UNIVERSITY POLITEHNICA OF BUCHAREST

Doctoral School of Electronics, Telecommunications and Information Technology

Decision No. 940 from 21.10.2022

Ph.D. THESIS SUMMARY

Ing. Cristian PASCARIU

UTILIZAREA TEHNICILOR INTELIGENȚEI ARTIFICIALE ÎN ANALIZA MALWARE

USING ARTIFICIAL INTELLIGENCE TECHNIQUES FOR MALWARE ANALYSIS

THESIS COMMITTEE

Prof. Dr. Ing. Ion MARGHESCU University Politehnica of Bucharest	President
Prof. Dr. Ing. Ioan BACIVAROV University Politehnica of Bucharest	PhD Supervisor
Prof. Dr. Ing. Mircea POPA University Politehnica of Timișoara	Referee
Prof. Dr. Ing. Gheorghe ȘERBAN University of Pitești	Referee
Prof. Dr. Ing. Paul ȘCHIOPU University Politehnica of Bucharest	Referee

BUCHAREST 2022

Content

Chapter 1

Introduction

Information technology is one of the fields that has experienced significant development in recent decades, with significant contributions to the improvement and modernization of communication and data processing in digital format. Currently, computer systems have become indispensable due to their extensive use in modern society. *Dependability* [1] of information systems is a very important element and has gained special interest in recent times because of the economic impact which can be caused by the improper operation of these systems.

Cyber security is a field that is based on three important factors: confidentiality, integrity and availability of information processed by a computer system. Risks regarding security incidents and breaches have a major impact on the smooth functioning of organizations, but more importantly on the standard of living since information technology affects the way individuals communicate, learn, and think. Existing methodologies and solutions for detecting and analyzing cyber-attacks and computer viruses have limitations that have been identified and cataloged because of reported security breaches that have been publicly documented.

1.1 The field of the doctoral thesis

For the presentation of the field of the doctoral thesis, I approached the concept of digitalization, which can be defined as a series of processes that were based on physical documents and were migrated and implemented digitally. Initially I established the degree of importance and the critical aspect of computer systems in modern society, in the next stage I approached the field of information security and computer systems. I have documented the fundamental aspects of this domain as confidentiality, integrity, and availability of information, and confirmed that *any action that degrades or negatively affects one or more of these aspects is considered a cyber threat*. Later I approached the substantial development of new fields such as the *Internet of Things* (IoT), *Cloud* and *Smart Cities*, which are modern concepts that have changed the approach to cyber security. In the final stage of the presentation of the field of the doctoral thesis, I highlighted the current challenges, which consist in the limitation of the existing capabilities and solutions responsible for detection and prevention of cyber-

attacks, the lack of a qualified security experts in the field and the high level of complexity of modern malware.

Both public institutions and private organizations use computer systems to provide services based on the processing of information in digital format. They have replaced traditional document-based processes and physical forms with digital document processing workflows. Gartner defines the digitalization phenomenon as "*the use of digital technologies to change the way we provide services*" [3].

Starting from the premise that computing systems have become an integrated and indispensable component in the way organizations provide services, it entails that the operational safety of these systems, both in the smooth running of normal activities and to protect data, has become a strategic objective, of utmost importance. The security of information and computing systems is based on three fundamental aspects that are defined by the ISO/IEC 27000 standard [4]:

- *Data confidentiality*: "information is not made available or disclosed to unauthorized persons, entities or processes";

- *Data integrity*: represents the "accuracy and completeness of data" stored or processed by a computing system;

- *Data availability*: the information is "accessible and usable upon request by an authorized entity" [5].

Any factor that degrades or has a negative impact on one or more factors of information security is considered a *cyber threat* [5]. Based on the analysis presented in the ENISA Threat Landscape 2021 report, 8 main groups of cyber threats are identified sorted based on risk (Table 1.1).

Table 1.1 *Top 8 cyber threats for 2020-2021 (according to ENISA) [16].*

Importance level	Type of cyber threat
1	Ransomware
2	Malware
3	Cryptojacking
4	Threats against e-mail
5	Threats against data
6	Threats against availability and integrity of services
7	Misinformation
8	Non-malicious threats

1.2 The scope of the doctoral thesis

Cyber security has become an important field that underlies the functioning of organizations and institutions today. The purpose of the doctoral thesis is to carry out an analysis of the current challenges, both technical and organizational, based on which I developed, tested, and proposed solutions to improve the capabilities and processes that are part of the cyber defense strategy.

In the context of challenges related to the implementation of organizational measures, the main objective is to identify threat-oriented cyber security methodologies based on which processes and workflows can be implemented to improve the efficiency with which an organization responds to security incidents, reduce the risk of security breaches as well as their impact on the organization.

Cyber-attack detection and prevention capabilities are an important component of the defense strategy. In this PhD thesis I identified and analyzed the limitations and opportunities of malware detection, based on which I proposed complementary solutions for identifying cyber-attacks using artificial intelligence algorithms.

The purpose of the research carried out during the doctoral studies consisted of the development of solutions aimed at minimizing the risks of cyber-attacks and mitigating the impact that security breaches can have on the confidentiality, integrity, and availability of data. A secondary goal is to highlight the opportunities for knowledge transfer from the academic domain to the commercial environment, but also to attract research funds and grants from the commercial environment to the academic domain, building a collaborative and symbiotic relationship with benefits for both parties. In terms of knowledge transfer from academia, advanced data processing algorithms can be implemented to solve the real problems that organizations of all sizes in all sectors face in the field of information security.

1.3 The content of the doctoral thesis

The doctoral thesis is structured in 5 chapters together with the list of tables, the list of figures and the list of abbreviations used. The paper ends by presenting the bibliographic references.

In **Chapter 1** I presented the introductory part of the field of the doctoral thesis related to the security of information and computer systems. I made a presentation of the evolution of computing systems and the current state that defines them. I indicated the factors that contributed to the major evolution of computing systems and the risks and impact that may result from incidents and security breaches. I conducted a study based on which I identified the current challenges that organizations and companies have in the field of information security, one of the most important, being the lack of resources and experts in the field. In parallel, I did another study on the evolution of libraries that implement artificial intelligence algorithms, and based on this, I defined the main objective of the doctoral thesis as the analysis and implementation of intelligent algorithms for automating and simplifying the processes of detection and analysis of cyber-attacks and malware. The chapter continues with the presentation of the main objectives of the thesis.

Chapter 2 elaborates on the concept of information security by addressing core components such as data confidentiality, integrity, and availability. Also in this chapter, I have classified the types of cyber-attacks and computer viruses. The chapter continues with an analysis of the paradigm shift in securing computing systems, which in the past

was implemented at the network perimeter level, where systems within the network perimeter are considered secure, and those outsides are considered hostile and must be checked. I continued with an in-depth presentation of new concepts such as Cloud Computing and IoT, where the perimeter-based defense paradigm can no longer be successfully applied. I continued by conducting an analysis of risks and security techniques for Smart Cities. The chapter continues with a survey of methodologies proposed by organizations whose purpose is to develop security standards, policies, and controls. Based on this study, I proposed two methodologies that represent the foundation for the following chapters.

Chapter 3 begins with a survey of existing approaches to cyber defense strategies based on which I will primarily focus on the threat-oriented defense strategy. I propose a reference architecture for an isolated environment using a virtualization solution for simulating and analyzing malware and cyber-attacks based on open-source technologies. Malware as well as other offensive techniques will be used to simulate attacks in a controlled manner; I will continue with the analysis of cyber-attack detection capabilities and techniques at both the network level and host levels. Based on simulated cyber-attacks, I aim to identify the limitations of detection techniques, but also the opportunities where artificial intelligence techniques can be successfully applied.

Chapter 4 presents a thorough analysis of the opportunities where algorithms from the field of artificial intelligence can be applied to solve classification problems. I identified three stages: the pre-infection stage, the infection stage, and the post-compromise stage. I continued by analyzing the characteristics of phishing websites and proposed a solution to detect them based on similarity analysis in an automated way. For the infection stage, I focused on analyzing the process hierarchy to discover patterns that identify when legitimate computing processes are used to infect a system and continued with the development of a neural network-based solution capable of identifying malware applications. For the post-compromise stage, I performed malware analysis in an isolated virtual environment of a Ransomware virus from the Stampado family. Based on the indicators obtained from the analysis, I implemented a honeypot solution for detecting the spread of Ransomware viruses in a computer network and for detecting patient 0, more precisely the first computer system that was infected.

Chapter 5 is dedicated to the main contributions and results obtained during the research, analysis, and development of solutions for detecting cyber-attacks based on algorithms from the artificial intelligence field. Finally, I have identified and documented some development directions that can bring improvements to the solutions presented in this doctoral thesis.

Chapter 2

Security of computer systems

In **Chapter 2** I conducted a thorough analysis of cyber risks and threats based on incidents and security breaches that occurred in the last decade and that were documented and released to the public. I continued by carrying out a study on the impact that these security breaches had on ordinary citizens and on organizations, where due to the phenomenon of digitization, security incidents can reduce or even stop everyday activities, which can lead to potential financial impact. I also conducted an analysis of existing methodologies that propose techniques for the detection and analysis of attacks and computer viruses. I have proposed approaches that can be adopted by organizations to identify, analyze, and prevent cyber-attacks based on existing methodologies developed by the global community of security analysts. The "Cyber Kill Chain" is a methodology used to consider cyber-attacks as a series of steps that the attacker must go through for the attack to be successful. The second methodology presented is the MITRE ATT&CK: this consists of a collection of attack techniques which are classified by attack tactics.

2.2 Analysis of SSL libraries for IoT devices

Among the personal contributions noted in [34] is the analysis of the risks and vulnerabilities of IoT platforms. Through a series of tests and reviews of available libraries I identified a vulnerability caused by a poor implementation of the standard SSL/TLS protocols. In situations where digital certificates are revoked because of a security breach, there are protocols such as the Certificate Revocation List (CRL) that can be used to void the trust relationship. RFC 5280 [44] describes a CRL as "a marked and signed data structure that a Certificate Authority (CA) or CRL issuer periodically issues to communicate the revocation status of affected digital certificates". I have identified a library for IoT solutions that does not implement protocols such as CRL and/or OSCP. In the absence of these, the IoT devices will continue to initiate communications with a compromised service whose digital certificate has been revoked. This represents a risk and a vulnerability [45].

The initial analysis was performed in 2016, revisiting the hardware platform in 2021, it is no longer available [46] and the manufacturer recommends using another

updated platform. From the software platform and libraries perspective, they have not been updated since 2017 according to the Github page where the source code can be found [47]. This leads to the fact that several IoT products that leverage these libraries, after a period of operation of several years, although functional, represent opportunities for cyber-attacks [48].

2.3 Using IoT platforms to secure computer networks

Risks and vulnerabilities also exist in IoT platforms, thus expanding exposure to cyber-attacks. At the same time, due to the increase in computing power and resources available in IoT devices, as well as their low cost, this creates the opportunity for the emergence of IoT security devices.

In the article "*Network security monitoring with embedded platforms*" [52] I contributed to the development of a solution [53] for the detection of cyber-attacks at the network level based on the Raspberry Pi single board computer [54]. Using the Python programming language [55] and the Scapy library [56] for traffic and packet analysis, I implemented a solution capable of detecting ARP Spoofing (Address Resolution Protocol) attacks [57] through which an attacker who already has network access attempts to impersonate the router in order to intercept traffic from and to certain hosts. The detection technique is based on identifying ARP replies that have a different gateway address than the original one.

2.5 Organizational measures to improve cyber security

The new industry trends such as digital transformation, cloud technologies and a sophisticated threat landscape are forcing organizations to rethink the functions of each role in their security teams. With billions of people around the world working from home, changes in the daily practice of implementing cybersecurity are accelerating. Organizations are moving away from traditional network perimeter defense to more effective strategies like *Zero Trust*. This transformation brings technological change and opens questions about what people's roles and responsibilities will look like in this new digital world.

At the same time, continuous delivery models require security teams to be more closely involved during application planning and development to effectively manage cyber risks (compared to traditional "end-2-end" security approaches). This requires security professionals to better understand the business context and collaborate more closely with non-security stakeholders [72].

Chapter 3

Detection of cyber-attacks

In **Chapter 3** I focused on evaluating the cyber defense strategies and the detection and analysis capabilities. From a defense strategy point of view, I have concluded that the traditional, vulnerability-oriented approach has limitations in detecting modern threats. I proposed the use of the threat-centric defense strategy, which involves a continuous process for detection and analysis, as well as identifying opportunities for improvement.

I continued my analysis by assessing the effectiveness of the security detection capabilities as well as their limitations. Based on simulated cyber-attacks in a controlled and isolated environment led me to identify potential limitations to traditional rule-based capabilities. Although these capabilities can detect already known viruses, attackers can modify or create new viruses, for which the degree of detection is low.

Through a detailed study of cyber threats, I highlighted the observation that cyber-attacks, due to the variety of techniques used and complexity, there is no single method or system by which they can be detected. The successful detection of intrusions is based on several capabilities that are integrated with each other and work in tandem. These can be classified into two broad categories: network level and host level capabilities. I found that a defense strategy based on automated detection capabilities alone is not enough, and an organization needs cybersecurity experts to perform analysis by correlating multiple data sources.

3.2 Simulating cyber-attacks

I have implemented a cyber-attack and malware analysis simulation solution that will allow both the simulation of attacks in a controlled environment and assessing of the effectiveness and of the detection and analysis capabilities. This technique is often used by cyber security specialists who analyze and study malware behavior [83].

As a platform I opted for the use of a VMware Workstation which is a virtualization solution [84]. Figure 3.2 illustrates the attack simulation and analysis environment with all its components. I simulated a computer network by creating virtual machines with both Windows and Linux operating systems. Connected to the same network is also a Kali Linux virtual machine [86], with the role of simulating the

attacker's system. Kali Linux is used by security experts to simulate attacks on computing systems and applications.

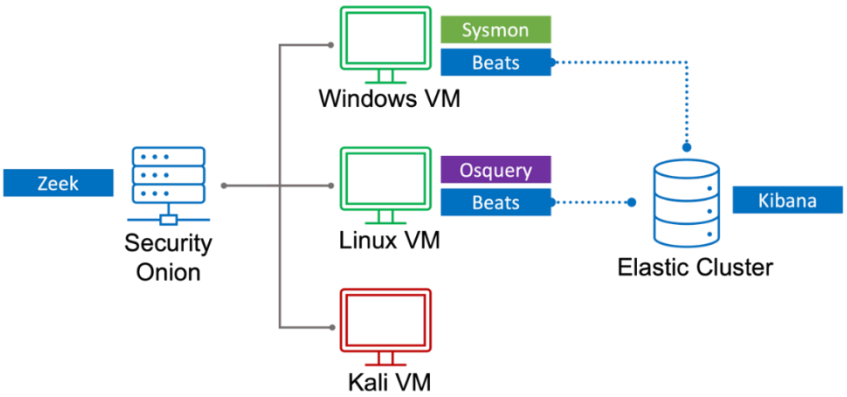

Figure 3.2 *Virtual attack simulation and analysis environment.*

On the Windows VM I installed and configured the Sysmon program [87], which monitors important system events such as new process creation, WMI commands, and network-level activity that is initiated by processes. Osquery [90] is a host-based event and configuration settings monitoring and detection solution for Linux systems. For network traffic monitoring I used the Security Onion solution [91]. One of the important IDS (Intrusion Detection System) solutions that is part of the Security Onion platform is Zeek [92], also known as Bro. I used the Elastic Stack [93] solution, for its capability to collect and store logs, this gives security analysts the opportunity to analyze security-related events. The Elastic platform is based on three components: Elasticsearch [94] the storage component, Kibana [95] the web interface, and Beats [97] the agents installed on computing systems to collect the logs.

3.3.3 Detection of cyber-attacks based on anomalies

I simulated a cyber-attack using a backdoor virus in the isolated environment, then continued by performing network-level analysis of the recorded data based on the traffic generated by the virus. In Figure 3.15 it can be seen how data connections occur at constant time intervals, as well as the constant amount of the number of packets. And in Figure 3.16 it can be notices how events are recorded every 10 seconds. These pulses are generated from the infected system, which tries to connect to the server or infrastructure controlled by the attacker.

I continued the analysis at the host level by performing a search to identify malicious processes that want to steal authentication details from other processes using a specific access rights code, namely 0x1010. In Figure 3.31 I identified a malicious process that uses this technique to extract passwords and unique codes from the *lsass* process.

Figure 3.15 *Anomaly detection based on time intervals between connections.*

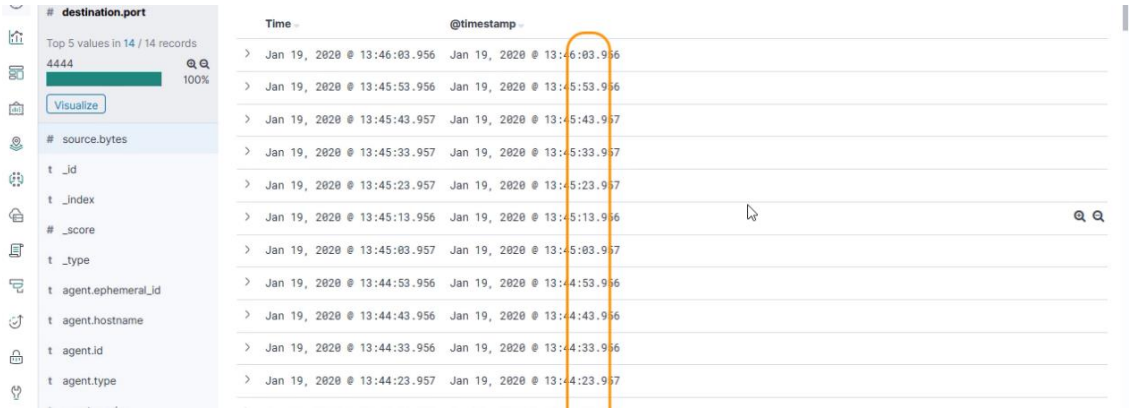

Figure 3.16 *"Beaconing" traffic detection.*

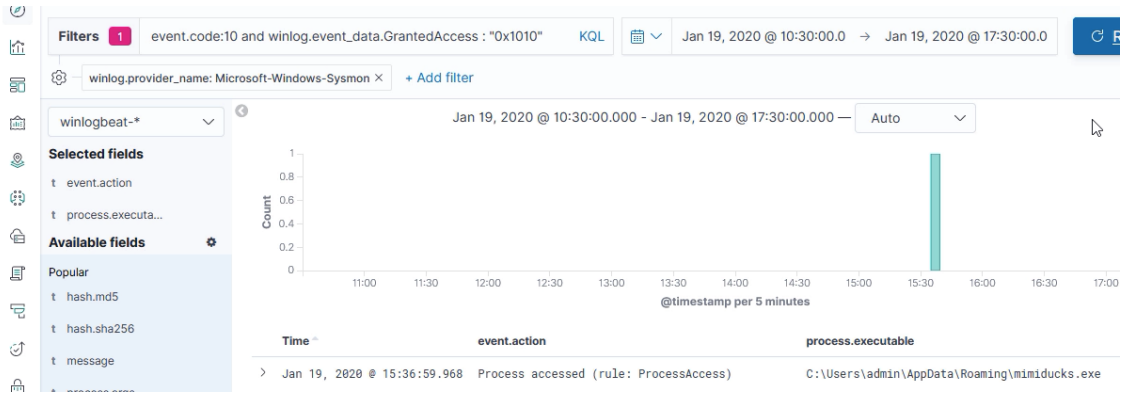

Figure 3.31 *Identifying a malicious process through access rights.*

Additionally, in Figure 3.34 I have identified a sequence of PowerShell commands nested within each other to avoid detection and hinder analysis efforts. The first command contains a set of commands encoded using the Base64 algorithm. The following event is generated for the same command, only it is decoded giving a security analyst the opportunity to extract indicators, based on this I concluded that the virus is used to steal passwords from system memory.

Figure 3.34 *Malware execution detection using PowerShell commands.*

3.5 Correlating system-level events with network-level events

In order to determine the purpose and impact of a cyber-attack, it is necessary to identify and document all the actions performed during the attack, which involves correlating the events and indicators recorded at the network level with those at the host level.

In Figure 3.50 I identified a sequence of suspicious events, starting from an indicator designed to detect suspicious processes whose corresponding file on disk was removed, I identified a process with a suspicious name but also which connects to an IP address on a non-standard port. Based on this address I performed a further search where we identified other legitimate processes that were executed to facilitate the attack. Correlating system-level events with network-level events, enables security analysts, which are conducting the investigation to understand the big picture. This extensive data set, as well as the ability to leverage filters and conditions, enables security analysts to identify the complete infection chain.

```
osquery> SELECT DISTINCT p.pid, p.name, p.path, s.remote_address, s.remote_port FROM processes AS
 p JOIN process_open_sockets AS s ON p.pid=s.pid WHERE p.on_disk=0;
+-------+-------+------------+----------------+-------------+
| pid   | name  | path       | remote_address | remote_port |
+-------+-------+------------+----------------+-------------+
| 10599 | PwUcb | /tmp/PwUcb | 192.168.253.162 | 4444        |
| 10599 | PwUcb | /tmp/PwUcb | 192.168.253.162 | 4433        |
+-------+-------+------------+----------------+-------------+
osquery> SELECT DISTINCT p.pid, p.name, s.remote_address, s.remote_port FROM processes AS p JOIN
process_open_sockets AS s ON p.pid=s.pid WHERE s.remote_address == "192.168.253.162";
+-------+--------+----------------+-------------+
| pid   | name   | remote_address | remote_port |
+-------+--------+----------------+-------------+
| 10347 | python | 192.168.253.162 | 4444        |
| 10348 | bash   | 192.168.253.162 | 4444        |
| 10599 | PwUcb  | 192.168.253.162 | 4444        |
| 10599 | PwUcb  | 192.168.253.162 | 4433        |
+-------+--------+----------------+-------------+
```

Figure 3.50 *Identifying suspicious processes based on connections with an IP address.*

Chapter 4

Using Artificial Intelligence techniques in malware analysis

In **Chapter 4**, following a study of solutions based on artificial intelligence algorithms, I concluded that these solutions focus more on the performance of the algorithm rather than the implementation itself. This conclusion is also supported by other authors who performed similar research and demonstrated the low efficiency of these solutions when the data set is different from the training one. As a result, I have proposed a set of solutions that adhere to the principles of intelligent systems, such as making decisions under uncertainty and forming inferences using a built-in knowledge base. This is aimed at helping security analysts by automating the manual tasks that they have to perform during an investigation.

I also concluded that any machine learning algorithm is a module within a larger solution rather than algorithm itself is not the solution itself. Conducting further experiments I observed the high degree of importance that the feature selection of an attack analysis task has, as these features become the input data set for the machine learning algorithms.

4.1 Study on phishing website detection techniques

Phishing is a technique aimed at misleading and deceiving users of a web service or digital platform through a web page similar to the original one, but with the aim of stealing login details. The study of existing solutions in the field led to the identification of several approaches to combat cyber-attacks that use phishing websites to steal user credentials.

Uniform Resource Locator (URL)-based detection is a set of techniques that focus on key indicators extracted from the URL, such as domain, subdomain, protocol, web directory, and other parameters of the URL. Attackers use a technique called "typosquatting" [125] to deliberately register similar domains to legitimate ones where a single character (a letter or number) may be different. This technique relies on errors that users may make when typing a website address.

Another worrisome technique is when attackers will create a legitimate subdomain for their malicious domain [126]. Web browsers will shorten the length of the full URL and victims will think they are visiting the legitimate service when in fact they are visiting the malicious one. In Figure 4.2 I illustrated a phishing website with a very long address, and the registered subdomain contains the name of the legitimate Paypal service.

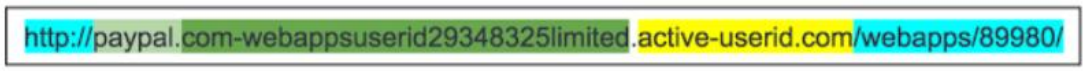

Figure 4.2 *Malicious domain using legitimate domain as a subdomain.*

4.2 Solution to detect phishing pages based on similarity analysis

I proposed a novel method for detecting phishing websites based on the similarity analysis between a legitimate page and a phishing page that tries to imitate it. This analysis is performed at both the URL and content levels. For the URL analysis I used an algorithm for identifying the string similarity ratio (Longest Common Subsequence - LCS) [130] to detect similar domains. Content-based analysis is used to identify phishing web pages that mimic legitimate ones based on the similarity of keywords in the title, as well as the presence of keywords specific to authentication pages. Another advantage of the proposed solution is that it does not require a large dataset for training, all the required information consist of the URL and content of legitimate pages. Based on these, all suspicious websites will be scanned, their features will be extracted and compared with the features of legitimate pages. The lack of need to train machine learning algorithms and the small set of features that are extracted and processed decrease the amount of computing resources required, which makes the solution fast.

Table 4.1 illustrates the features extracted for URL analysis. One of the most important is similarity ratio at the domain level. In Python I used the *Difflib* module to generate string-level similarity percentage.

Table 4.1 *URL-based features.*

Feature	Description
Domain similarity	Similarity ratio between the legitimate and the suspect domain
Subdomain	If the legitimate domain is used as a subdomain

Table 4.2 contains the content-based features used by the proposed solution to detect whether there are keywords in the content of the scanned website that match the legitimate service, but also contains typical elements of a login page.

Table 4.2 *Content-based features.*

Feature	Description
Title keywords	Individual keywords which can be found in the title section of the website
Authentication keywords	Generic keywords which are used for authentication pages, such as “Sign in”, “Log in”, “Authenticate”, “<input type = “password”>“

4.2.3 Testing and results

Phishing websites have a short lifetime [131] ranging from a few hours to several days before they are discovered and removed by web hosting providers. As an alternative, I opted to use a list of phishing pages that are already known and test the effectiveness of detecting similar domain names.

Table 4.3 *URL detection results 0.*

Domain	Similarity ratio
googlej [.] com	95%
googled [.] com	95%
xn - oole-z7bi [.] com	59%
yahoo [.] com	63%
hotmail [.] com	57%
microsoft [.] com	52%

Table 4.3 [123] highlights the results for a set of domains compared to the “google.com” domain, which was selected as a reference domain and due to its popularity. It can be seen that known malicious domains similar to the base domain generated a percentage of over 90%, while other legitimate domains generated a lower score of 75%. Extending the tests and analysis to similar domains for google.com, I obtained about 264 unique domains that are similar to google.com. The list of domains was extracted from the online service DNStwist [132] which generates a list of misspelled domains and checks if they have been registered. These were used as the test data set. The results of the analysis are illustrated in Table 4.4. Of these, 40% of the domains achieved a similarity score greater than 90%.

Table 4.4 *Domain similarity analysis.*

Similarity (%)	No. of domains	Details
> 90%	105 (40%)	Similar domains
70% - 89%	74 (28%)	The legitimate domain is part of the scanned domain
<60%	85 (32%)	Unicode domains

To test the effectiveness of content-based detection, I used the *Social Engineer Toolkit* (SET) [133], which is a solution for performing security assessments focused on *social engineering*. For testing purposes, the page generated by the Social Engineer Toolkit was analyzed against the legitimate Gmail login page. The results of this analysis are shown in Table 4.5.

***Table 4.5** Content detection analysis.*

Content-Based Features	Malicious Score
Title Similarity	100%
Authentication keywords	(3/5)

Testing continued focused on the effectiveness of the solution for a more realistic scenario. The *Easychair* platform was chosen as the target. In this regard I registered the domain *Easychalr.org*, where the letter "i" is replaced by the letter "l", in such a way that a user cannot detect the error at first glance. Then I used the SET tool to create a copy of the authentication page of the Easychair platform.

I continued by performing the similarity analysis using the proposed solution from the command line, where the results were positive based on the URL similarity analysis which yielded a score of 92%, as illustrated in Figure 4.5.

***Figure 4.5** URL similarity analysis report.*

Regarding the content analysis, the following results were identified (as illustrated in Figure 4.6):

- a title similarity was identified with a score of 97%;
- followed by the identification of the keywords ("login", "log in") and the parameters specific to the authentication pages (*password input field*).

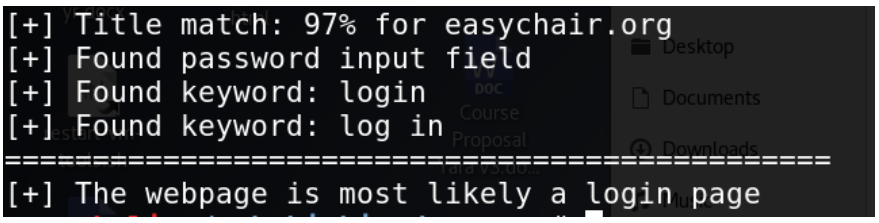

***Figure 4.6** Content similarity analysis report.*

4.3 Ransomware analysis (Stampado)

Before proposing a mechanism for detecting ransomware infections and how they spread across the computer network, I performed malware analysis on a type of ransomware called Stampado.

I used the VirusTotal.com service, for the static analysis part, based on which I noticed the high detection score. According to Figure 4.7, a total of 54 out of 70 antivirus solutions detected the file as malicious.

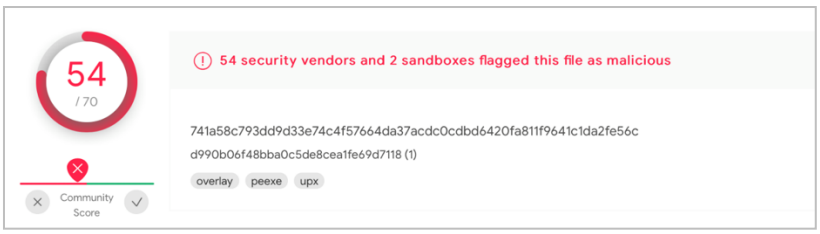

Figure 4.7 *Detection score for ransomware Stampado.*

Using the Hybrid-Analysis sandbox solution for dynamic analysis, it was identified, as shown in Figure 4.11, that the virus was initially delivered using an infected document. Another important indicator is the name: scvhost.exe, this is a similar name to a legitimate svchost.exe process. The way I determined that the process is not the legitimate one is based on the location on the disk as well as the similar name.

Figure 4.11 *Process execution hierarchy within Stampado infection.*

I continued to analyze the virus in the isolated test environment, I identified two mechanisms of propagation. The first, illustrated in Figure 4.14, uses the *autoplay* function [139] of the operating system to launch programs via the autorun.inf file: depending on system settings and policies, the virus can be executed automatically when peripheral storage devices such as USB sticks are connected to the computer.

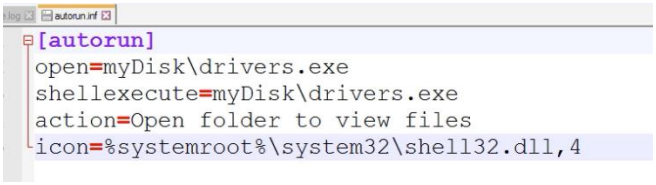

Figure 4.14 *Autorun.inf file generated by ransomware for automatic execution.*

The second propagation mechanism, illustrated in Figure 4.15, uses shortcut files with the .LNK extensions [140], the target parameter is used to specify the location on disk of the actual file for which the shortcut was created, but in the case of the ransomware virus, the path is replaced by a command which will execute the ransomware virus on machines which have not yet been infected. In case of file shares, when another user accesses this their machines will get infected as well.

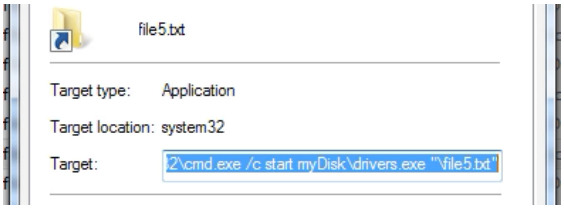

Figure 4.15 *LNK file used to execute ransomware.*

4.4 Honeypot solution for ransomware detection

I proposed a ransomware detection solution [144] designed to identify ransomware attacks that spread through a computer network. This platform will host worthless documents and data, tricking ransomware viruses into infecting them and creating the opportunity for analysts to investigate the infection. The solution, illustrated in Figure 4.18, is composed of standard libraries and services that can emulate file sharing services.

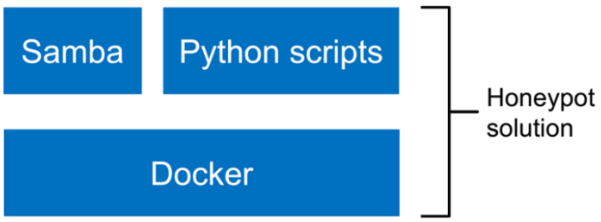

Figure 4.18 *The hardware and software platform for the honeypot solution.*

To provide an additional layer of protection against infected files and preserve possible malware samples, the entire solution will be deployed in a container using Docker. In the container there will be two modules to implement the solution: the Samba service [147], and a set of scripts developed using the Python programming language, which implement the logic for detecting ransomware infections. The Samba service was configured by enabling the VFS Full Audit module [148] to record operations performed on shared files.

The ransomware infection detection mechanism is implemented in a program written in the Python programming language, which manages all the detection logic. One important concept is the use of decoy files to be sacrificed in the event of a ransomware infection. To detect ransomware infections, the detection program will use information both from decoy files that will be stored and shared via the Samba service, and from logs containing file access operations (Figure 4.20). On initialization, the program will generate a series of “decoy” files along with their hash values. The latter are necessary to check the integrity of the files; when the content of a file is changed, generating the hash value will produce a different value.

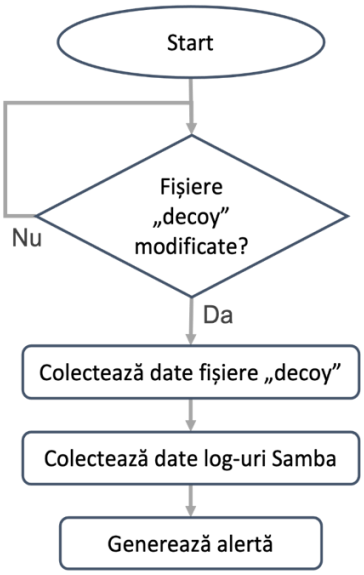

Figure 4.20 *Detection logic workflow.*

The program will periodically monitor both the existence of these decoy files and their integrity. The moment they have been deleted or modified, then a series of detection techniques will commence. The program will scan common extensions used by ransomware viruses. If the “decoy” files have been modified, the analysis of the logs continues, identifying the user and the IP address of the system that connected to the Samba service and performed these actions. In this way, patient 0 can also be identified.

To ensure the effectiveness of the proposed solution, a series of tests were performed to ensure that the final solution can detect the infection stage of Stampado, discussed in detail in Chapter 4.3. Figure 4.21 illustrates the alert generated after the ransomware infection and shows the highlights: one of the decoy files was deleted, which initiated automatic analysis of Samba logs where the “.locked” extension was identified and also patient 0, the user and the IP address of the virtual machine from which the infection originated.

```
Ransomware Alert!
--------------------------------------------------
[!] Decoy file removed: file7.txt
[!] Ransomware extension detected: *.locked
[!] Patient 0: win7-victim (192.168.55.128) @ 08:19:39
```

Figure 4.21 *Ransomware detection alert.*

I continued the analysis by inspecting the logs to verify the correctness of the information in the generated alert. In Figure 4.22 both the modification of “decoy” files and the generation of files with the extension “.locked”, which is an extension used by the Stampado virus, were identified.

```
08:19:39 : win7-victim|192.168.55.128|RShare|pwrite|ok|Mallory/5B294E575745FABD9D192F2E96BB8FF8685F.locked
08:19:39 : win7-victim|192.168.55.128|RShare|pwrite|ok|Mallory/file7.txt
08:19:39 : message repeated 2 times: [ win7-victim|192.168.55.128|RShare|pwrite|ok|Mallory/file7.txt]
08:19:39 : win7-victim|192.168.55.128|RShare|rename|ok|Mallory/file7.txt|Mallory/~hfspbvz.tmp
08:19:39 : win7-victim|192.168.55.128|RShare|rename|ok|Mallory/~hfspbvz.tmp|Mallory/~mrzdacr.tmp
08:19:39 : win7-victim|192.168.55.128|RShare|rename|ok|Mallory/~mrzdacr.tmp|Mallory/~nfsjixz.tmp
08:19:39 : win7-victim|192.168.55.128|RShare|rename|ok|Mallory/~nfsjixz.tmp|Mallory/~apgkvzm.tmp
08:19:39 : win7-victim|192.168.55.128|RShare|rename|ok|Mallory/~apgkvzm.tmp|Mallory/~jrxvjqx.tmp
08:19:39 : win7-victim|192.168.55.128|RShare|rename|ok|Mallory/~jrxvjqx.tmp|Mallory/~ijgqnmo.tmp
08:19:39 : win7-victim|192.168.55.128|RShare|rename|ok|Mallory/~ijgqnmo.tmp|Mallory/~kurlkkc.tmp
08:19:39 : win7-victim|192.168.55.128|RShare|rename|ok|Mallory/~kurlkkc.tmp|Mallory/~vtzopio.tmp
08:19:39 : win7-victim|192.168.55.128|RShare|rename|ok|Mallory/~vtzopio.tmp|Mallory/~hyldufk.tmp
08:19:39 : win7-victim|192.168.55.128|RShare|rename|ok|Mallory/~hyldufk.tmp|Mallory/~wtlokvs.tmp
08:19:39 : win7-victim|192.168.55.128|RShare|unlink|ok|Mallory/~wtlokvs.tmp
08:19:39 : win7-victim|192.168.55.128|RShare|pwrite|ok|Mallory/5B294E575745FABD9D1F2F2E96BB8FF8685F.locked
```

Figure 4.22 *Detection of ".locked" files.*

Another aspect identified based on the logs recorded during the infection is the high volume of events generated in a very short period of time, as illustrated in Figure 4.24. In Python I used the Matplotlib library [150] to generate a chart, as it can be seen in Figure 4.24 (the designed code is presented in detail in the Appendix of the doctoral thesis) based on which a security analyst can visually interpret this data to determine that an anomaly exists and may initiate a more thorough investigation.

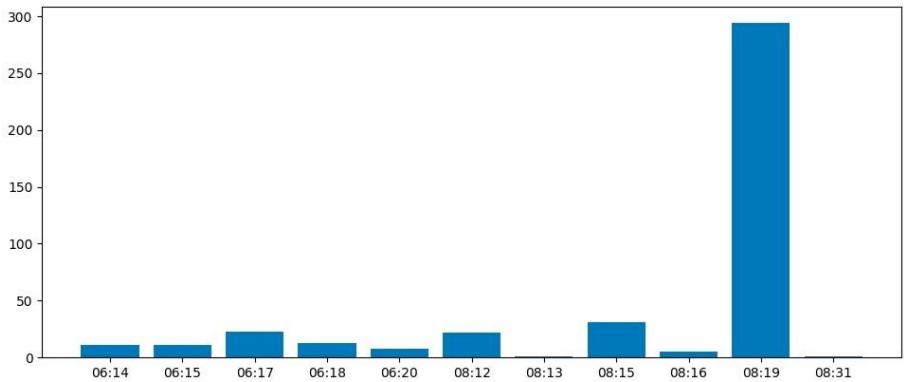

Figure 4.24 *High volume of events at a certain interval.*

4.6 Application of AI algorithms in the field of cyber security

I have studied the efficiency of AI algorithms noting that the solutions proposed in scientific articles generate favorable results only in the context of the dataset they were trained on and the type of detection they implement. Any variation in the attack techniques used will result in a reduced detection score. These conclusions are supported by other experts in the field, such as G. Apruzzese and M. Colajanni, who performed an analysis of machine learning techniques applied to the detection of security breaches, malware, and spam messages [175]. The analysis carried out by them had two objectives: the first was to evaluate the current maturity of detection solutions based on machine learning algorithms. The results obtained provide evidence that

current machine learning techniques have limitations that reduce their effectiveness for cybersecurity. All approaches require continuous retraining and careful parameter tuning and cannot be automated when different attack techniques are used.

In Figure 4.26 the traditional detection process is illustrated, where rule-based detection capabilities and solutions are used to classify a large volume of events, and a security analyst then inspects and analyzes these alerts to determine whether a cyber-attack has occurred. As a primary objective I proposed the use of artificial intelligence techniques as an intermediate capability to complement existing detection capabilities and to automate the certain analysis tasks of the domain specialist partially or fully.

Starting from the work published by Zadeh [176], I considered the characteristics that a system must meet to be considered intelligent, namely: the ability to reason based on the knowledge accumulated in the database and the ability to make decisions in case of uncertainty. Based on these characteristics, we have proposed a series of solutions that have the role of helping a security analyst in his tasks of investigating incidents and performing the analysis of malware applications at different stages of a cyber-attack.

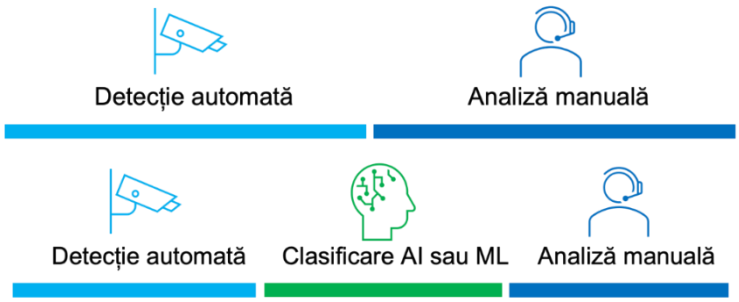

Figure 4.26 *Integrating AI techniques in the intrusion analysis process.*
(a) Traditional detection-analysis process; (b) Modern process with integrated AI

4.7 Dynamic malware analysis using neural networks

In this chapter I proposed an automated solution for dynamic malware analysis that uses a neural network to identify specific patterns where malware will use legitimate programs and utilities to install itself in an autonomous fashion. These patterns are based on process execution history. The behavioral analysis system is composed of four modules (Figure 4.27) and their role is to collect and process the data, and later it will be sent it to the artificial neural network for classification. For the task of collecting information about newly created processes, as well as the parent process, on a computer system I used the Sysmon program.

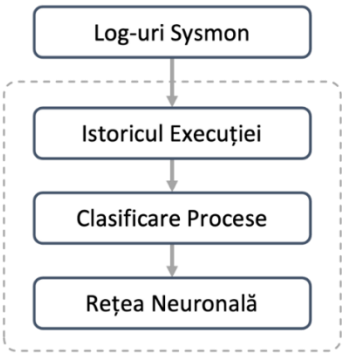

Figure 4.27 *Components of the analysis solution.*

The second module will process the events generated by Sysmon when a process is launched and will generate the execution history (Figure 4.28).

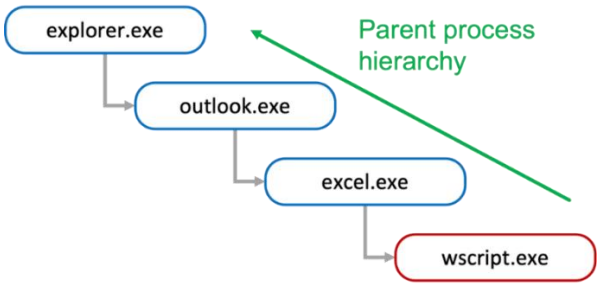

Figure 4.28 *Hierarchy of running processes [179].*

The classification process is done by an encoder which will receive as input all five process names from the previous module. The purpose of this module is to assign a risk coefficient according to the type of process, and for this purpose Table 4.9 has been created. This table displays the values assigned to risk coefficients.

Table 4.9 *Process classification [179].*

Category	Process	Value
Unknown	Unknown	0
OS (1-3)	Explorer.exe	1
	Svchost.exe	2
	noparent	1
Scripting (4-6)	Cscript.exe	3
	Wscript.exe	3
	Powershell.exe	3
	Cmd.exe	4
User (7-9)	Winword.exe	7
	Excel.exe	7
	Notepad.exe	8
	Calc.exe	9
	Iexplore.exe	7
	Outlook.exe	7

For the artificial neural network to interpret the data and make sense of it, numerical values must be sent as inputs. All the necessary information is already available, which allows the use of *feed forward* networks. The neural network, illustrated in Figure 4.29, is used to recognize the different patterns detected and processed by previous modules to determine whether a process execution falls into one of the following three categories: *Legitimate*, *Suspect* or *Virus*.

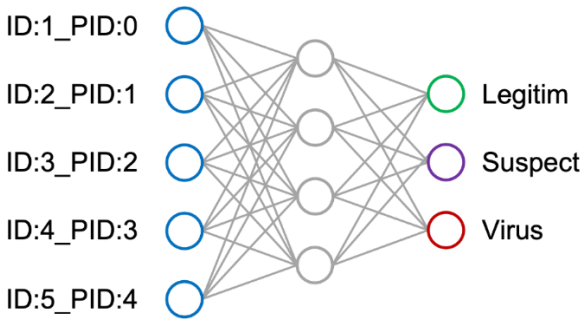

Figure 4.29 *The structure of neural network.*

To train the neural network I have leveraged a supervised learning algorithm because both the current data and the desired detection capabilities are known for creating models for training and testing. As for the algorithm itself, I opted to use backpropagation algorithm [185], which is a very popular algorithm for training feed-forward neural networks with a high efficiency rate.

4.7.6 Data model

I created three malware execution models, illustrated in Figure 4.30, with the numerical values assigned to the processes corresponding to those in Table 4.9. For these models, it can be seen how all the initial processes are part of the system (OS) category, then processes from the user category will be launched, and in the case of infected documents these will invoke and execute scripting processes.

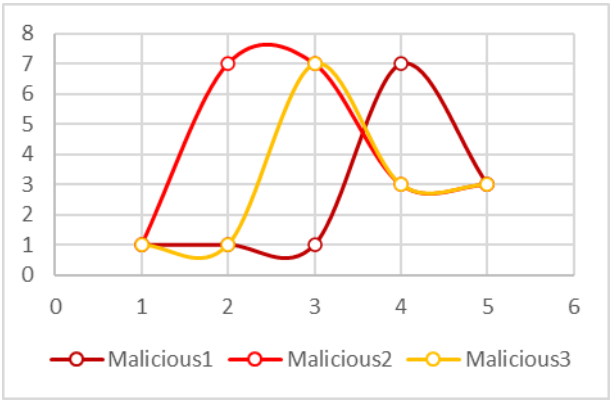

Figure 4.30 *Numerical representation of viruses.*

The real benefit of artificial neural networks is the ability to classify behavior based on different patterns. In order for the application to be effective and able to distinguish both malicious and legitimate behavior, I generated the normal behavior patterns to train the neural network (Figure 4.31).

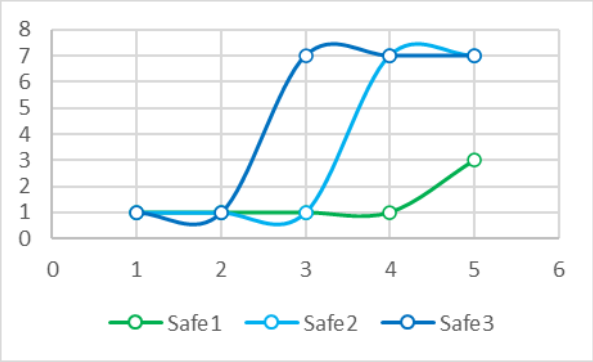

Figure 4.31 *Numerical representation of legitimate processes.*

When all the processes in the hierarchy are part of the system category, or when the processes in the user category do not execute other processes such as those in the scripting category, these models will describe the legitimate activity. The main purpose of data modeling is to identify computer viruses that are delivered via Office or PDF documents and that attempt to run malicious scripts by starting processes in the scripting category.

On the other hand, running scripting processes (such as powershell.exe or the operating system command prompt) is considered legitimate activity. The same normal behavior is considered when running user-category programs executed from the same context by the operating system.

In conclusion, using the provided models to train the neural network, the result was successful, and the artificial neural network converges. In a test environment, when executing a document infected with malware, it was successfully detected because the process hierarchy is similar to the models on which the neural network was trained. The success of this small-scale technique increases the interest for much deeper research in the field.

Starting from the concept of applying machine learning algorithms to the behavioral analysis of computer viruses, it can be concluded that one of the challenges is to model the data produced by security monitoring tools into numerically represented models. These can be learned by an artificial neural network with the goal of classifying a wide variety of security events to prevent future infections with computer viruses and other malware.

Chapter 5

Conclusions

In this chapter I have synthesized the main aspects presented in the thesis. I have highlighted the results that were obtained, the original contributions and the list of articles published during the doctoral internship. Based on the emerging cyber threats and their continuous evolution from a technical point of view, I have proposed several directions for development and improvement of the solutions presented in this paper.

5.1 Obtained results

One of the main objectives of the doctoral thesis was to identify methods of detection and analysis of cyber-attacks and viruses by leveraging artificial intelligence techniques. Neural network algorithms have been successfully applied to classification problems within the engineering domain. In the doctoral thesis I analyzed and implemented a series of solutions based on the fundamental principles of intelligent systems with the aim of improving their efficiency in terms of malware detection and reducing analysis time.

In **Chapter 2** I highlighted the most important aspects of the information security domain, such as confidentiality, integrity, and availability of data. Based on these I have defined a cyber threat as any action that affects the components of information security. I have conducted a thorough analysis of cyber risks and threats based on incidents and security breaches that have occurred in the past decade and have been documented in the publicly.

I continued by conducting a study on the impact that these security breaches had on users, such as the theft of personal data, and on organizations, where due to the trend of digitization (which means that organizations have migrated to digital solutions to manage their processes and service offering), security incidents can reduce or even stop day-to-day activities, which can materialize in financial impact.

In order to address cyber threats in an appropriate manner where the impact is minimal, I conducted an analysis on existing methodologies such as the MITRE ATT&CK Framework and Lockheed Martin Cyber Kill Chain that propose methods of detection and analysis of cyber-attacks and viruses.

Chapter 3 focuses on the detection and analysis of cyber-attacks starting from a survey of existing security capabilities. These are classified into two broad categories: network-based detection capabilities - such as intrusion detection and prevention systems (Intrusion Detection System - IDS, Intrusion Prevention System - IPS). The second category of detection capabilities is at the system level, also known as host-based detections, such as antivirus programs or event collection systems (Endpoint Detection and Response - EDR).

I continued by analyzing the effectiveness of these detection capabilities, as well as their limitations. One of the most important aspects is that the rules and indicators are able to detect viruses that are already known with high efficiency. Any threat that creates and uses new viruses can only be detected after malware analysis, after which new rules can be generated. Through a detailed study I have highlighted that cyber-attacks are beginning to abuse legitimate programs and utilities that already exist in a computer system to facilitate an infection and be able to maintain control to the remote computer system. Based on this study, I created an analysis environment using open-source technologies to collect a much wider range of events at the network and host levels, which facilitated the detection of anomalies that can identify indicators of a cyber-attack.

Chapter 4 addresses the main objective also found in the title of the PhD thesis, namely the identification and implementation of artificial intelligence algorithms to improve the efficiency of cyber-attack detection capabilities and reduce the time and resources required for malware analysis.

Based on a thorough study of existing capabilities using heuristics and algorithms in the field of artificial intelligence, I have identified three sub-domains of applications:

- Spam and phishing messages;
- Detection of intrusions and compromised systems;
- Classification of polymorphic and metamorphic malware by family.

Based on this study, I have generalized and identified three major categories in which AI algorithms can be applied. The first category is called the pre-compromise stage, in which cyber adversaries use a series of techniques to be able to initiate an attack and infect a computer system or network of computers. In the second stage, the infection stage, one or more malware components are installed and through which a cyber adversary will gain unauthorized access and control over one or more computer systems. In the post-compromise stage, the cyber adversary, based on the gained access, will execute a series of actions to consolidate its position by installing persistence techniques and avoiding detection. Also, at this stage the adversary can exfiltrate information and infect other systems in the network.

At the pre-compromise stage I conducted a study on the phishing techniques used and analyzed in detail the scenario where legitimate web pages are cloned and hosted to mislead users into entering authentication details such as usernames, passwords, and codes associated with multi-factor authentication. I developed and implemented a phishing page detection algorithm based on the similarities between the

original page and the phishing page based on similarities at the web address level, but also based on the analysis of similarities at the content level. To test the effectiveness of the proposed solution, we used a series of tools such as SET (Social-Engineer Toolkit) to generate phishing websites based on legitimate ones.

During the infection stage, I conducted a study on malware detection based on the parent-child relationship at the process execution level. Based on it, I proposed a solution that uses artificial neural networks to identify processes that are used for malicious purposes including legitimate processes. For training the network I used indicators extracted from previous investigations, and for testing I used a series of tools in an isolated environment to test the effectiveness of the proposed solution.

For the post-compromise stage, I analyzed a ransomware-type malware from the Stampado family. During analysis, I discovered two indicators of compromise through which the infection can spread to other computers on the network. For the detection of a ransomware infection in a computer network, I developed and implemented a honeypot solution based on a file sharing service to which all computer systems in a network can connect. The detection mode is based on the fact that at the time of ransomware infection, decoy files are continuously monitored, and any change in them triggers an alert.

The results obtained as a result of the doctoral research were disseminated within a series of activities carried out in a professional environment and I believe that they contributed to increasing the level of awareness regarding the implementation of cyber security in the organizational environment. Among these activities I can list:

- the facilitation of incident simulations to identify deficiencies at the process level, but also to train the team that responds to security incidents;
- developing courses and quizzes for developing malware analysis capabilities;
- developing courses in the field of IoT and specific technologies;
- designing courses at a global level, with a number of about 27,000 visitors and an average of over 5,000 students in the last 5 years (2017 - 2021). Among the courses supported I mention: *Threat Intelligence: The Big Picture*, *Getting Started Analyzing Malware Infections*, *Threat Hunting with Yara*, *Advanced Malware Analysis: Ransomware*.

5.2 Original contributions

Below is the full list of original contributions from this paper. These were obtained from simulations, experiments, and actual implementations either directly in organizational environment (for eligible contributions) or using virtual machines and software isolated test environments (for contributions with potentially malicious impact).

1. I proposed a solution to detect phishing websites **[A1]** based on the analysis of similarities between the original and the fake page. Based on a study of the specific attributes between the phishing page and the original one, I extracted a specific set of features that facilitates the successful detection of cloned pages.
2. I implemented a ransomware detection solution based on honeypot technologies **[A2]** where, leveraging a file sharing service, I created a program that continuously monitors the integrity of files and can generate alerts when their integrity is compromised. The solution can also identify "patient-0", i.e. the first system that initiated the infection.
3. I proposed a solution for dynamic malware analysis using neural networks **[A3]**, which is based on classifying processes into specific categories. Detection is performed based on the analysis of anomalies in the execution hierarchy for each process **[D1]**. Classifying processes into categories makes it possible to improve and modify the list of processes without the need to retrain the neural network.
4. I statically and dynamically analyzed a ransomware virus - Stampado **[B7]** - and extracted the indicators of compromise, but also their behavior in relation to the spread of the infection in a network of computers through file sharing services.
5. I analyzed and synthesized the application of AI techniques in the context of cyber-attack detection identifying three major categories: network intrusion detection, malware detection and spam/phishing.
6. I conducted a study on SSL libraries for IoT devices **[B5] [B6]**. And I have identified a vulnerability in the SSL libraries, caused by the lack of implementation of the CLR and OCSP protocols used to validate expired or compromised cryptographic certificates. Another identified vulnerability consisted in the lack of updating and maintaining the libraries for the studied hardware platform.
7. I conducted a study of system-level and network-level computer virus detection and analysis capabilities and techniques. I compared the approach to cyber-attack detection between two paradigms, the vulnerability-centric approach (which assumes that an organization will take action after a security incident has occurred) and the threat-centric defense, which assumes the proactive approach in which an organization will take action prevention before an incident occurs.
8. In the context of the measures that an organization must take upon the occurrence of an incident (physical or logical) that leads to a critical effect (non-operation / lack of provision of a service), I contributed to the development of a Fault Tree Analysis (FTA) **[A4]** to evaluate the causes that can fail a Web server taking into account hardware and software aspects.
9. I proposed a reference architecture based on open-source technologies for an isolated environment for simulating cyber-attacks, testing techniques and detection mechanisms. This environment was used to generate data for training detection algorithms based on AI techniques.
10. I conducted a study on the challenges of securing computer systems. Based on reports and surveys conducted by organizations in the field of cyber security, I have identified the important factors that underlie the challenges related to securing computing systems. Among them are the lack of analysts and experts in the field of

information security, as well as the increase in both the volume and the complexity of cyber-attacks.

11. I developed a study on the methodologies for detecting and analyzing complex cyber-attacks **[D2]** and proposed the use of new methodologies for approaching attacks: from the point of view of the Kill Chain, approaching attacks as a consecutive series of stages, and from the point of view of view of MITRE, the classification based on the techniques used, not only the type or family of computer viruses used.

12. I conducted an analysis of cyber-attacks that had a major impact **[B4]** both on the organizations that were victims and on ordinary citizens who use the services of these organizations **[D4]**. I analyzed their impact as well as the initial cause of security breaches based on reports published by experts in the field.

13. I defined and implemented a cyber-attack simulation plan in an organization, which aimed to improve the level of training of analysts within the cyber security department and their ability to detect and analyze modern cyber-attacks. As a relevance we proposed the simulation of incidents following a ransomware virus attack **[D3]**. Some exercises were based on assessing the ability to restore critical services such as Active Directory.

14. I contributed to a comparative study **[B8]** regarding organization within a security department. More specifically, we evaluated the roles of analysts **[B9]** within SOC (Security Operations Center) and how the migration to SIC (Security Intelligence Center) brings benefits in terms of mode and efficiency in combating cyber-attacks.

15. Based on the behavioral characteristics of computer viruses, I have created for educational purposes malware applications using technologies for simulating cyber-attacks for both Windows and Linux platforms. This facilitated the simulation of attacks to test detection and analysis capabilities.

16. I proposed methods for detecting attacks at the network level based on network statistics and processing of traffic in a computer network **[D1]**. Based on these methods, I have identified both volumetric DoS-type resource exhaustion attacks, data theft through DNS tunnels, as well as identification of phishing sites.

17. We performed a comparative analysis of malware detection methods at the system level: based on hash signatures, based on Yara rules **[D5]** and based on events recorded at the operating system level. I elaborated on methods to detect malware persistence on a system and concluded by detecting malware based on the correlation of events logged at the system level with corresponding events at the network level.

18. I proposed **[A2] [B1] [B2]** and contributed **[A5]** to cyber-attack detection solutions based on IoT platforms **[B3]** as tactical or complementary alternatives to strategic hardware or cloud solutions.

5.3 List of original publications

A. Scientific articles in ISI / IEEE Xplore indexed publications

[A1] C. Pascariu, I.C. Bacivarov, *Detecting Phishing Websites Through Domain and Content Analysis*, 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Pitești, România, 2021, eISBN: 978-1-6654-2534-6, DOI: 10.1109/ECAI52376.2021.9515165.

[A2] C. Pascariu, I.D. Barbu, *Ransomware Honeypot. Honeypot solution designed to detect a ransomware infection identify the ransomware family*, 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Pitești, România, 2019, eISBN: 978-1-7281-1624-2, DOI: 10.1109/ECAI46879.2019.9042158, WOS: 000569985400166.

[A3] C. Pascariu, I.D. Barbu, I.C. Bacivarov, *Dynamic analysis of malware using Artificial Neural Networks. Applying Machine Learning to identify malicious behavior based on parent process hierarchy*, 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Targoviste, România, 2017, eISBN: 978-1-5090-6458-8, DOI: 10.1109/ECAI.2017.8166505, WOS: 000425865900121.

[A4] G. Petrică, I.D. Barbu, S.D. Axinte, **C. Pascariu**, *Reliability analysis of a Web server by FTA method*, The 10th International Symposium on Advanced Topics in Electrical Engineering (ATEE), Bucharest, 2017, pp. 683-686, DOI: 10.1109/ATEE.2017.7905101, WOS: 000403399400133.

[A5] I.D. Barbu, **C. Pascariu**, I.C. Bacivarov, S.D. Axinte, M. Firoiu, *Intruder monitoring system for local networks using Python*, 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Targoviste, România, 2017, eISBN: 978-1-5090-6458-8, DOI: 10.1109/ECAI.2017.8166457, WOS: 000425865900073.

B. Scientific articles in IDB indexed publications

[B1] C. Pascariu, *Getting Started with Vulnerability Disclosure and Bug Bounty Programs*, International Journal of Information Security and Cybercrime (IJISC), Vol. 11, No. 1, 2022, pp. 25-30, ISSN: 2285-9225, DOI: 10.19107/IJISC.2022.01.03.

[B2] C. Pascariu, I.D. Barbu, I.C. Bacivarov, *Network security monitoring with embedded platforms*, Proc. of the 16th International Conference on Quality and Dependability Sinaia, Romania, September 26th-28th, 2018, pp. 243-246, ISSN: 1842-3566.

[B3] C. Pascariu, I.D. Barbu, *Using Embedded Platforms to Monitor Network Security*; International Journal of Information Security and Cybercrime (IJISC), Vol. 7, No. 2, 2018, pp. 9-13, ISSN: 2285-9225, DOI: 10.19107/IJISC.2018.02.01.

[B4] C. Pascariu, I.D. Barbu, I.C. Bacivarov, *Investigative Analysis and Technical Overview of Ransomware Based Attacks. Case Study: WannaCry*, International Journal of Information Security and Cybercrime (IJISC), Vol. 6, No. 1, 2017, pp. 57-62, ISSN: 2285-9225, DOI: 10.19107/IJISC.2017.01.06.

[B5] C. Pascariu, I.D. Barbu, I.C. Bacivarov, *WannaCry ransomware analysis. 1 day, 150 countries, >57k infected computers*, Asigurarea Calității - Quality Assurance, Vol. XXIII, Numărul 90, Aprilie-Iunie 2017, pag. 4-7, ISSN 1224-5410.
[B6] C. Pascariu, I.D. Barbu, I.C. Bacivarov, *Secure Smart Cities*, Proceedings of the 15th International Conference on Quality and Dependability Sinaia, Romania, September 14th-16th, 2016, pp. 161-166, ISSN 1842-3566.
[B7] C. Pascariu, I.D. Barbu, *Ransomware - an emerging threat*, International Journal of Information Security and Cybercrime (IJISC), Vol. 4, No. 2, 2015, pp. 27-32, ISSN: 2285-9225, DOI: 10.19107/IJISC.2015.02.03.
[B8] I.D. Barbu, **C. Pascariu**, I.C. Bacivarov, *Migration of a SOC to SIC. Security Operations Center vs. Security Intelligence Center. The use of honeypots for threat intelligence*, Proc. of the 15th International Conference on Quality and Dependability Sinaia, Romania, September 14th-16th, 2016, pp. 150-155, ISSN 1842-3566.
[B9] I.D. Barbu, **C. Pascariu**, *Information security analyst profile*, International Journal of Information Security and Cybercrime (IJISC), Vol. 3, No. 1, 2014, pp. 29-36, ISSN: 2285-9225, DOI: 10.19107/IJISC.2014.01.03.

C. Presentations in specialized conferences

[C1] C. Pascariu, *Crowdsourcing Information Security*, Digital 2021 ISF World Congress, eveniment online.
[C2] C. Pascariu, *Adversary Emulation: Building your Purple Team*, ISF Congress 2020, eveniment online.
[C3] C. Pascariu, *Cyber Deception: Hunting for Ransomware*, ISF Congress 2019, Dublin, Irlanda.

D. Scientific reports within the doctoral program

[D1] C. Pascariu, *Prelucrarea statistică a evenimentelor de securitate*, Scientific report 1, June 2016.
[D2] C. Pascariu, *Detectarea atacurilor cibernetice la nivel de rețea*, Scientific report 2, December 2016.
[D3] C. Pascariu, *Analiza virușilor de tip ransomware*, Scientific report 3, June 2017.
[D4] C. Pascariu, *Analiza riscurilor și securizarea orașelor inteligente (Smart Cities)*, Scientific report 4, December 2017.
[D5] C. Pascariu, *Analiza și clasificarea virușilor cibernetici folosind reguli Yara*, Scientific report 5, June 2018.

Bibliography

[1] V.M. Cătuneanu, I.C. Bacivarov, *Fiabilitatea sistemelor de telecomunicații*, Ed. Militară, București, 1985.

[3] Gartner Glossary, Digitalization, https://www.gartner.com/en/information-technology/glossary/digitalization#:~:text=Digitalization%20is%20the%20use%20of,roadmap%20for%20digital%20business%20transformation.

[4] ISO, ISO/IEC 27000:2018 Information technology - Security techniques - Information security management systems - Overview and vocabulary, https://www.iso.org/standard/73906.html.

[5] ISO, ISO/IEC 27000:2018(en), https://www.iso.org/obp/ui/#iso:std:iso-iec:27000:ed-5:v1:en.

[16] ENISA Thread Landscape report - 2021, https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends.

[34] C. Pascariu, *Analiza riscurilor și securizarea orașelor inteligente (Smart Cities)*, Raport științific nr. 4, decembrie 2017.

[44] RFC 5280, Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, https://www.ietf.org/rfc/rfc5280.txt.

[45] R. Sanders, *What is a Certificate Revocation List (CRL) vs OCSP?*, https://www.keyfactor.com/blog/what-is-a-certificate-revocation-list-crl-vs-ocsp/.

[46] Adafruit WICED WiFi Feather - STM32F205 with Cypress WICED WiFi - Discontinued, https://www.adafruit.com/product/3056.

[47] Github, Adafruit WICED Feather Arduino BSP, https://github.com/adafruit/Adafruit_WICED_Arduino.

[48] C. Pascariu, I.D. Barbu, I.C. Bacivarov, *WannaCry ransomware analysis. 1 day, 150 countries, >57k infected computers*, Asigurarea Calității - Quality Assurance, Vol. XXIII, Numărul 90, Aprilie-Iunie 2017, pag. 4-7, ISSN 1224-5410.

[52] C. Pascariu, I.D. Barbu, I.C. Bacivarov, *Network security monitoring with embedded platforms*, Proceedings of the 16th International Conference on Quality and Dependability Sinaia, Romania, September 26th-28th, 2018, pp. 243-246, ISSN 1842-3566.

[53] I.D. Barbu, C. Pascariu, I.C. Bacivarov, S.D. Axinte, M. Firoiu, *Intruder monitoring system for local networks using Python*, 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Targoviste, România, 2017, eISBN: 978-1-5090-6458-8, DOI: 10.1109/ECAI.2017.8166457, WOS: 000425865900073.

[54] C. Pascariu, I.D. Barbu, *Using Embedded Platforms to Monitor Network Security*, International Journal of Information Security and Cybercrime (IJISC), Vol. 7, No. 2, 2018, pp. 9-13, ISSN: 2285-9225, DOI: 10.19107/IJISC.2018.02.01.

[55] Python, https://www.python.org.

[56] Scapy Project, https://scapy.net.

[57] V. Ramachandran, S. Nandi, *Detecting ARP Spoofing: An Active Technique*, Information systems security: first international conference, ICISS 2005, Kolkata, India, December 19-21, 2005, p. 239, ISBN 978-3-540-30706-8.

[72] C. Pascariu, *Getting Started with Vulnerability Disclosure and Bug Bounty Programs*, International Journal of Information Security and Cybercrime (IJISC), Vol. 11, No. 1, 2022, pp. 25-30, ISSN: 2285-9225, DOI: 10.19107/IJISC.2022.01.03.

[83] L. Zeltser, *How to get and set up a free Windows VM for Malware Analysis*, https://zeltser.com/free-malware-analysis-windows-vm/.

[84] VMware Workstation PRO, desktop hypervisor, https://www.vmware.com/nl/products/workstation-pro.html.

[86] Kali Linux | Penetration Testing and Ethical Hacking Linux Distribution, https://www.kali.org.

[87] Sysmon | System Monitor, https://docs.microsoft.com/en-us/sysinternals/downloads/sysmon.

[90] Osquery | Performant endpoint visibility, https://osquery.io.

[91] Security Onion 2 | Security monitoring and log management, https://securityonionsolutions.com/software.

[92] Zeek (Bro), https://zeek.org.

[93] Elastic Stack, https://www.elastic.co/elastic-stack/.

[94] Elasticsearch, https://www.elastic.co/elasticsearch/.

[95] Kibana, https://www.elastic.co/kibana/.

[97] Beats, https://www.elastic.co/beats/.

[123] C. Pascariu, I.C. Bacivarov, *Detecting Phishing Websites Through Domain and Content Analysis*, 13th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Pitesti, Romania, 2021, eISBN: 978-1-6654-2534-6, DOI: 10.1109/ECAI52376.2021.9515165.

[125] Kaspersky, What is Typosquatting? - Definition and Explanation, https://www.kaspersky.com/resource-center/definitions/what-is-typosquatting.

[126] D. O'Regan, *How scammers use sub-domains*, https://easykey.uk/computer-safety/how-scammers-use-sub-domains.

[130] N. Jaiswal, SequenceMatcher in Python, 2019, https://towardsdatascience.com/sequence matcher-in-python-6b1e6f3915fc.

[131] S. Coble, Most Phishing Pages are Short-lived, Infosecurity magazine, 2021, https:// www.infosecurity-magazine.com/news/most-phishing-pages-are-shortlived/.

[132] DNStwist phishing domain scanner, https://dnstwist.it.

[133] The Social-Engineer Toolkit (SET), https://www.trustedsec.com/tools/the-social-engi neer-toolkit-set/.

[139] Microsoft, Using and Configuring Autoplay, https://docs.microsoft.com/en-us/previous-versions/windows/desktop/legacy/cc144212(v=vs.85).

[140] Shell Link | LNK binary file format, https://docs.microsoft.com/en-us/openspecs/ windows_protocols/ms-shllink/16cb4ca1-9339-4d0c-a68d-bf1d6cc0f943.

[144] C. Pascariu, I.D. Barbu, *Ransomware Honeypot. Honeypot solution designed to detect a ransomware infection identify the ransomware family*, 11th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Pitesti, Romania, 2019, eISBN: 978-1-7281-1624-2, DOI: 10.1109/ECAI46879.2019.9042158, WOS: 000569985400166.

[147] SMB | Samba sharing service, https://www.samba.org.

[148] Samba VFS Full Audit Config, https://www.samba.org/samba/docs/current/man-html/ vfs_full_audit.8.html.

[150] Matplotlib: Visualization with Python, https://matplotlib.org.

[175] G. Apruzzese, M. Colajanni, *On the effectiveness of Machine and Deep Learning for Cyber Security*, NATO CCD COE Publications, Tallinn, 2018.

[176] L.A. Zadeh, *Fuzzy logic, neural networks, and soft computing*, Commun. ACM 37, 3, 1994, 77-84, DOI: 10.1145/175247.175255.

[179] C. Pascariu, I.D. Barbu, I.C. Bacivarov, *Dynamic analysis of malware using Artificial Neural Networks. Applying Machine Learning to identify malicious behavior based on parent process hierarchy*, 9th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Targoviste, Romania, 2017, eISBN 978-1-5090-6458-8, DOI: 10.1109/ECAI.2017.8166505, WOS: 000425865900121.

[185] D.E. Rumelhard, *Backpropagation: The Basic Theory*, Backpropagation: Theory, Architectures, and Applications, SUA: Library of Congress, 1995, pp. 1-35.