiple reliability on external services. There are also extensions available to override DNS requests to other DNS servers and redirect to the internal DNS server [120]. This network organization would be beneficial for Digital Forensics Incident Response (DFIR) and troubleshooting since it would be structured and easily controllable in a centralized manner, meaning that detecting if a compromised host that is contacting a known malicious domain becomes an easier task. For instance, if there is a DNS cache somewhere and the server updates the A record (resolution of an address accordingly with the RFC 1035 [121]) the cache could take some time to update itself and in troubleshooting this could become arduous. Other security concerns such as privacy arise since the DNS request is intercepted, the operator can obtain the address that the computer is trying to access.

www.google.com	1.1.1.1	November 07, 2018 04:03AM	November 07, 2018 04:03AM
www.google.com	10.5.0.1	November 07, 2018 03:18AM	November 07, 2018 03:18AM
www.google.com	10.163.0.1	November 07, 2018 03:41AM	November 07, 2018 03:41AM
www.google.com	172.30.12.59	November 07, 2018 02:40AM	November 07, 2018 02:40AM
www.google.com	172.30.12.50	November 07, 2018 02:43AM	November 07, 2018 02:43AM
www.google.com	172.30.12.46	November 07, 2018 03:04AM	November 07, 2018 03:04AM
www.google.com	172.30.12.40	November 07, 2018 03:06AM	November 07, 2018 03:06AM
www.google.com	172.30.12.52	November 07, 2018 03:22AM	November 07, 2018 03:22AM
www.google.com	172.30.12.33	November 07, 2018 03:27AM	November 07, 2018 03:27AM
www.google.com	172.30.12.64	November 07, 2018 03:34AM	November 07, 2018 03:34AM

Figure 4.1 DNS Misconfiguration on an University Network

Continuing the analysis, it appears that some unrecommended services are available to the whole Internet. Services such as development platforms where the security rules are more permissive should be contained in the developed network only, and not available to the Internet. Such systems should be accessible through an encrypted VPN.

digitdev.ugqpt	193.104.27.85	November 07, 2018 02:36AM
sridev.ugqpt	193.104.27.85	November 07, 2018 02:49AM
signserver-eptstdev.ugqpt	193.104.27.85	November 07, 2018 02:55AM
atomdev.ugqpt	193.104.27.85	November 07, 2018 02:55AM
signserver.ugqpt	193.104.27.85	November 07, 2018 02:57AM
signserver-dev.ugqpt	193.104.27.85	November 07, 2018 02:58AM
dendro-dev.ugqpt	193.104.27.85	November 07, 2018 03:03AM
wpdev.ugqpt	193.104.27.85	November 07, 2018 03:09AM
dev.ugqpt	193.104.27.85	November 07, 2018 03:17AM

Figure 4.2 Exposed Development environment on an University Network

Starting the port analysis, many FTP services are exposed. This service, without extensions, communicates using unsecured channels and an attacker listening to the medium can obtain all the information transmitted (passwords; files transferred and others). The Shodan banner grabbing also tests for some accounts (usually the anonymous login), and some of them were successfully however some services are intended to be open (such as mirrors services that allow people to download free software), but that might not be the case on all the results. However, to confirm that vulnerability, direct access to the service needs to be made.

193.104.27.85	Shodan	21	220----- Welcome to Pure-FTPD [privs...	[u'cpe:/a:pureftpd:pure-ftpd']
193.104.27.85	Shodan	21	220----- Welcome to Pure-FTPD [privs...	[u'cpe:/a:pureftpd:pure-ftpd']
193.104.27.85	Shodan	21	220----- Welcome to Pure-FTPD [privs...	[u'cpe:/a:pureftpd:pure-ftpd']
193.104.27.85	Shodan	21	220----- Welcome to Pure-FTPD [privs...	[u'cpe:/a:pureftpd:pure-ftpd']
193.104.27.85	Shodan	21	220 ProFTPD 1.3.5e Server (Sentora FTP ...	[u'cpe:/a:proftpd:proftpd:1.3.5e']
193.104.27.85	Shodan	21	220 ProFTPD 1.3.5b Server (ProFTPD) [1...	[u'cpe:/a:proftpd:proftpd:1.3.5b']
193.104.27.85	Shodan	21	220 ProFTPD 1.3.1 Server (ProFTPD Defa...	[u'cpe:/a:proftpd:proftpd:1.3.1']
193.104.27.85	Shodan	21	220 ProFTPD Server (FTPD [REDACTED]) [192.1...	

Figure 4.3 Exposed FTP Services on an University Network

There is also an instance of PortMap [122] found. This service works like the yellow pages (registering services) thus helping attacker map out the attack.

At the beginning of this dissertation, it was stated that the application should not only search for registered addresses but also names that are on the cloud. Since port tcp/445 - Server Message Block (SMB) [123] service was blocked at the perimeter firewall (due to mitigations against WannaCry attack), finding a host exposing that port should raise some suspicions. A host in sharing hosting with the port tcp/445 opened and with SMB authentication disabled was found on the results. Even though it appears that it does not have any useful data further exploration was not attempted.

Microsoft Point-to-Point Tunnel Protocol (PPTP) [124] was also found in the result set, which should be avoided since it is an obsolete VPN protocol since even the most strong encryption,

using MS-CHAPv2, can be reduced to a 56-bit key of entropy using the DES algorithm that can be quickly broken by any modern computer [125].

Databases, MySQL/MariaDB, were also found in the set. Usually, these are segregated in internal networks since they may contain sensitive information such as credentials or confidential information. The versions used also are not up to date and have already critical CVE associated with it.

██████████	Shodan	3306	5.1.73
██████████	Shodan	3306	5.6.41

Figure 4.4 Exposed MySQL/MariaDB Services on an University Network

PostgreSQL was also found in the results. This database engine is an object-relation database and, for the same reasons as the MySQL/MariaDB should be in internal networks.

Similar to RDP, Virtual Network Computing (VNC) is a protocol that allows for remote interaction with the machine despite of the fact that the connection can be sniffed and the password can be extracted more easily than the RDP, if no SSL/TLS tunnel is made prior to the connection. RDP uses a challenge-response mechanism (in the default configuration) that needs to be cracked to obtain the password, VNC sends the password in clear-text. One instance was found using this protocol. Using these protocols to remotely administer machines are not recommended due to lack of confidentiality on the connection.

An unknown service was also discovered running on a well-known port [126]. This service was running on port tcp/666 and was not following the well-known protocols either DOOM (DOOM server, the game) or MDQS (Multiple Device Queueing System). Even though this is not a direct vulnerability, it is often regarded as a bad practice to use low-level ports for other services than the ones from the RFC. Using this knowledge, the CSIRT was contacted to investigate the incident further. It appears that the service was an interface to a custom application that manages a critical service and should not be exposed to the outside world. The incident was mitigated with a firewall configuration to avoid exposure to the Internet.

A Dynamic Hash Tables (DHT) node was also found running in the University network. DHT act as routers to interchange information (files usually) between peers. This service does not represent a vulnerability itself, but it is odd to see a university running their DHT node in the servers (figure 4.5. DHT is associated with piracy, but legal usage is also reasonable.

██████████	Shodan	10400	DHT Nodes1 ██████████172.193...
------------	--------	-------	---------------------------------

Figure 4.5 DHT on an University Network

Looking back at the interactions made with this CSIRT it was observed a time delay in the correction of the incidents. Often taking weeks just to remove the asset exposure.

Some incidents are regarded as low severity and take a long process to be mitigated. Older technologies take time to migrate, and some do not even justify the time being the risk accepted.

The exposure of services that should not be exposed is mitigated promptly with adequate action. With this in mind, it is possible to extrapolate that the pipeline for the reports is long and requires refinement.

Statistically speaking, from the 26 reports made 12 were resolved and 14 are still "on hold". These values do not reflect the severity of systems since the flaws reported are considered high severity. The reports made did not include outdated versions nor expired certificates. This represents, alone, an 46% correction rate which is low and perhaps indicates that the institution in question should speed up the patching process.

Considering the size of the network analyzed, it is safe to say that the SOC operators need to monitor several systems

Considering the size of the network analyzed, it is safe to say that the SOC operators need to monitor several systems in order to ensure the security of the network. This is an herculean task. Therefore every tool that will help monitor these assets could reduce the risk of the enterprise. It is possible to see in table 4.1 that there were discovered 3167 unique services. The critical vulnerabilities that were found could be considered residual facing the number of exposed services. However, the same principle applies where a small risk vulnerability could lead to a complete enterprise takedown by the attacker. This procedure helps mitigate this risk by attempting to discover assets that are left out of the scope of the monitoring tools.

Discovered hosts	2426
Discovered services	3167
Critical vulnerabilities disclosed	26
Patched vulnerabilities (28-01-2019)	12

Table 4.1 Result Statistics of a University Network

4.2 Analysis of a Government Network

The second network to be analyzed is the governmental network. This network is considered a high-value network since many services that citizens depend on (i.e.: Internal Revenue Service (IRS) and social security websites allow citizens to pay their due taxes). For this services to work (IRS, and social security) there is a need to hold information about the citizen. Personal information such as name, address, and age is considered critical and when leaked can impact the life of the citizen in several ways. For example, with all the personal data it is possible to request a loan in that person's name. Although these institutions do not need to enforce GDPR at the moment (since they have prolonged the date to enforce it on these institutions), they need to be secure and available at all times to ensure the progress of the country. In case some of the data is compromised several problems could arise, depending on the nature of the data.

These critical networks are monitored continuously by "Centro Nacional de Cibersegurança" (CNCS) to ensure that everything remains operating under normal conditions. Any vulnerability

or attack will trigger an internal process where the CNCS could be asked to help by investigating and, in some cases, mitigating the vulnerability.

This risk means that this network (or collections of networks) should be secured and not prone to known vulnerabilities such as software with known vulnerabilities or services that should not be exposed as the best practices in computer networks states.

The fact is that the hosts discovered using the proposed methodology are quite troublesome. Several services detected, and their respective versions suffer from a high number of CVEs assigned to them. These CVEs range from low severity to high severity (accordingly with the Base Score of the CVE Score System Version 2).

This fact indicates that the monitoring capabilities could be improved to detect these problems and deploy mitigations by contacting the administrator of the correspondent network.

From a passive standpoint, it is theoretically possible that some services could be exploited by automated scripts given the service and version running on the hosts.

Even though no exploitation was attempted, some services were checked to confirm the correct service detection. For example services with a known PHP version and the Joomla Core vulnerability were checked to detect if it were vulnerable, but no exploitation was attempted.

This concept of the environment is vital since there should be other mitigations in place to block these attempts of exploitation. In other words, what is theoretically possible could not be practically possible due to those mitigations. This concept is known as the environment as described earlier in the CVSS score. This is an example where the CVSSv3 score could be beneficial if information about other technologies implemented could be enumerated.

Other interesting vulnerabilities that are highly dependent on the context of the PHP CVEs. Usually, vulnerabilities in this programming language are discovered by using specific functions of that language without appropriate user sanitation. For example, the "preg_replace" function allows for code execution if an unrestricted or lack of good sanitation is put in place of the replacement parameter. The usage of this function will trigger an "eval" call (function eval) that will execute the code passed to it [127], allowing to escape the script intentions. Finding this usage without access to the code can be difficult and therefore hard to mitigate something that one does not know that is there. However, the lack of PHP updates could indicate that the service is not maintained as much as the desired.

Of course, this methodology does not intend to exploit vulnerabilities but to detect indicators that could lead to them. For example, in this case, a lookup for the Web services with a higher number of CVEs assigned to them was made. Opening their pages to see if they have been updated or just have been forgotten. The results are quite interesting, not only were found outdated websites with vulnerabilities that are easily exploitable (Cross-Site Scripting -XSS-, SQL injection, RCE) but were also found compromised hosts that were left out of the scope of the monitoring scope.

One exciting finding was the use of revoked certificates. A certificate can be revoked in case of compromise of the chain of trust of it. The fact that the certificate is still being used poses a risk to clients and the organization, since a certificate is revoked by human decision when a certificate is compromised, therefore the certificate was indeed compromised, revoked but still left on some

systems. If a certificate was compromised and users are required to be forced to use it, an attacker can use the same certificate to attack clients. This reflects the bad practices in the sector, and lack of security compliance.

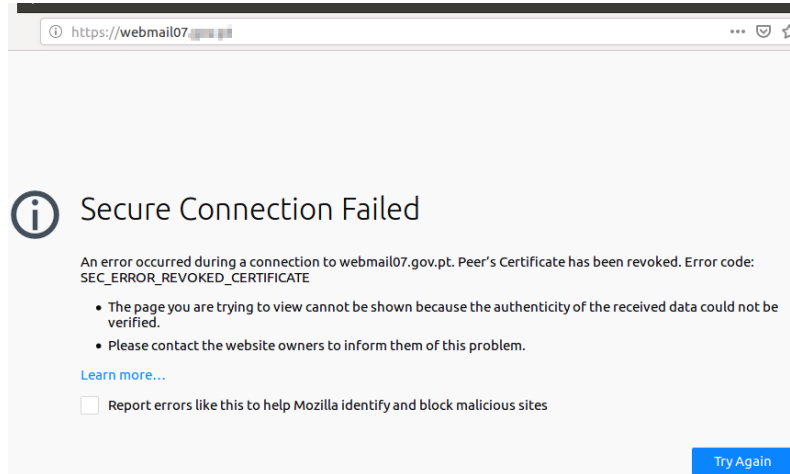


Figure 4.6 Revoked Certificate on a Government Network

Examining the discovered Services, it was immediately seen several FTP services opened. These services allow for the sharing of files but lack security in the connection. Mitigations can be used by using encryption. They do not seem to be implemented in the majority of the discovered services (some use it, but it is a small portion of the set).

10.10.10.10	Shodan	21	220 Microsoft FTP Service530 User cann...	[u'cpe:/o:microsoft:windows']
10.10.10.10	Shodan	21	220 Microsoft FTP Service530 User anon...	[u'cpe:/o:microsoft:windows']
10.10.10.10	Shodan	21	220 Microsoft FTP Service530 User cann...	[u'cpe:/o:microsoft:windows']
10.10.10.10	Shodan	21	220 Microsoft FTP Service530 User cann...	[u'cpe:/o:microsoft:windows']
10.10.10.10	Shodan	21	220 Microsoft FTP Service530 User cann...	[u'cpe:/o:microsoft:windows']
10.10.10.10	Shodan	21	220----- Welcome to Pure-FTPd [privs...	[u'cpe:/a:pureftpd:pure-ftpd']
10.10.10.10	Shodan	21	220----- Welcome to Pure-FTPd [privs...	[u'cpe:/a:pureftpd:pure-ftpd']
10.10.10.10	Shodan	21	220----- Welcome to Pure-FTPd [privs...	[u'cpe:/a:pureftpd:pure-ftpd']
10.10.10.10	Shodan	21	220----- Welcome to Pure-FTPd [privs...	[u'cpe:/a:pureftpd:pure-ftpd']

Figure 4.7 Exposed FTP on a Government Network

Another finding was the use of DNS recursive resolvers in the network. A DNS recursive resolver is useful for internal facing networks since they can act as cache servers and use it in a more centralized and controlled way within the enterprise. However, facing this functionality to the Internet could lead to severe problems like being used in a Distributed Reflexion Denial of Service (DrDOS) or Distributed Denial of Service (DDOS) attack. Therefore this services could lead to exploitation in the sense of "being used" in a coordinated attack.

192.168.1.100	Shodan	53	dnsmasq-2.47	Recursion: enabled
192.168.1.101	Shodan	53	dnsmasq-2.47	Recursion: enabled
192.168.1.102	Shodan	53	dnsmasq-2.47	Recursion: enabled
192.168.1.103	Shodan	53	dnsmasq-2.47	Recursion: enabled
192.168.1.104	Shodan	53	dnsmasq-2.47	Recursion: enabled
192.168.1.105	Shodan	53	dnsmasq-2.47	Recursion: enabled

Figure 4.8 DNS Recursion on a Government Network

PortMap is also found on the results. This service acts as yellow pages where services are registered to it in order to be easier to find and contact them. Due to that fact, by enumerating the list, it is possible to understand what the host is running giving a glimpse of information.

IP Address	Source	Port	Program	Version	Protocol	Port
192.168.1.100	Shodan	111	PortmapProgram	Version	Protocol	Port
192.168.1.101	Shodan	111	PortmapProgram	Version	Protocol	Port
192.168.1.102	Shodan	111	PortmapProgram	Version	Protocol	Port
192.168.1.103	Shodan	111	PortmapProgram	Version	Protocol	Port

Figure 4.9 Exposition of Portmapper on a Government Network

Lightweight Directory Access Protocol (LDAP) [128] was found in three hosts. Typically this service handles authentication, authorization, and accounting (AAA) of an enterprise and it is found on internal networks. The AAA allows for users to login on certain systems that they have access and allowing them to have an unique account that provides the necessary access controls to the enterprise keeping everything logged to detect malicious activity or illegal access. The only reason to be exposed is to give the ability to other remote clients to (i.e.: in trust relationships) however this should not be exposed to the whole Internet since it can be brute-forced and cause instability on the enterprise.

SMB is also found opened in specific ports. This service allows for remote management, file sharing, and resource sharing. It is typically found in Microsoft Operating Systems, but it is not exclusive (Linux implementations are available). This service is not recommended to be exposed to the Internet due to the plethora of attacks that can be made against it.

Microsoft PPTP VPN is also used in several hosts. This VPN service is considered obsolete and should be replaced by other, more secure and reliable, types of VPN services. This service can be brute-forced or even possible to steal credentials if an attacker intercepts out the connection made to the server due to low entropy of the algorithm that supports the secure connection. Different types of connection can be made, but even the most secure option lacks the cryptographic strength that would prevent malicious actors to listen to the connection, as described earlier.

Another exposed service found open is the MySQL/MariaDB database service. This service stores data (sometimes critical data) necessary to run services like Web applications. Usually, this service should only be present in internal networks or only allowed by specific IP addresses. The

compromise of said service through an exploit or brute-force compromises the integrity of the data in that database.

MySQL/MariaDB	Shodan	3306	5.6.41
MySQL/MariaDB	Shodan	3306	5.5.61-ctl
MySQL/MariaDB	Shodan	3306	\x04Host \'90.24.166.44\' is not allowed ...
MySQL/MariaDB	Shodan	3306	\x04Host \'213.42.156.142\' is not allowe...
MySQL/MariaDB	Shodan	3306	5.5.5-10.0.36-MariaDB
MySQL/MariaDB	Shodan	3306	5.6.41
MySQL/MariaDB	Shodan	3306	5.7.23
MySQL/MariaDB	Shodan	3306	5.1.73-1+deb6u1-log
MySQL/MariaDB	Shodan	3306	5.5.61-ctl
MySQL/MariaDB	Shodan	3306	5.7.24

Figure 4.10 Exposition of MySQL/MariaDB on a Government Network

RDP is also exposed in the set. This service is prone to brute-force attacks on the Internet since it allows to interact with the remote machine, copy files and remote mount partitions. This service should not be exposed to the Internet. It is found open in many companies so subcontractors can easily administer the machine without asking for access.

It was also possible to discover an Oracle WebLogic service vulnerable to deserialization [129] attack through the banner grabbing. As stated this was discovered through the analysis of the banner provided by the service since Shodan did not provide a CPE to it. This lack of detection could change in the future, but as of the time of writing, this was not the case. An attacker can compromise and gain command execution using very easily by exploiting this vulnerability.

One interesting finding was, through banner analysis the detection of HTTP Basic Authorization without SSL/TLS. Although this is not a direct vulnerability, it acts similar to the FTP service in the way the credentials go through the network but encoded in Base64. This lack of encryption means an attacker can quickly reverse the encoding and retrieve the credentials. HTTP Basic Authorisation asks the user to authenticate before a web application and, without the use of any tunnel secure communications (SSL/TLS) it is easy to get credentials stolen.

All the evidence were sent to the appropriate CSIRT to be processed appropriately and mitigated.

As an experiment, it was compiled all the Apache CPE versions detected in this network and shown in figure 4.11. Then it was correlated with existing CVEs of that version to check for unpatched systems. The result is shown in figure 4.12. As it is possible to see, there are several vulnerabilities associated with the services, some of them passing the 40 unique CVE entries by version. This does not reflect risk, but it could infer that the system should be patched. Complete detailed information about the versions and CVEs associated can be found in the correspondent section of Appendix A.

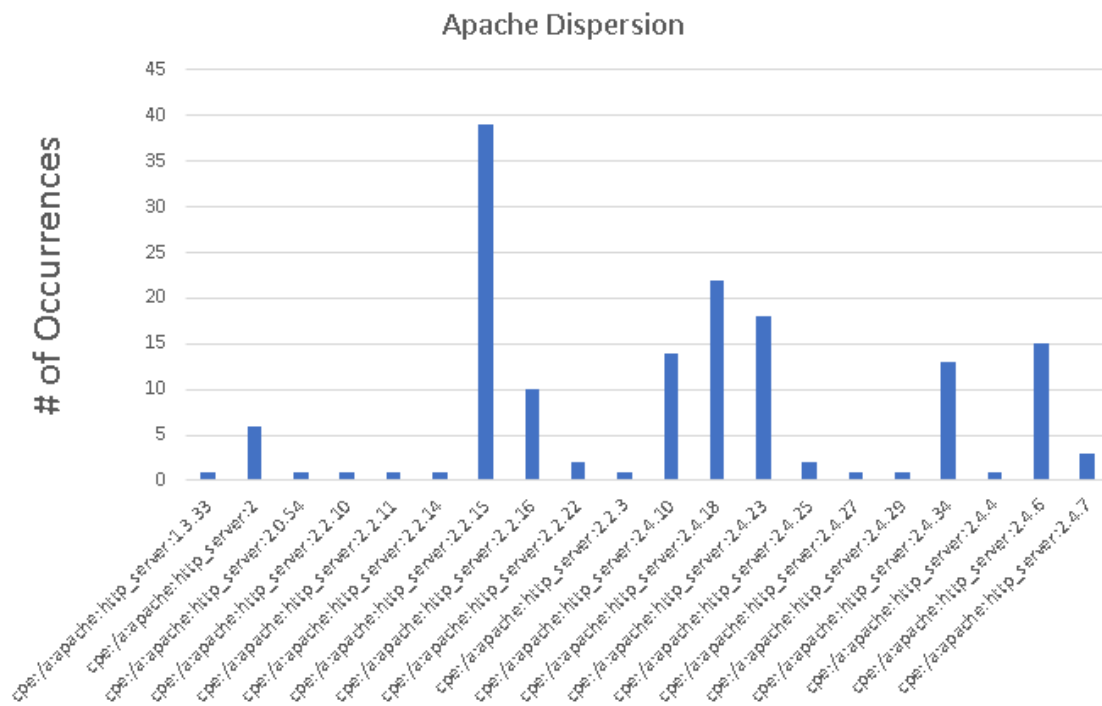


Figure 4.11 Apache version in Government Network

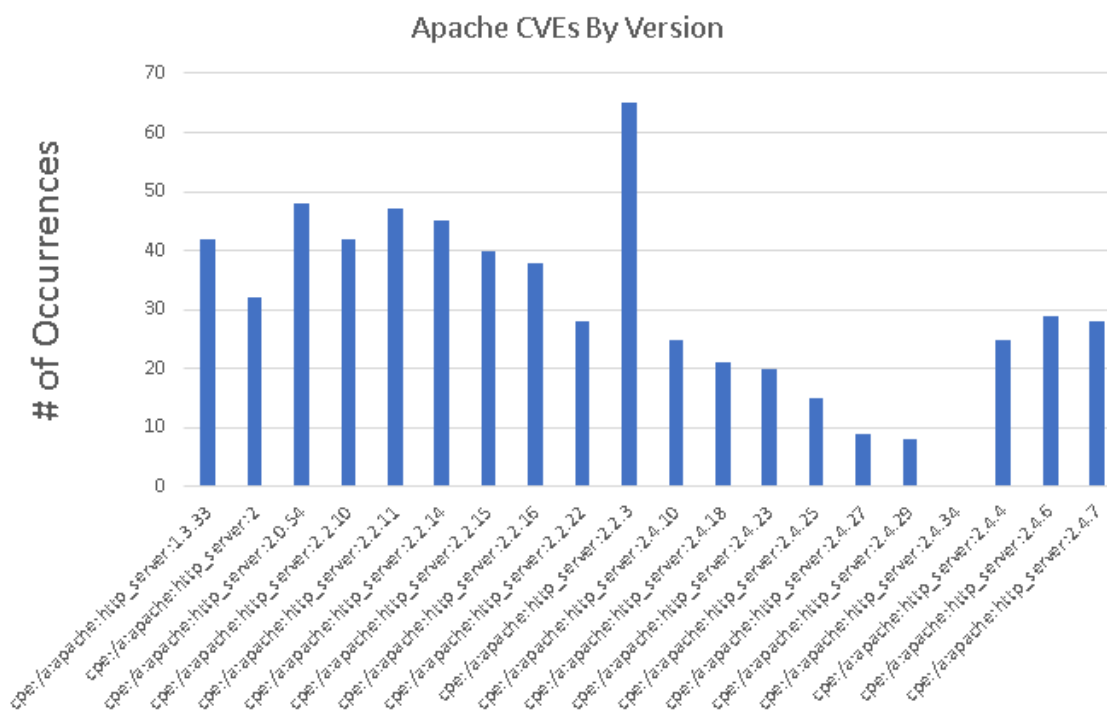


Figure 4.12 Apache CVEs in Government Network

Statistically speaking and without considering the outdated versions of running services and expired certificates (only considering the revoked certificate), 35 reports were made, and from

those reports, only 9 services were patched. This represents 25% patch rate that is well under the optimal rate.

Discovered hosts	1833
Discovered services	4792
Critical vulnerabilities disclosed	35
Patched vulnerabilities (28-01-2019)	9

Table 4.2 Result Statistics of a Government Network

4.3 Analysis of a Network of Health Institutions

The health institutions house some of the most critical services in their infrastructure such as several portals to each hospital and clinic under their management. This data gathering excludes some hospitals that have private management. This institution is now switching management and delegations which will require centralized services and forming a new CSIRT under the management of said services. This institution hosts critical data on each of the patients and employees. Databases with medical records are housed in their infrastructure to enable doctors to come up with a diagnose faster and providing patients with a cure faster. Historical data and performance analyzes of doctors and staff are also housed to enable human resources to handle each incident it occurs. In the case, this information is leaked or deleted several problems arise such as identity theft or insurance problems (if a health condition was concealed while doing a life insurance). In the case this data is deleted, in the case of ransomware, the ability to faster diagnose a patient is lost and in some extreme cases could mean life, or death.

Analyzing the services, it is possible to see the use of several outdated services. Some services such as Microsoft Internet Information Services (IIS) 5.0 are outdated and do not receive patches anymore by Microsoft. However, this service is still in use by a set of hosts in this institution. In table 4.3 it is possible to see the dispersion of Web services detected in this network. It is also possible to see the use of several outdated versions of said services with a high number of critical CVEs associated and should not be used.

This data shows that some hospitals are using outdated technologies that are prone and likely to be attacked. Some related news even describes software that was broken, an attacker exploited and demanded a ransom for the users' clinical data back [130].

CPE	Count	CVE Count	Max CVE Base Score
cpe:/a:apache:http_server	164		
cpe:/a:apache:http_server:2.2.12	2	46	10
cpe:/a:apache:http_server:2.2.15	6	40	7,8
cpe:/a:apache:http_server:2.2.22	1	28	7,5
cpe:/a:apache:http_server:2.2.3	6	65	10
cpe:/a:apache:http_server:2.2.32	1	21	7,5

Table 4.3 continued from previous page

CPE	Count	CVE Count	Max CVE Base Score
cpe:/a:apache:http_server:2.4.18	4	38	7,8
cpe:/a:apache:http_server:2.4.20	1	36	7,5
cpe:/a:apache:http_server:2.4.27	1	22	7,5
cpe:/a:apache:http_server:2.4.6	3	54	10
cpe:/a:apache:http_server:2.4.7	1	29	10
cpe:/a:microsoft:iis	194		
cpe:/a:microsoft:iis:10.0	59	0	0
cpe:/a:microsoft:iis:5.0	20	7	9
cpe:/a:microsoft:iis:6.0	16	10	10
cpe:/a:microsoft:iis:7.5	2	5	10
cpe:/a:microsoft:iis:8.0	4	0	0
cpe:/a:microsoft:iis:8.5	21	0	0

Table 4.3 Detected software on a network of Health Institutions

This finding is critical in the sense that patients personal data could be breached by the lack of security management in critical infrastructure that the patients are required to use to be able to receive treatment.

Furthermore, in these institutions, several undesired open ports are found like the SMB (port tcp/445) are open. Traditionally this protocol had critical vulnerabilities associated, and although almost impossible to determine the vulnerability using passive methods, its exposure is not recommended. Not only because of the vulnerability risk but because it could allow for administrative access if appropriated credentials are given (PS Exec[131]). With the information retrieved it is possible to observe that authentication is enabled, however, brute-force online attacks could target this service to try to gain access.

Another opened port that was found is the Telnet [132] service (port tcp/23). This service allows for remote administration under an unencrypted channel. The usage of this service indicates a severe risk since anyone listening to the network will be able to retrieve the credentials in clear text. In a theoretical, scenario Border Gateway Protocol (BGP) [133] hijacking attack could redirect all traffic to a particular Autonomous System (AS) then steal these credentials [134]. Off course being this highly theoretical the risk of using insecure communications is there.



Shodan	23	
Shodan	23	CentOS release 6.2 (Final)Kernel 2.6.32-220.el6.x86_64 on an x86_64

Figure 4.13 Exposition of Telnet on a network of Health Institutions

Another port found is the "date" port (port tcp/13) this port sends the current date of the device. It is one of the ports that should not be open to the Internet since several applications rely on the date to generate pseudo-random numbers for encryption and cookie generation. One attacker by

knowing the precise date of the system can reduce the brute-force range to more accurately break in the system. Other services such as Network Time Protocol (NTP) should be used instead of this port. Although NTP is not completely safe, it is recommended instead of this protocol. NTP services should be checked for exploits vulnerabilities [135] as well as DrDOS vulnerabilities [136].

One of the old protocols for file transfer (FTP) is also present. It is easy to set up and lightweight. However, the typical configuration is prone to several vulnerabilities such as the clear text authentication mechanism. This mechanism, as stated before, allows for an attacker who intercepts the connection to steal credentials and data. Other authentication mechanisms exist such as FTP over SSH and the use of certificates to authenticate the user, but they do not seem to be implemented. It was also found a FTP service with anonymous login access which could allow an attacker to read (and sometimes write) files in the service.

192.168.1.104	Shodan	21	220 Microsoft FTP Service530 User cann...	[cpe:/o:microsoft:windows]
192.168.1.11	Shodan	21	220----- Welcome to Pure-FTPd [privs...	[cpe:/a:pureftpd:pure-ftpd]
192.168.1.11	Shodan	21	220----- Welcome to Pure-FTPd [privs...	[cpe:/a:pureftpd:pure-ftpd]
192.168.1.11	Shodan	21	220----- Welcome to Pure-FTPd [privs...	[cpe:/a:pureftpd:pure-ftpd]
192.168.1.11	Shodan	21	220 Microsoft FTP Service530 User cann...	[cpe:/o:microsoft:windows]
192.168.1.11	Shodan	21	220 Benvindo ao servico de FTP da Logic ...	
192.168.1.11	Shodan	21	220-In case of any questions related to t...	
192.168.1.11	Shodan	21	220 Microsoft FTP Service530 User cann...	[cpe:/o:microsoft:windows]
192.168.1.11	Shodan	21	220 Microsoft FTP Service530 User cann...	[cpe:/o:microsoft:windows]
192.168.1.11	Shodan	21	220 ::ffff:195.201.173.42 FTP server read...	

Figure 4.14 Exposition of FTP on a network of Health Institutions

PortMapper was also detected among the results. This protocol (port tcp/111) maps ports to services. This information could help an attacker to understand what is the system that they are attacking. The usage of this protocol is considered an information leakage vulnerability.

Another open port found was NETBIOS [137] (port tcp/137). This historical protocol was primarily implemented on Microsoft Domain to exchange data and share resources. Several tools can be used to contact the service and retrieve information about the host.

Microsoft RDP is also found in the search results. These protocols allow for remote graphical administration of a machine.

There was also an instance of LDAP open. This protocol allows for the management of the domain authentication. It is typically contacted from hosts joint in a domain to verify credentials. Since the value of this port is critical, this is typically not exposed to the Internet.

There was also Microsoft PPTP VPN services found. Although they are VPNs servers, PPTP is considered obsolete and should be replaced by other, more resilient and secure, types of VPN.

[REDACTED]	Shodan	1723	BLOB
[REDACTED]	Shodan	1723	BLOB
[REDACTED]	Shodan	1723	BLOB
[REDACTED]	Shodan	1723	BLOB
[REDACTED]	Shodan	1723	BLOB

Figure 4.15 Exponion of PPTP on a network of Health Institutions

One interesting finding was the SAP Router TCP port open in one of the findings. This service acts as a reverse proxy to SAP systems. SAP Systems handle the management of the enterprise (i.e.: HR; financial and sometimes line production).

One host was using a revoked certificate. As already described, this is considered a bad practice and shows that the asset was left forgotten and exposed to the Internet.

A different exposed asset was found with indicators of compromise (IOC). This indicator was expressed by the use of external JavaScript functions that use the browser and subsequently the hardware of the computer to mine cryptocurrency (mining Monero [138] in the analyzed case). This scripts are considered intrusive and are now blocked in several browsers. A discussion of using said scripts is complex: on the one hand, the monetization of the website to support operating costs, and on another hand, the use of resources of the clients to make that monetization. Due to the purpose of this website it was considered an IOC since it was an institutional website.

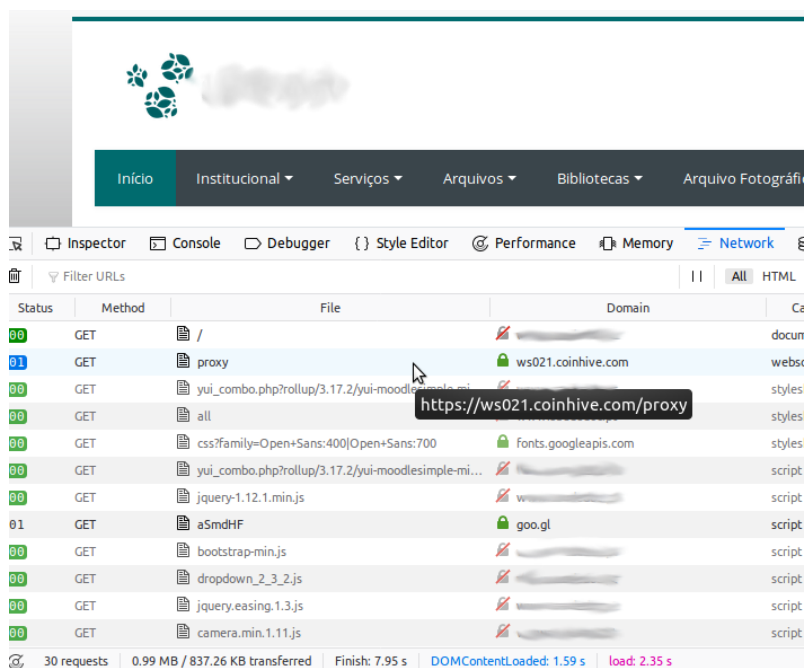


Figure 4.16 IOC on a network of Health Institutions

Considering the severity of the data involved, every discovery is considered critical. Although

every critical finding was reported, not all were fixed due to the short period of time given for the fix. There was a total of disclosures of 26 critical vulnerabilities that are marked to fix in the near future. Considering the size of the institution, it is possible to argue that this is a small subset of exposed services, however considering that the infrastructure is all connected and, by gaining access to a specific system others can be reached, it is possible that the danger is more significant since the perimeter is now breached. From the 647 services exposed discovered, having 26 with critical vulnerabilities (being 4% of the detected services) could indicate that the services were not known or were disregarded. Considering the last hypothesis, this dissertation aims to mitigate just that, discovering assets that normally are not in the scope of the monitoring capabilities of the SOCs. With the additional capabilities to deliver the discovered services and assets to other application, it could greatly benefit SOCs in discovering new vulnerabilities within their scope.

Discovered hosts	1264
Discovered services	647
Critical vulnerabilities disclosed	24
Patched vulnerabilities (28-01-2019)	NA

Table 4.4 Result Statistics of a network of Health Institutions

Chapter 5

Conclusions and Future Work

In this chapter a summary of the developed work will be presented as well as some conclusions. First and foremost, although this project was intended to be used as an automated perimeter assessment, it was pinpointed that some people could benefit from using in a penetration testing engagement [139]. Also, some bug bounty researchers could use it to enumerate potential flaws or applications in the scope of the bounty.

It was also observed that enumerating hosts and pass them to additional assessment tools like Arachni or even OpenVAS, or Nessus was easy, and therefore the following pipeline could be implemented to achieve a complete security assessment monitoring tool possible. This is the first step to a completely automated system that could even be achieved by integrating the rest of the applications in the IntelMQ pipeline, each result could be passed into a queue that will later scan the targets, and if the targets report a flaw or vulnerability, an alert could be emitted to the operator.

It is also possible to use this tool to enumerate the exposed perimeter (as stated before) and then use it to help determine the value of the contract of the new online insurances that start to appear during the implementation and enforcement of GDPR. Through the CPE enumeration, it is possible to have an overlook of the update policy of the organization and the security cleanness of it.

Not only it was possible to enumerate services without interacting with them, but the tool was also able to send its results to other tools effortlessly to continue the pipeline and proceed the engagement in two different ways: by consulting the created database, or pipelining the results to other tools such as Nmap. This behavior helps to contribute to the engagement and allows for a complete and automated pipeline. This is necessary for some systems that want to be a "penetration test as a service" where, with a cost, makes a complete vulnerability scanning. Other services can be commercialized to detect what is reachable from the Internet to the enterprise. Some similar systems could already benefit from this process to better protect their clients' infrastructure. Projects such as DiSIEM [140] could use this approach to detect new assets and create alerts to upper management to avoid excessive exposure and correlate even more information in present Security Information and Event Management (SIEM) systems. Connecting to other systems such as OpenVAS could also benefit for continuous security

compliance and monitoring system. Combining these technologies in a structured way could lead to a better protection of the network intended to protect.

There were several critical problems identified and reported to the corresponding entities in order to fix the issues. These entities have monitoring capabilities and have monitoring set in place to alert for any attack on their infrastructure. However, they miss several hosts in their scope, leaving the opportunity for attackers to exploit the systems and have access to internal networks. As seen in the previous chapter this was not only a single vulnerability or some system left out and forgotten, there were several issues showed that could be leveraged by attackers. This fact points a critical vulnerability on the whole enterprises which is security compliance. Even though the networks analyzed have CSIRT set in place, they miss on evaluating their security network. This points a deficiency on the security and the ability of the CSIRTs itself, when speaking of security, leaving space for exploitation will probably be exploited in the future with techniques that could not be known in the time being but could be discovered later. Another deficiency is the lack of knowledge in this field, the management of these CSIRTs are often lead by people who did not have contact with practical security. Despite that these persons focus on security compliance that, although correct, is inefficient when more technical components appear. In several occasions there was a need to have a meeting in order to explain base security concepts (such as XSS or SQL injections) for the mitigations or fixes be applied correctly. This dissertation and related work helped discover a vulnerability in the management of several health institutions that affect the national emergency systems. This vulnerability in question allows for an attacker to gain control of all the fleet and staff of said institutions as well as allow them to gather credentials for internal networks on those institutions. For this incident a CVE ID was assigned CVE-2019-6491 [141] and details would be released when all parties have mitigated the incident.

On the other side, some challenges arose with the developed work.

One of the main objectives was to reduce the time that a vulnerable host remains exposed. Although this reduced (since several vulnerabilities were not known) the resolution process still has several issues to be improved and fixed. Some teams take more time than other to apply patches, and some of them need several tries in order to mitigate the problem entirely.

Some of them do not even contact back to resolve the issue, meaning that some vulnerabilities are put on hold due to political reasons (change of management or determination of responsibilities). This management is an internal problem of some institutions that put on hold several tickets by the CSIRTs. This fact means that if a problem is detected, the time from detection to patch is greatly influenced. This work helps to detect some of the forgotten hosts, but the patch of vulnerabilities diverge significantly from institution to institution.

5.1 Future Work

Finishing this dissertation, some improvements could be made as well as new features that could help better detect these hosts.

The continuation of this concept will presuppose the rest of the pipeline. There is room to develop a complete pipeline that can gather information, scan, exploit and get persistence on a target. However, this ideology is focused on red teams it is indeed possible.

It is also possible to port this work to an online service like other solutions (such as Hacker Target) to perform scans on demand.

From the standpoint of the application itself it can undoubtedly be improved, either by the addition of new modules and a redesign of the framework to allow for remote procedure call (similar to the RPC interface of the Recon-ng framework).

Also, there should be a pool of DNS resolutions so one API will not be stressed out.

Other services that scan for services could be used, however, they are hard to find, and sometimes they are costly.

Another aspect that would be interesting to add is in a complete solution that correlates with Network Monitoring, for example, if the network monitoring detects an attack signature and if this detection detects that the server version is possible exploitable, then a critic alarm should be raised. This information can also be correlated with additional services, for example, public exposure of exploited websites like zone-H.org. This information could significantly help the detection of incidents.

By combining all these technologies (Network and Host Intrusion Detection Systems -NIDS/HIDS- , Open Source feed analysis, and Automatic Scanning) this could be helpful in giving a piece of mind for network operators.

5.2 Legal

During this dissertation, information was exchanged between the authors to the corresponding CSIRTs. Accordingly with Portuguese law, since no interaction was made with the targets, there are no indications of compromise. However, if a problem was suspected an advisory to the corresponding CSIRT to handle the possible issue was made. No automated tests were done and no attempt to compromise the systems were made. Some services that remain vulnerable would not be public in this dissertation to avoid compromise upon the publishing of this dissertation. Some services may be left unchanged since a possible vulnerability was not confirmed.

Every disclose was done accordingly with the ISO-29147 standard[49].

Only if a vulnerability has been idle for 60 days, it can be considered abandoned and could be publicized.

Appendix A

Raw data results

A.1 University Network

CPE	Count	CVE Count	Max CVE Base Score
cpe:/a:exim:exim:4.91	17	0	0
cpe:/a:igor_sysoev:nginx	255		
cpe:/a:igor_sysoev:nginx:1.10.3	127	0	0
cpe:/a:igor_sysoev:nginx:1.1.19	1	1	7.5
cpe:/a:igor_sysoev:nginx:1.12.2	5	0	0
cpe:/a:igor_sysoev:nginx:1.14.0	19	0	0
cpe:/a:igor_sysoev:nginx:1.2.1	1	2	7.5
cpe:/a:igor_sysoev:nginx:1.4.6	2	0	0
cpe:/a:igor_sysoev:nginx:1.6.2	1	0	0
cpe:/a:igor_sysoev:nginx:1.9.3	2	0	0
cpe:/a:indy:httpd:17.2.31.2018	2	0	0
cpe:/a:indy:httpd:17.3.33.2753	2	0	0
cpe:/a:microsoft:exchange_server	8		
cpe:/a:microsoft:exchange_server:2010	1	16	10
cpe:/a:microsoft:iis	30		
cpe:/a:microsoft:iis:10.0	12	0	0
cpe:/a:microsoft:iis:5.1	1	6	9
cpe:/a:microsoft:iis:6.0	4	10	10
cpe:/a:microsoft:iis:7.0	2	6	9
cpe:/a:microsoft:iis:7.5	83	5	10
cpe:/a:microsoft:iis:8.0	1	0	0
cpe:/a:microsoft:iis:8.5	38	0	0
cpe:/a:mortbay:jetty:9.2.13.v20150730	1	0	0
cpe:/a:mysql:mysql:5.1.73	1	0	0

Table A.1 continued from previous page

CPE	Count	CVE Count	Max CVE Base Score
cpe:/a:mysql:mysql:5.6.41	1	0	0
cpe:/a:openbsd:openssh:4.3	4	34	9,3
cpe:/a:openbsd:openssh:4.3p2	1	32	8,5
cpe:/a:openbsd:openssh:5.2	2	26	8,5
cpe:/a:openbsd:openssh:5.3	16	26	8,5
cpe:/a:openbsd:openssh:5.3p1	1	22	8,5
cpe:/a:openbsd:openssh:5.4	1	27	8,5
cpe:/a:openbsd:openssh:5.5	1	27	8,5
cpe:/a:openbsd:openssh:5.8p1	1	21	8,5
cpe:/a:openbsd:openssh:5.9	2	23	8,5
cpe:/a:openbsd:openssh:5.9p1	3	20	8,5
cpe:/a:openbsd:openssh:6.2	1	23	8,5
cpe:/a:openbsd:openssh:6.4	6	22	8,5
cpe:/a:openbsd:openssh:6.6.1	24	16	8,5
cpe:/a:openbsd:openssh:6.6.1p1	4	16	8,5
cpe:/a:openbsd:openssh:6.7p1	3	15	8,5
cpe:/a:openbsd:openssh:7.2	3	13	7,8
cpe:/a:openbsd:openssh:7.4	16	3	5
cpe:/a:openbsd:openssh:7.4p1	3	2	5
cpe:/a:php:php:4.3.9	1	392	10
cpe:/a:php:php:5.1.6	17	363	10
cpe:/a:php:php:5.2.17	2	256	10
cpe:/a:php:php:5.2.5	1	315	10
cpe:/a:php:php:5.3.10-1ubuntu3.14	1	228	10
cpe:/a:php:php:5.3.20	1	216	10
cpe:/a:php:php:5.3.2-1ubuntu4.30	2	286	10
cpe:/a:php:php:5.3.28	1	208	10
cpe:/a:php:php:5.3.3	31	267	10
cpe:/a:php:php:5.3.3-7	1	243	10
cpe:/a:php:php:5.3.6	2	239	10
cpe:/a:php:php:5.3.6-13ubuntu3.9	2	226	10
cpe:/a:php:php:5.4.16	18	209	10
cpe:/a:php:php:5.4.45-0	1	138	10
cpe:/a:php:php:5.5.26	3	146	10
cpe:/a:php:php:5.5.38	2	82	10
cpe:/a:php:php:5.5.6	1	212	10
cpe:/a:php:php:5.5.9-1ubuntu4.25	2	204	10

Table A.1 continued from previous page

CPE	Count	CVE Count	Max CVE Base Score
cpe:/a:php:php:5.5.9-1ubuntu4.26	1	204	10
cpe:/a:php:php:5.6.11	2	134	10
cpe:/a:php:php:5.6.36	9	23	10
cpe:/a:php:php:5.6.37-0	2	23	10
cpe:/a:php:php:5.6.38	7	22	10
cpe:/a:php:php:5.6.38-0	4	22	10
cpe:/a:php:php:7.0.32	12	14	7,5
cpe:/a:php:php:7.1.6	1	23	7,5
cpe:/a:postfix:postfix	29		
cpe:/a:postfix:postfix:2.2.5	2	3	6,8
cpe:/a:postfix:postfix:2.2.8	1	3	6,8
cpe:/a:proftpd:proftpd:1.3.1	1	10	9
cpe:/a:proftpd:proftpd:1.3.5b	1	0	0
cpe:/a:proftpd:proftpd:1.3.5e	1	0	0
cpe:/a:pureftpd:pure-ftpd	5		
cpe:/a:sendmail:sendmail:8.13.8/8.13.8	1	2	7,5
cpe:/a:wordpress:wordpress	7		
cpe:/a:wordpress:wordpress:4.5.3	1	51	7,5
cpe:/a:wordpress:wordpress:4.7.3	1	31	7,5
cpe:/a:wordpress:wordpress:4.9.3	1	5	6,5
cpe:/a:wordpress:wordpress:4.9.8	247	1	6,5
cpe:/o:microsoft:windows	12		
cpe:/o:microsoft:windows:server2008r2	1	1174	10

Table A.1 Detected software in a University Network

A.2 Government Network

CPE	Count	CVE Count	Max CVE Base Score
cpe:/a:apache:http_server	173		
cpe:/a:apache:http_server:1.3.33	1	42	10
cpe:/a:apache:http_server:2	6	32	10
cpe:/a:apache:http_server:2.0.54	1	48	10
cpe:/a:apache:http_server:2.2.10	1	42	10
cpe:/a:apache:http_server:2.2.11	1	47	10
cpe:/a:apache:http_server:2.2.14	1	45	10
cpe:/a:apache:http_server:2.2.15	39	40	7,8

Table A.2 continued from previous page

CPE	Count	CVE Count	Max CVE Base Score
cpe:/a:apache:http_server:2.2.16	10	38	7,8
cpe:/a:apache:http_server:2.2.22	2	28	7,5
cpe:/a:apache:http_server:2.2.3	1	65	10
cpe:/a:apache:http_server:2.4.10	14	25	7,5
cpe:/a:apache:http_server:2.4.18	22	21	7,5
cpe:/a:apache:http_server:2.4.23	18	20	7,5
cpe:/a:apache:http_server:2.4.25	2	15	7,5
cpe:/a:apache:http_server:2.4.27	1	9	6,8
cpe:/a:apache:http_server:2.4.29	1	8	6,8
cpe:/a:apache:http_server:2.4.34	13	0	0
cpe:/a:apache:http_server:2.4.4	1	25	7,5
cpe:/a:apache:http_server:2.4.6	15	29	6,8
cpe:/a:apache:http_server:2.4.7	3	28	7,5
cpe:/a:exim:exim:4.84_2	3	0	0
cpe:/a:exim:exim:4.91	29	0	0
cpe:/a:ibm:http_server	2		
cpe:/a:igor_sysoev:nginx	474		
cpe:/a:igor_sysoev:nginx:1.0.15	3	0	0
cpe:/a:igor_sysoev:nginx:1.10.1	2	0	0
cpe:/a:igor_sysoev:nginx:1.13.12	8	0	0
cpe:/a:igor_sysoev:nginx:1.14.0	4	0	0
cpe:/a:igor_sysoev:nginx:1.4.6	8	0	0
cpe:/a:igor_sysoev:nginx:1.7.8	2	0	0
cpe:/a:microsoft:exchange_server	9		
cpe:/a:microsoft:exchange_server:2010	1	16	10
cpe:/a:microsoft:iis	274		
cpe:/a:microsoft:iis:10.0	31	0	0
cpe:/a:microsoft:iis:6.0	57	10	10
cpe:/a:microsoft:iis:7.0	9	6	9
cpe:/a:microsoft:iis:7.5	152	5	10
cpe:/a:microsoft:iis:8.0	16	0	0
cpe:/a:microsoft:iis:8.5	129	0	0
cpe:/a:mysql:mysql	1		
cpe:/a:mysql:mysql:5.1.73-1	1	0	0
cpe:/a:mysql:mysql:5.1.73-community	1	0	0
cpe:/a:mysql:mysql:5.5.47-0	1	0	0
cpe:/a:mysql:mysql:5.5.5-10.0.36...	1	116	10

Table A.2 continued from previous page

CPE	Count	CVE Count	Max CVE Base Score
cpe:/a:mysql:mysql:5.5.61-cll	2	0	0
cpe:/a:mysql:mysql:5.6.41	2	0	0
cpe:/a:mysql:mysql:5.7.23	1	0	0
cpe:/a:mysql:mysql:5.7.24	2	0	0
cpe:/a:openbsd:openssh:4.3	1	34	9,3
cpe:/a:openbsd:openssh:5.1p1	1	22	8,5
cpe:/a:openbsd:openssh:5.2	1	26	8,5
cpe:/a:openbsd:openssh:5.3	4	26	8,5
cpe:/a:openbsd:openssh:5.5p1	1	23	8,5
cpe:/a:openbsd:openssh:5.9p1	1	19	8,5
cpe:/a:openbsd:openssh:6.0p1	1	19	8,5
cpe:/a:openbsd:openssh:6.7p1	6	15	8,5
cpe:/a:openbsd:openssh:7.4	6	3	5
cpe:/a:openbsd:openssh:7.4p1	2	3	5
cpe:/a:php:php:5.3.2-1ubuntu4.10	3	286	10
cpe:/a:php:php:5.3.3	1	267	10
cpe:/a:php:php:5.3.6-6	1	239	10
cpe:/a:php:php:5.3.9	3	229	10
cpe:/a:php:php:5.4.14	1	212	10
cpe:/a:php:php:5.4.16	1	209	10
cpe:/a:php:php:5.4.41	2	156	10
cpe:/a:php:php:5.4.45	2	138	10
cpe:/a:php:php:5.5.38	1	178	10
cpe:/a:php:php:5.5.9-1ubuntu4.21	4	204	10
cpe:/a:php:php:5.6.3	1	181	10
cpe:/a:php:php:5.6.30	1	39	10
cpe:/a:php:php:7.0.29	7	19	7,5
cpe:/a:postfix:postfix	19		
cpe:/a:sendmail:sendmail:8.12.8/8.12.8	1	10	10
cpe:/a:sendmail:sendmail:8.14.3/8.14.3	1	2	7,5
cpe:/a:proftpd:proftpd:1.3.5d	1	1	2,1
cpe:/a:pureftpd:pure-ftpd	12		
cpe:/a:wordpress:wordpress	8		
cpe:/a:wordpress:wordpress:4.7.11	5	0	0
cpe:/a:wordpress:wordpress:4.7.4	1	31	7,5
cpe:/a:wordpress:wordpress:4.9.8	16	1	6,5
cpe:/o:canonical:ubuntu_linux	1		

Table A.2 continued from previous page

CPE	Count	CVE Count	Max CVE Base Score
cpe:/o:debian:debian_linux	1		
cpe:/o:microsoft:windows	100		
cpe:/o:microsoft:windows:server2003	2	444	10

Table A.2 Detected software in a Government Network

A.3 Health Institutions

CPE	Count	CVE Count	Max CVE Base Score
cpe:/a:apache:http_server	164		
cpe:/a:apache:http_server:2.2.12	2	46	10
cpe:/a:apache:http_server:2.2.15	6	40	7,8
cpe:/a:apache:http_server:2.2.22	1	28	7,5
cpe:/a:apache:http_server:2.2.3	6	65	10
cpe:/a:apache:http_server:2.2.32	1	21	7,5
cpe:/a:apache:http_server:2.4.18	4	38	7,8
cpe:/a:apache:http_server:2.4.20	1	36	7,5
cpe:/a:apache:http_server:2.4.27	1	22	7,5
cpe:/a:apache:http_server:2.4.6	3	54	10
cpe:/a:apache:http_server:2.4.7	1	29	10
cpe:/a:bitwise:winsshd:7.39	1	0	0
cpe:/a:exim:exim:4.91	14	0	0
cpe:/a:igor_sysoev:nginx	38		
cpe:/a:igor_sysoev:nginx:1.10.3	2	0	0
cpe:/a:igor_sysoev:nginx:1.12.2	6	0	0
cpe:/a:igor_sysoev:nginx:1.14.0	1	0	0
cpe:/a:igor_sysoev:nginx:1.14.1	2	0	0
cpe:/a:microsoft:exchange_server:2010	1	16	10
cpe:/a:microsoft:iis	194		
cpe:/a:microsoft:iis:10.0	59	0	0
cpe:/a:microsoft:iis:5.0	20	7	9
cpe:/a:microsoft:iis:6.0	16	10	10
cpe:/a:microsoft:iis:7.5	2	5	10
cpe:/a:microsoft:iis:8.0	4	0	0
cpe:/a:microsoft:iis:8.5	21	0	0
cpe:/a:mysql:mysql:5.5.5-10.1.31...	1	115	10
cpe:/a:mysql:mysql:5.6.41	1	0	0

Table A.3 continued from previous page

CPE	Count	CVE Count	Max CVE Base Score
cpe:/a:mysql:mysql:5.6.41-84.1	1	0	0
cpe:/a:mysql:mysql:5.7.23-cll-lve	1	0	0
cpe:/a:openbsd:openssh:4.3	1	34	9,3
cpe:/a:openbsd:openssh:5.3	3	26	8,5
cpe:/a:openbsd:openssh:5.9p1	1	19	8,5
cpe:/a:openbsd:openssh:6.6.1	1	16	8.5
cpe:/a:openbsd:openssh:7.4	2	3	5
cpe:/a:php:php:5.3.10-1ubuntu3.10	1	228	10
cpe:/a:php:php:5.3.29	1	201	10
cpe:/a:php:php:5.5.38	1	82	10
cpe:/a:php:php:5.5.9-1ubuntu4.24	1	204	10
cpe:/a:php:php:5.6.37	4	23	10
cpe:/a:pureftpd:pure-ftpd	5		
cpe:/a:wordpress:wordpress	1		
cpe:/o:cisco:ios	1		
cpe:/o:microsoft:windows	27		

Table A.3 Detected software in a network of Health Institutions

Bibliography

- [1] W. Tounsi and H. Rais, “A survey on technical threat intelligence in the age of sophisticated cyber attacks,” *Computers and Security*, vol. 72, pp. 212–233, 2018.
- [2] “Art. 32 GDPR – Security.” Electronical version on: <https://gdpr-info.eu/art-32-gdpr/>. [Last accessed 22-01-2019].
- [3] Lusa, “Hospital do Barreiro contesta judicialmente coima de 400 mil euros de Comissão de Dados,” Oct 2018. Electronical version on: <https://goo.gl/vQhVcB>. [Last accessed 22-01-2019].
- [4] S. TekİR, “An Implementation Model for Open Sources Evaluation,” *Intelligence*, 2004. Electronical version on: <http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.427.6355&rep=rep1&type=pdf>. [Last accessed 14-01-2019].
- [5] A. Romano, “How Facebook made it impossible to delete Facebook,” Dec 2018. Electronical version on: <https://goo.gl/yV6Gr2>. [Last accessed 22-01-2019].
- [6] D. Zekowitz, M.V.; Wallace, “Experimental models for validating technology,” *Computer*, vol. 31, 1998. Electronical version on: <https://ieeexplore.ieee.org/abstract/document/675630>.
- [7] B. Genge and C. Enăchescu, “ShoVAT: Shodan-based Vulnerability Assessment Tool for Internet-facing Services,” *Sec. and Commun. Netw.*, vol. 9, pp. 2696–2714, Oct. 2016.
- [8] “Shodan - The search engine for the Internet of Things.” <https://www.shodan.io/>. [Last accessed 27-01-2019].
- [9] “CPE Specifications.” Electronical version on: <https://cpe.mitre.org/specification/>. [Last accessed 22-01-2019].
- [10] “HyperText Transfer Protocol (HTTP/1.1): Message Syntax and Routing.” <https://tools.ietf.org/html/rfc7230>.
- [11] “Censys.” <https://censys.io/>. [Last accessed 27-01-2019].
- [12] “RIPE Database Query.” <https://apps.db.ripe.net/db-web-ui/#/fulltextsearch>. [Last accessed 27-01-2019].

- [13] “HackerTarget.com | Find DNS Host Records | Subdomain Finder.” <https://hackertarget.com/find-dns-host-records/>. [Last accessed 27-01-2019].
- [14] “Bing Web Search API | Microsoft Azure.” <https://azure.microsoft.com/en-us/services/cognitive-services/bing-web-search-api/>. [Last accessed 27-01-2019].
- [15] “Custom Search JSON API | Custom Search | Google Developers.” <https://developers.google.com/custom-search/v1/overview>. [Last accessed 27-01-2019].
- [16] “Threat Crowd | Threatcrowd.org Open Source Threat Intelligence.” <https://www.threatcrowd.org/>. [Last accessed 27-01-2019].
- [17] “Netcraft | Internet Research, Anti-Phishing and PCI Security Services.” <https://www.netcraft.com/>. [Last accessed 27-01-2019].
- [18] “Certificate Search.” Electronical version on: <https://crt.sh/>. [Last accessed 14-01-2019].
- [19] “SSL Scanner.” <http://www.ssltools.com/>. [Last accessed 27-01-2019].
- [20] “The Secure Shell (SSH) Connection Protocol.” <https://tools.ietf.org/html/rfc4254>.
- [21] N. Schagen, K. Koning, H. Bos, and C. Giuffrida, “Towards Automated Vulnerability Scanning of Network Servers,” *Proceedings of the 11th European Workshop on Systems Security*, pp. 5:1–5:6, 2018.
- [22] “The ZMap Project.” Electronical version on: <https://zmap.io/research>. [Last accessed 22-01-2019].
- [23] “ZoomEye - Cyberspace Search Engine.” <https://www.zoomeye.org/>. [Last accessed 27-01-2019].
- [24] “FOFA Pro.” <https://fofa.so/>. [Last accessed 27-01-2019].
- [25] “OpenVAS Compendium – NVT.” <http://www.openvas.org/openvas-nvt-feed.html>. [Last accessed 27-01-2019].
- [26] MITRE, “Open Vulnerability and Assessment Language.” Electronical version on: <https://oval.mitre.org/language/about/overview.html>. [Last accessed 13-01-2019].
- [27] “US-CERT - Bulletins.” <https://www.us-cert.gov/ncas/bulletins>. [Last accessed 27-01-2019].

- [28] “CVE - Common Vulnerabilities and Exposures (CVE).” <https://cve.mitre.org/>. [Last accessed 27-01-2019].
- [29] “NIST - CCE.” <https://nvd.nist.gov/config/cce>. [Last accessed 27-01-2019].
- [30] “NIST CPE.” <https://nvd.nist.gov/products/cpe>. [Last accessed 27-01-2019].
- [31] “NIST XCCDF.” <https://csrc.nist.gov/projects/security-content-automation-protocol/specifications/xccdf/>. [Last accessed 27-01-2019].
- [32] “NVD - Data Feeds.” Electronical version on: <https://nvd.nist.gov/vuln/data-feeds>. [Last accessed 14-01-2019].
- [33] “CWE Database.” <https://cwe.mitre.org/>. [Last accessed 27-01-2019].
- [34] Toolswatch, “Github vFeed,” Nov 2017. <https://github.com/toolswatch/vFeed>. [Last accessed 27-01-2019].
- [35] S. D. Quinn, P. Mell, and K. Kent, “The Security Content Automation Program (SCAP): Automating Compliance Checking, Vulnerability Management, and Security Measurement,” *NIST Interagency Report*, vol. 7343, 2006.
- [36] “FY 2018 CIO FISMA Metrics,” Sep 2018. Electronical version on: https://www.dhs.gov/sites/default/files/publications/FY2018CIOFISMAMetrics_V2.0.1_final_0.pdf. [Last accessed 14-01-2019].
- [37] NIST, “Security Content Automation Protocol,” 2007. Electronical version on: <https://nvd.nist.gov/scap/docs/SCAP.doc>. [Last accessed 13-01-2019].
- [38] “CVSS v3.0 Specification Document.” Electronical version on: <https://www.first.org/cvss/specification-document>. [Last accessed 14-01-2019].
- [39] “BeautifulSoup Documentation.” Electronical version on: <https://www.crummy.com/software/BeautifulSoup/bs4/doc/>. [Last accessed 22-01-2019].
- [40] “lxml - XML and HTML with Python.” Electronical version on: <https://lxml.de/>. [Last accessed 22-01-2019].
- [41] T. Tomes, “Recon-ng | BitBucket.” Electronical version on: <https://bitbucket.org/LaNMaSteR53/recon-ng>. [Last accessed 31-01-2019].
- [42] “Regular Expressions.” Electronical version on: http://pubs.opengroup.org/onlinepubs/9699919799/basedefs/V1_chap09.html. [Last accessed 22-01-2019].

- [43] “Guide to Supervisory Control and Data Acquisition (SCADA) and Industrial Control Systems Security.” Electrical version on: <https://goo.gl/rzdmqG>. [Last accessed 22-01-2019].
- [44] D. Murdoch, *Blue Team handbook: incident response edition: a condensed field guide for the cyber security incident responder*. CreateSpace Independent Publishing Platform, 2014.
- [45] T. Grance, K. Kent, and B. Kim, ““Computer Security Incident Handling Guide” Recommendations of the National Institute of Standards and Technology NIST800-61,” 2004.
- [46] Google Inc., “Site Reliability Engineering.” Electrical version on: <https://landing.google.com/sre/book/chapters/postmortem-culture.html>. [Last accessed 13-01-2019].
- [47] ENISA, “Good Practice Guide for Incident Management,” *European Network and Information Security Agency*, 2010. Electrical version on: <http://www.enisa.europa.eu/activities/cert/support/incident-management/files/good-practice-guide-for-incident-management>. [Last accessed 13-01-2019].
- [48] “ISO/IEC 38500:2015,” Feb 2015. Electrical version on: <https://www.iso.org/standard/62816.html>.
- [49] “ISO/IEC 29147:2018,” Oct 2018. Electrical version on: <https://www.iso.org/standard/72311.html>.
- [50] AlienVault, “Insider’s Guide to Incident Response,” Electrical version: https://learn.alienvault.com/c/alien-vault-incident?utm_internal=soc-irlookbook&x=5v9G6V&xs=1248. [Last accessed 13-01-2019].
- [51] “Snort - Network Intrusion Detection & Prevention System.” Electrical version on: <https://www.snort.org/>. [Last accessed 31-01-2019].
- [52] “Suricata.” Electrical version on: <https://suricata-ids.org/>. [Last accessed 31-01-2019].
- [53] H. Doreau, “Vulnerability management with OpenVAS,” no. January, pp. 1–4, 2011. Electrical version on: http://moutane.net/RMLL2011/jour_2/5-Henri-OpenVAS-RMLL2011.pdf. [Last accessed 25-01-2019].
- [54] “AlienVault - Open Threat Exchange.” <https://otx.alienvault.com/>. [Last accessed 27-01-2019].
- [55] “certtools/intelmq,” Jul 2018. Electrical version on: <https://github.com/certtools/intelmq>. [Last accessed 13-01-2019].

- [56] ENISA, “Incident Handling Automation,” Feb 2016. Electronical version on: <https://www.enisa.europa.eu/topics/csirt-cert-services/community-projects/incident-handling-automation>. [Last accessed 13-01-2019].
- [57] “IntelMQ Developers Guide.” Electronical version on: <https://intelmq.readthedocs.io/en/latest/Developers-Guide/>. [Last accessed 14-01-2019].
- [58] Electronical version on: <https://redis.io/>. [Last accessed 31-01-2019].
- [59] M. El, E. McMahon, S. Samtani, M. Patton, and H. Chen, “Benchmarking vulnerability scanners: An experiment on SCADA devices and scientific instruments,” *2017 IEEE International Conference on Intelligence and Security Informatics: Security and Big Data, ISI 2017*, pp. 83–88, 2017.
- [60] “Performance Measurement Guide for Information Security.” Electronical version on: <https://csrc.nist.gov/publications/detail/sp/800-55/rev-1/final>. [Last accessed 22-01-2019].
- [61] “Security Risk Analysis of Enterprise Networks Using Probabilistic Attack Graphs.” Electronical version on: <https://www.govinfo.gov/content/pkg/GOVPUB-C13-7342227fee4e945f5dd1bb8f165ecbae/pdf/GOVPUB-C13-7342227fee4e945f5dd1bb8f165ecbae.pdf>. [Last accessed 17-01-2019].
- [62] T. Base and T. Base, “Common Vulnerability Scoring System v3.0 Examples,” no. July, pp. 1–38, 2016. Electronical version on: <https://www.first.org/cvss/examples>. [Last accessed 25-01-2019].
- [63] “Metasploit - penetration testing software, pen testing security.” Electronical version on: <https://www.metasploit.com/>. [Last accessed 31-01-2019].
- [64] BetaFred, “Microsoft Security Bulletin MS17-010 - Critical.” Electronical version on: <https://docs.microsoft.com/en-us/security-updates/securitybulletins/2017/ms17-010>. [Last accessed 22-01-2019].
- [65] LingyuWang, S. Jajodia, and A. Singhal, *Network Security Metrics*. 2017.
- [66] “Tor Project.” Electronical version on: <https://www.torproject.org/docs/documentation.html.en>. [Last accessed 22-01-2019].
- [67] K. A. Farris, A. Shah, G. Cybenko, R. Ganesan, and S. Jajodia, “VULCON: A System for Vulnerability Prioritization, Mitigation, and Management,” *ACM Trans. Priv. Secur.*, vol. 21, pp. 16:1–16:28, June 2018.

- [68] H. Séneca, “Exame Informatica - Site que dá acesso a dados clínicos de 1,8 milhões utentes do SNS esteve vulnerável dois anos .” Electronical version on: <https://goo.gl/m5wpiU>. [Last accessed 08-12-2018].
- [69] J. Crowe, “WannaCry Ransomware Statistics: The Numbers Behind the Outbreak,” May 2017. Electronical version on: <https://blog.barkly.com/wannacry-ransomware-statistics-2017>. [Last accessed 08-12-2018].
- [70] Computerworld, “PT/MEO alvo de ataque de ransomware (actualizado),” May 2017. Electronical version on: <https://www.computerworld.com.pt/2017/05/12/pt-alvo-de-ataque-de-ransomware>. [Last accessed 08-12-2018].
- [71] H. M. Malte and M. Hoppler, “Vulnerability Assessment Tool.” Electronical version on: <https://osr.cs.fau.de/wp-content/uploads/2013/04/Bosch-AEY1-Amos-proposal.pdf>. [Last accessed 25-01-2019].
- [72] “Vulnerability Details : CVE-2015-8562.” Electronical version on: <https://www.cvedetails.com/cve/cve-2015-8562>. [Last accessed 08-12-2018].
- [73] Synopsys, “The Heartbleed Bug.” Electronical version on: <http://heartbleed.com/>. [Last accessed 22-01-2019].
- [74] “NVD - CVE-2017-5638.” Electronical version on: <https://nvd.nist.gov/vuln/detail/CVE-2015-0204>. [Last accessed 22-01-2019].
- [75] “What is the Mirai Botnet?.” Electronical version on: <https://www.cloudflare.com/learning/ddos/glossary/mirai-botnet/>. [Last accessed 22-01-2019].
- [76] “Open Resolver Project.” Electronical version on: <http://openresolverproject.org/>. [Last accessed 22-01-2019].
- [77] “Offensive Security’s Exploit Database Archive.” Electronical version on: <https://www.exploit-db.com/google-hacking-database>. [Last accessed 22-01-2019].
- [78] “Baidu.” <http://www.baidu.com/>. [Last accessed 27-01-2019].
- [79] “DNSdumpster.com - DNS recon and research.” <https://dnsdumpster.com/>. [Last accessed 27-01-2019].
- [80] “VirusTotal.” <https://www.virustotal.com/>. [Last accessed 27-01-2019].
- [81] “Yahoo.” <https://www.yahoo.com/>. [Last accessed 27-01-2019].
- [82] “Yandex.” <https://yandex.com/>. [Last accessed 27-01-2019].
- [83] S. Helme, “Security Headers.” <https://securityheaders.com/>. [Last accessed 27-01-2019].

- [84] “Exalead.” <https://www.exalead.com/search/>. [Last accessed 27-01-2019].
- [85] “RiskIQ PassiveTotal Threat Investigation Platform.” <https://www.riskiq.com/products/passivetotal/>. [Last accessed 27-01-2019].
- [86] “Qwant – The search engine that respects your privacy.” <https://www.qwant.com/>. [Last accessed 27-01-2019].
- [87] “searx.” <https://searx.me/>. [Last accessed 27-01-2019].
- [88] “BuiltWith.” <https://builtwith.com/>. [Last accessed 27-01-2019].
- [89] “Certificate Transparency.” Electronical version on: <https://www.certificate-transparency.org/>. [Last accessed 14-01-2019].
- [90] M. Yip, “ThreatMiner.org | Data Mining for Threat Intelligence.” <https://www.threatminer.org/>. [Last accessed 27-01-2019].
- [91] “Find subdomains online - FindSubDomains.” <https://findsubdomains.com/>. [Last accessed 27-01-2019].
- [92] BinaryEdge, “BinaryEdge.” <https://www.binaryedge.io/>. [Last accessed 27-01-2019].
- [93] “SecurityTrails.” <https://securitytrails.com/>. [Last accessed 27-01-2019].
- [94] “DMARC.” Electronical version on: <https://dmarc.org/>. [Last accessed 22-01-2019].
- [95] “Amap,” Feb 2014. Electronical version on: <https://tools.kali.org/information-gathering/amap>. [Last accessed 22-01-2019].
- [96] “Arachni - Web Application Security Scanner Framework.” Electronical version on: <http://www.arachni-scanner.com/>. [Last accessed 22-01-2019].
- [97] “Acunetix.” Electronical version on: <https://www.acunetix.com/>. [Last accessed 22-01-2019].
- [98] “OWASP Top Ten Project.” Electronical version on: https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project. [Last accessed 27-01-2019].
- [99] N. Jaswal and S. Rahalkar, “Metasploit Revealed: Secrets of the Expert Pentester.” Electronical version on: <https://goo.gl/qjoiqU>. [Last accessed 22-01-2019].
- [100] “Incident Handling Automation,” Feb 2016. Electronical version on: <https://www.enisa.europa.eu/topics/csirt-cert-services/community-projects/incident-handling-automation>. [Last accessed 14-01-2019].

- [101] P. Rodrigues, “Added ThreatMiner subdomain search.” Electronical version on: <https://bitbucket.org/LaNMaSteR53/recon-ng/pull-requests/284>. [Last accessed 14-01-2019].
- [102] “crontab(5) - Linux man page.” Electronical version on: <https://linux.die.net/man/5/crontab>. [Last accessed 14-01-2019].
- [103] ScrapeHero, “How to Solve Simple Captchas using Python Tesseract,” Jan 2018. Electronical version on: <https://www.scrapehero.com/how-to-solve-simple-captchas-using-python-tesseract/>. [Last accessed 14-01-2019].
- [104] “Description of the INETNUM Object.” Electronical version on: <https://www.ripe.net/manage-ips-and-asns/db/support/documentation/ripe-database-documentation/rpsl-object-types/4-2-descriptions-of-primary-objects/4-2-4-description-of-the-inetnum-object>. [Last accessed 14-01-2019].
- [105] “4.2.9 Description of the INET-RTR Object.” Electronical version on: <https://goo.gl/tZToxh>. [Last accessed 14-01-2019].
- [106] “RIPE Database Documentation.” Electronical version on: <https://www.ripe.net/manage-ips-and-asns/db/support/documentation/ripe-database-documentation>. [Last accessed 13-01-2019].
- [107] “PyQt5.” Electronical version on: <https://pypi.org/project/PyQt5/>. [Last accessed 22-01-2019].
- [108] A. Quina and L. Stavliotis, “Sparta.” Electronical version on: <https://sparta.secforce.com/>. [Last accessed 14-01-2019].
- [109] “Nikto 2.” Electronical version on: <https://cirt.net/Nikto2>. [Last accessed 22-01-2019].
- [110] “A Simple Network Management Protocol (SNMP).” <https://tools.ietf.org/html/rfc1157>.
- [111] “X11 Protocol.” Electronical version on: <https://www.x.org/releases/X11R7.5/doc/>. [Last accessed 22-01-2019].
- [112] “Automation of Vulnerability Assessments with OpenVAS,” Jul 2013. Electronical version on: <https://elasticsecurity.wordpress.com/2013/07/18/automation-of-vulnerability-assessments-with-openvas/>. [Last accessed 14-01-2019].

- [113] “Using the MSFcli Interface.” Electronical version on: <https://www.offensive-security.com/metasploit-unleashed/msfcli/>. [Last accessed 14-01-2019].
- [114] “REST Architecture.” Electronical version on: http://roy.gbiv.com/pubs/dissertation/rest_arch_style.htm. [Last accessed 22-01-2019].
- [115] “Teredo: Tunneling IPv6 over UDP through Network Address Translations (NATs).” <https://tools.ietf.org/html/rfc4380>.
- [116] “Developers Guide - intelmq - Data-Harmonization.” Electronical version on: <https://intelmq.readthedocs.io/en/latest/Data-Harmonization/>. [Last accessed 17-01-2019].
- [117] “Incident Classification/Incident Taxonomy according to eCSIRT.net - adapted.” Electronical version on: <https://www.trusted-introducer.org/Incident-Classification-Taxonomy.pdf>. [Last accessed 17-01-2019].
- [118] “Zone-H.” Electronical version on: <http://www.zone-h.org/>. [Last accessed 22-01-2019].
- [119] Y. Y. Rekhter, “Address Allocation for Private Internets.” <https://tools.ietf.org/html/rfc1918>.
- [120] “Transparent DNS proxies.” Electronical version on: <https://www.dnsleaktest.com/what-is-transparent-dns-proxy.html>. [Last accessed 18-01-2019].
- [121] P. Mockapetris, “DOMAIN NAMES - IMPLEMENTATION AND SPECIFICATION.” <https://www.ietf.org/rfc/rfc1035.txt>.
- [122] “RPC: Remote Procedure Call Protocol Specification Version 2.” <https://tools.ietf.org/html/rfc5531>.
- [123] “[MS-SMB]: Server Message Block (SMB) Protocol.” Electronical version on: <https://msdn.microsoft.com/en-us/library/cc246231.aspx>. [Last accessed 22-01-2019].
- [124] “[MS-PTPT]: Point-to-Point Tunneling Protocol (PPTP) Profile.” Electronical version on: <https://msdn.microsoft.com/en-us/library/dd644854.aspx>. [Last accessed 22-01-2019].
- [125] “Malware FAQ: Microsoft PPTP VPN.” Electronical version on: <https://www.sans.org/security-resources/malwarefaq/pptp-vpn>. [Last accessed 04-02-2019].
- [126] “ASSIGNED NUMBERS.” <https://www.ietf.org/rfc/rfc1700.txt>.

- [127] J. C. K. Keane, “Exploiting PHP PCRE Functions.” Electronical version on: <http://www.madirish.net/402>. [Last accessed 14-01-2019].
- [128] “Lightweight Directory Access Protocol (LDAP): The Protocol.” <https://tools.ietf.org/html/rfc4511>.
- [129] Brianwrf, “Oracle Weblogic Server 10.3.6.0 / 12.1.3.0 / 12.2.1.2 / 12.2.1.3 - Deserialization Remote Command Execution,” Apr 2018. Electronical version on: <https://www.exploit-db.com/exploits/44553>. [Last accessed 14-01-2019].
- [130] A. S. Santos, “CUF explica ataque informático à Comissão de Proteção de Dados.” Electronical version on: <https://goo.gl/tF4crF>. [Last accessed 14-01-2019].
- [131] Markruss, “PsExec - Windows Sysinternals.” Electronical version on: <https://docs.microsoft.com/en-us/sysinternals/downloads/psexec>. [Last accessed 19-01-2019].
- [132] “Telnet Protocol Specification.” <https://tools.ietf.org/html/rfc854>.
- [133] “Border Gateway Protocol (BGP).” <https://tools.ietf.org/html/rfc1105>.
- [134] “What Is BGP Hijacking?.” Electronical version on: <https://www.cloudflare.com/learning/security/glossary/bgp-hijacking/>. [Last accessed 19-01-2019].
- [135] B. Padlina, “NTPd - Remote Buffer Overflow,” Apr 2001. Electronical version on: <https://www.exploit-db.com/exploits/20727>. [Last accessed 04-02-2019].
- [136] B. A. Sassani, C. Abarro, I. Pitton, C. Young, and F. Mehdipour, “Analysis of NTP DRDoS attacks’ performance effects and mitigation techniques,” in *2016 14th Annual Conference on Privacy, Security and Trust (PST)*, pp. 421–427, Dec 2016.
- [137] “Protocol standard for a NetBIOS service on a TCP/UDP transport: Detailed specifications.” <https://tools.ietf.org/html/rfc1002>.
- [138] “Monero.” Electronical version on: <https://www.getmonero.org/>. [Last accessed 16-01-2019].
- [139] “The Penetration Testing Execution Standard.” Electronical version on: <http://www.pentest-standard.org/>. [Last accessed 26-01-2019].
- [140] “DiSIEM Project.” Electronical version on: <http://disiem-project.eu/>. [Last accessed 26-01-2019].
- [141] “CVE-2019-6491.” <http://cve.mitre.org/cgi-bin/cvename.cgi?name=2019-6491>. [Last accessed 22-01-2019].