UNIVERSITÀ
DI PARMA

Dipartimento di Scienze Matematiche, Fisiche e Informatiche

Corso di Laurea Triennale in Informatica

Cybersecurity, malware e social engineering: rilevamento e prevenzione

Cybersecurity, malware and social engineering: detection and prevention

Candidato:
Tommaso Pellegrini

Relatore:
Prof. Roberto Alfieri

Anno Accademico 2021–2022

To my parents, my friends, my partner.

Cybersecurity, malware and social engineering: detection and prevention

Abstract

This text discusses information security, malware and social engineering. In particular, their attack vectors, how to detect, prevent, and remediate them are shown. From their history, evolution, and spread to their role in today's world, it is intended to show precisely how the use of social engineering and malware are strongly intertwined with each other and the impact they have had on society and global organizations, both in the past and in the present day. Special attention is also given to how the symbiosis between malware and social engineering is arrived at. After demonstrating multiple examples of attacks of different types and the dangers derived from them, a solution to them is provided, as well as multiple preventive measures and best practices to be used to best protect one's network, be it that of a private individual or a corporate network. In addition to commonly used defensive measures such as firewalls and antivirus, special attention is paid to a certain type of tool, namely that of SIEMs (security information and event management). These security systems are discussed in depth, so as to provide enough information to understand what they are, how they operate, and what their capabilities are. The text then concludes with a well-detailed example of how to design, install, configure and maintain a system such as the one mentioned above, as part of the internship held at the University of Parma University in conjunction between the author and the University's security team.

Introduction

When people talk about hackers and cyber attacks, they often think of individuals or organizations that by exploiting computer flaws manage to take control of computers, cell phones, lock down databases, access secret information, and similar threats. There is a common tendency to ignore the true workings of a hacker, covering it with a veil of mystery often fueled by news and media outlets. In most cases, however, it is not just a matter of system vulnerabilities, misconfigurations, or unmodified default passwords. Rather, it is about exploiting the weak link in the security chain to get what you want: the human being.

Cybersecurity is a topic that has been gaining increasing attention in recent years from both companies and individuals. The reason is trivial: the world is becoming computerized at a tremendous speed, and there is no denying that personal information, even that of the individual such as interests and habits is growing in value by leaps and bounds, not to mention the importance of private information of any company or organization. That information that, if it falls into the wrong hands, is enough for a hacker to take total control of a computer or even an entire system. "A company may have purchased the best security technology money can buy, trained every single employee so well that the latter hides his or her data perfectly before going home at night, and even hired the best security guards available. This company is still vulnerable. Employees could follow the best security

practices recommended by experts, install every product designed to protect data and passwords, and be careful to always install the latest update or patch that is. These individuals are still vulnerable, however." According to statistics published by Purplesec, cyber attacks have increased tremendously over the past decade. From 2010 with 12.4 million attacks per year, to 2014 with 308.96 million to 2018 with as many as 812.67 million. Of these attacks, 98% make use of social engineering (or social engineering), albeit in a small way. This text explains what malware and social engineering are, how they work, which techniques are most widespread and used; discusses their history and evolution, concrete examples, and ends with an analysis of possible methods to secure information.

Chapter 1

Malware, their history and evolution

A malware is "software that, when executed, damages the operation and security of the operating system"; the term comes from the contraction of malicious and software and literally means "evil program." There are several types of malware, including the best known and often misused type to define them all, viruses. Although the definition tells us that the operation and security of an infected machine is being damaged, malware can have different purposes. Some are created as a joke, others have compromised systems of enormous size and importance. The reality is that every malware has a well-defined goal, normally thought of by the creator of the malware itself, and this can vary depending on the author's purpose.

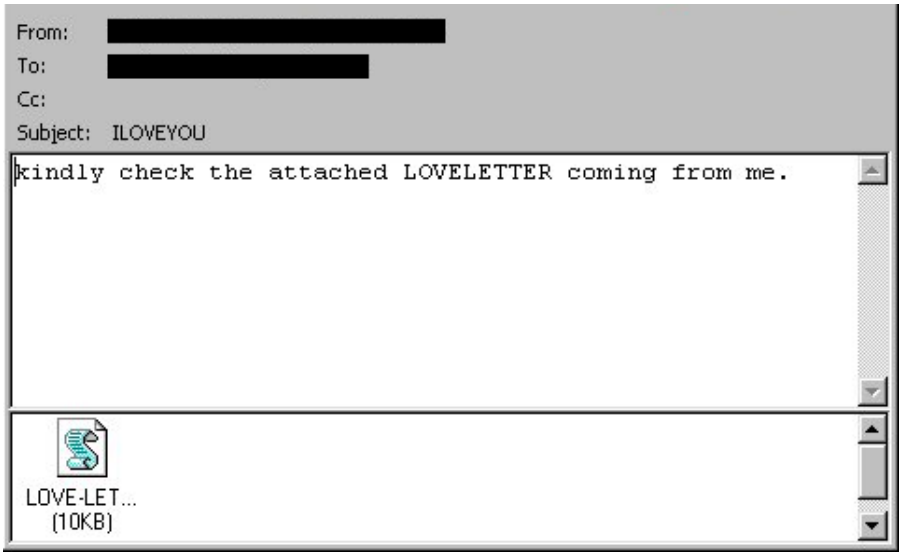
1.1 The first malware

The spread of malware via the Internet (at the time ARPANET) dates back to 1971, when Bob Thomas of BBN developed the malware "The Creeper Program." Given the malware's ability to create copies of itself, it will be considered a worm type. Although the program in question had no real malicious intentions and was merely a test to see if a program that replicated itself countless times automatically was possible, it is commonly regarded as the first malware created. In the following years several pieces of malware are created following the footprints of their predecessor such as "The Rabbit Virus" or Wabbit, created in 1974 by an unknown source, also a worm category malware that replicates hundreds of times on the infected machine slowing down its performance considerably . Then in 1975 the first Trojan, called "ANIMAL," was created by John Walker. The latter was nothing more than one of many computer games popular at the time where the program tried to guess which animal the user was thinking of. During the installation of ANIMAL, however, a second program called PREVADE was installed, which, while running its counterpart, examined all the directories on the computer and installed a new copy of ANIMAL in each of them. Hidden inside ANIMAL is another program that performs actions not authorized by the user. A few years later in 1986, two brothers named Basit and Amjad Farooq Alvi, owners of a computer store in Pakistan, created the first virus recognized as such: "Brain." The virus targeted 5.2'' floppy disks and replaced their boot partition with a virus that

although not malicious, contained a copyright message. Three years later, in 1989, the first ransomware came to light, a type of malware that was destined to become one of the most widely used by cyber criminals to get companies into trouble and also one of the most dangerous. Created by Dr. Joseph Popp, the trojan named "AIDS" was sent via physical mail to several researchers around the world of the then highly relevant disease of the same name using over twenty thousand infected floppy disks. The Trojan initially contained a questionnaire about AIDS, but after the nineteenth reboot the system filenames were encrypted and hidden from the user, who was charged a fee of $189 for an annual license or $385 for a perpetual license. Moving forward a few years, we arrive in the early 1990s, where for the first time a computer malware gained such fame that it was talked about both in newspapers and on various television channels. It was in 1991 that the Michelangelo virus was developed from an anonymous source. This virus operates at the BIOS level and infects DOS systems but does not make any system calls or interact with the operating system. Although the virus was almost harmless, it gave rise to a kind of mass hysteria being programmed to activate on March 6 of that year, with consequences unknown at the time, the date of the famous painter's birthday. John McAfee estimated that the virus had infected over five million computers and the media brought the story to the front page numerous times, raising awareness for the first time among the masses about malware and its possible consequences. In reality, the infected computers numbered just under a few thousand, but Michelangelo remained the first malware to make headlines worldwide when there were still few personal computer users. By the mid-1990s, Internet users had definitely increased compared to previous decades, and this is where we see the first phishing-type attacks, when in '94 or '95 a group of hackers decided to impersonate AOL staff and proceeded to steal several user accounts through bogus emails and messages on AOL's instant messaging service. We will then see how in subsequent years phishing-type attacks, like other various malware, have evolved and changed their methods of creation and especially dissemination to stay with the times.

1.2 The dissemination

The method of spreading early malware was confined to the use of the then ARPANET, as in the case of The Creeper Program or floppy disks for ANIMAL and others. There were no methods as effective as distributing infected disks or exploiting networks of private companies, and at that time the use of social engineering was still scarce, except in the late 1990s where, as we saw in the previous chapter, Microsoft Words files infected often with Trojans and Worms and the first phishing attacks began to show up sporadically. A big step in favor of the spread of malware occurred in March 2000, when LoveLetter first appeared. Created by Onel de Guzman, the worm showed up in the inbox not as an infected Word document, but as a VBS file, a scripting language whose code could run directly from Windows as OS or from Internet Explorer. Once opened, the malware was designed to find every single file on the system and overwrite it with a copy of itself, a simple text file containing the phrase "I LOVE YOU."



An email containing the I LOVE YOU malware. Source: Kaspersky

Copies of the file were then automatically sent to all email contacts of the infected user. Despite being simple and straightforward malware, still unwary users were not used to having to be wary of the 'unexpected email attachment. This mistake is repeated very often even today, and this is how in a large number of cases hackers manage to gain access to data, computers or systems. Statistics published by the Kaspersky Virus Lab show that in 2001 cyber attacks consisting of malware attached to emails not only increased by 5 percent from the previous year, but also accounted for nearly 90 percent of malware incidents in that year. Opening a malicious file as an email attachment is the perfect example of phishing (more on this later), which, in addition to being one of the most classic examples of social engineering, is still widely used along with its counterpart via SMS or push notification. In fact, it is in the early years of the new millennium that we see how the spread of malware had also expanded to mobile devices, using new technologies that were still young and had several security problems. In 2004 Cabir, the first worm-type malware capable of infecting cell phones based on Symbian OS, used mainly by Nokia, came to light. The worm was able to reproduce itself by sending a copy of itself using cell phones' built-in Bluetooth or via MMS. Unlike many other early malware, Cabir was created as a "proof of concept," and instead of distributing it among individuals it was immediately sent to several cybersecurity firms so that they could find a solution to the problem. It is evident how as new methods of connecting devices to each other arise, from ArpaNet, to Bluetooth, to the latest smart home devices, new methods of distributing malware also arise.

1.3 The evolution

The year 2001 is not only the year that saw the rise to success of phishing, but also the year when other new forms of malware distribution were born. Prior to that time, the vast majority of malware disseminated in those years, often worms and Trojans, exploited serious vulnerabilities in programs widely used in the business environment such as Microsoft Word, Outlook and others and were distributed via email, but to be infected one had to make a major, often avoidable mistake: downloading the malware. By that time,

however, there was no need to download some strange file sent by a supposed work colleague. Just visiting an infected website was enough to get infected accordingly. Web pages were replaced with malicious counterparts containing code that, by exploiting flaws in MS Internet Explorer, could be executed directly by the client. Another step for social engineering occurred at the same time, when several websites began providing free versions of paid programs that were, however, accompanied by malware. Again, we see how exploiting the naiveté and flaws of the human personality brings a great deal of advantage. Offering something expensive for free turns out to be a hard offer to refuse. Taken by avarice and curiosity, there are millions of users who infect their machines every year.

Although the early 2000s saw a great evolution of antivirus systems, the amount of malware created and shared continued to increase, demonstrating how the explosion of the Internet and its increasingly easy access around the world brought with it its own risks and dangers to its still uneducated users. A final tool that saw enormous exploitation by hackers aimed at spreading malware was instant messaging, or "IM." From IRCs that took their first steps in 1988 to AOL's AIM, Microsoft's MSN, instant messaging services gained millions of users until they surpassed 300 million users in 2005. Given such popularity and widespread use by businesses as well (think Skype for example), IM platforms became targets and sources of numerous cyber attacks. In the first quarter of 2004 alone, the number of attacks aimed at major IM platforms saw a 400% increase. In place of infected word files, however, we also see a huge increase in the use of social engineering, as the vast majority of these attacks are perpetrated with the same modus operandi used in phishing. The numbers we have seen so far are bound to increase, as are the victims of phishing and social engineering in general.

1.4 The dawn of cyber crime

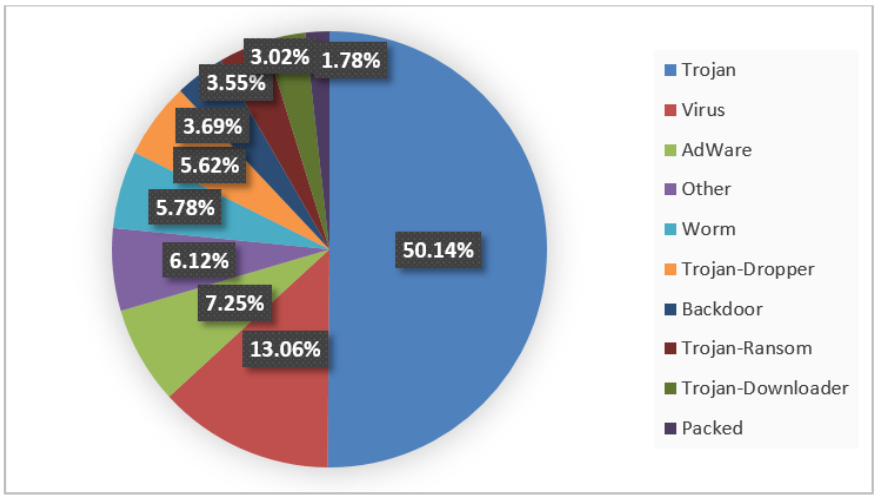
We have seen how very little time passes between the creation of a new technology and its possible use for malicious or fraudulent purposes. However, early malware and malicious actors should not be confused as organized crime activities. Although several pieces of malware were born with the purpose of illicitly making money, they remain mostly operations carried out by individuals or small groups of individuals. But as with anything, when certain novelties are brought to the attention of many, malware and the Internet attracted several criminal groups who saw a new potential source of income at significantly lower risk than criminal activities carried out in person. Studies show how several organized crime groups exploited these new technologies to insert themselves into online gambling and scamming rings. For example, there is evidence of how in 2016 several members belonging to the Camorra and 'Ndrangheta were arrested because of an illegal online betting ring. There are also several instances where criminal groups have started cyber crimes in order to facilitate their offline crimes, such as a group of drug traffickers who in early 2010 hired a group of hackers to gain access to the cyber infrastructure of the port of Antwerp in Belgium, with the aim of obtaining information related to containers in order to ship drug shipments in an unsuspicious way. Thus, it is easy to imagine how several criminal groups can arise just to operate online or transfer their activities completely or partially online. Although several of these groups were formed online, research related to the creation and development of organized cyber-crime networks

shows that geographic areas and offline contacts play an extremely important role in their formation and expansion. Indeed, several hotspots and meeting centers for online organized crime groups have been found in Eastern Europe, and it has also been revealed by Europol how the majority of scams using social engineering aimed at European countries are mainly carried out by criminal groups residing in areas of West Africa.

Another tool that has fostered the creation of such groups is definitely the so-called Dark Web, in strong conjunction with the birth of the first cryptocurrency, BitCoin. The Dark Web is that part of the internet whose content is not indexed by the search engines we are used to, i.e. Google, Bing, Yahoo etc. It is a collective of websites accessible only by using specialized web browsers, including the most famous Tor. The dark web was not born with criminal intent, however, as its origin was to create an encrypted and completely anonymous network with which U.S. spies could communicate with each other while protecting the psychic content that actors exchanged. However, when Tor, a private browsing browser, was released to the world in 2002, everyone was able to take advantage of the anonymization services we have discussed. While some users use it to evade government censorship, it has become well known for being used to engage in highly illegal activities. From buying and selling huge databases stolen from private and government companies, to selling and shipping large quantities of drugs and weapons online to using it to hire hackers or even groups of assassins, the Dark Web becomes to this day one of the most widely used networks for the accomplishment of illegal activities, both by individuals and large organized crime groups.

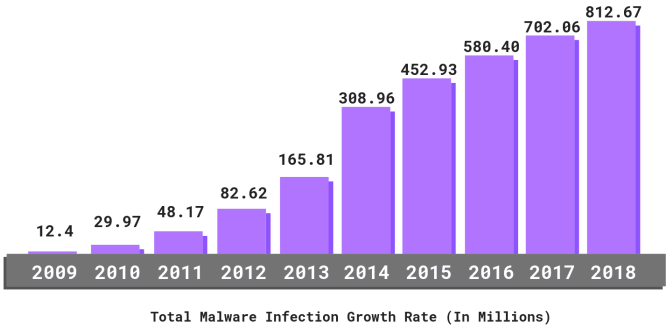
1.5 Today

Let's now look at some statistics: 92% of malware is delivered via email. Mobile malware and variants are on the rise, with 54% increase in 2018 alone. Malware for macOS has increased by 165% in the last year. Even now, Trojans account for about 50% of malware on the network.



Statistics on the type of malware. Source: Kaspersky

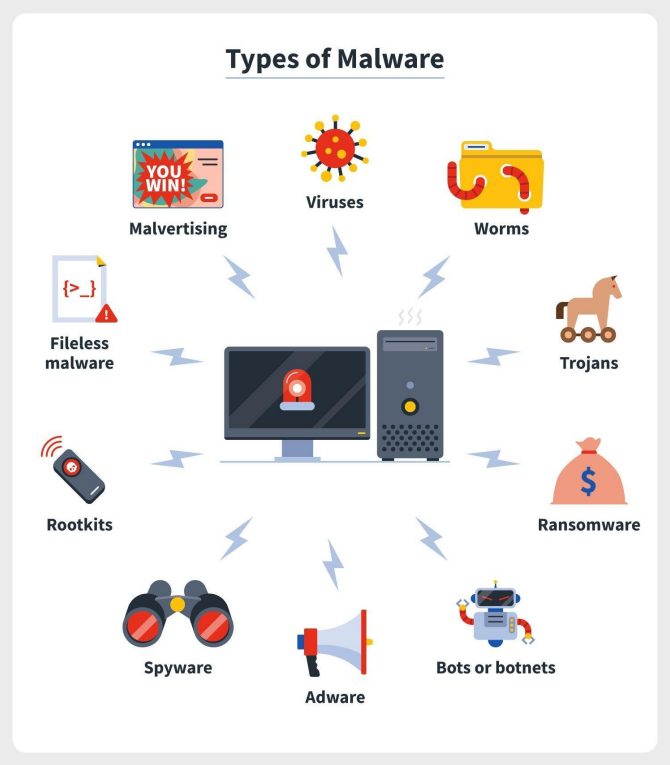
More than 18 million websites are infected with malware every week. 90% of financial institutions reported at least one cyber attack in 2018. As mentioned earlier, 95% of cyber attacks make use of social engineering. We will return to this important data later. A necessary mention must also go to a type of malware that has had a frightening 358% increase in recent years in 2018, ransomware: capable of encrypting user data and preventing access by the user by threatening its destruction after a countdown unless a ransom is paid, often untraceable because it is paid via cryptocurrency. The numbers don't lie, cyber threats have increased disproportionately in the last 20 years and are set to increase. For this reason, it is important to educate users, from individuals to government employees, about the possibility of being a victim of these attacks in hopes of bringing users to a level of knowledge where they can avoid crisis situations.



Growth over the years of malware. Source: Purplesec

1.6 The types of malware

The following is a list that specifically explains the types of malware most widely known and used to date. We will deal later with attack types that cannot be attributed directly to malware, but how much more to attack vectors (DDoS, SQL injections etc.), password attacks (such as brute forcing), and Man in the Middle (or MitM), which are very important in social engineering, but do not belong to the malware category. It is intended to define a clear dividing line between a malware and an attack on an IT infrastructure. The following list does not contain every single type of malware that exists, but it is sufficiently substantial to give a general idea of the largest categories present.



Different types of malware. Source: Norton

- Virus: is a type of malicious program that once executed begins to replicate itself by modifying other programs inside the infected machine. As a rule, viruses need a host program, a base program from which to start the infection, and then spread to others. Viruses can then be part of other categories of malware, effectively becoming keyloggers, Trojans or ransomware as well.
- Trojan: a Trojan Horse is a type of malware that mirrors the horse behavior used by the Greeks: pretending to be a legitimate program, through social engineering users are tricked into running it. The payload of the trojan can be any, often a backdoor, and they are usually used to steal information. Unlike worms and viruses, Trojans do not inject malicious code into files or attempt to replicate themselves.
- Worm: more similar to viruses, worms try to infect other computers by duplicating themselves while also remaining active on previously infected machines. Worms primarily use the network to spread, exploiting vulnerabilities to then get to their next victim. Like viruses, worms can also take subforms: they can create botnets, keyloggers, and more.
- Ransomware: are a form of malware designed to encrypt and deny access to the infected machine's data until a ransom is paid. Often associated with a countdown, ransomware victims are often asked to make payment in the form of cryptocurrencies, which are extremely difficult to trace.

- Rootkit: among the most dangerous, are a set of malware designed to take total control of the system by masquerading as legitimate programs already existing on the machine. They can be installed manually post bruteforce, by phishing or by direct attacks on vulnerabilities. Particularly difficult to detect and nearly impossible to remove, some rootkits can install themselves in the kernel or firmware, requiring the replacement of hardware parts of the machine in order to remove them.

- Keylogger: simple malware that keeps track of every keystroke on the infected machine and then sends the logs containing the data to the hacker. Also often transmitted via social engineering they are particularly dangerous for the possible theft of data, passwords, banking info etcetera.

- Greyware: a term coined in September 2004, grayware describes a type of malware that has no particular malicious intent but worsens the performance of the infected machine. They are a larger category of malware that alludes to adware and spyware, for example.

- Fileless malware: as guessed from the name, are types of malware that use legitimate programs to infect a machine, but leave no traces on the system and do not write to the hard drive. Because they are loaded only on volatile memory such as RAM, they are difficult to detect and die when the machine is rebooted. They can be particularly tedious because once they vanish they leave little or no trace useful to forensic investigators.

- Adware: are a type of grayware created to produce unwanted advertisements often on browsers, desktops or popups. While not dangerous, they turn out to be particularly annoying and possibly complicated to remove if installed among system registries. Adware turn out to be the least dangerous but most profitable type of malware for hackers, being able to generate income simply by showing ads. They are often loaded by other programs during the installation phase if care is not taken. This also turns out to be a form of social engineering.

- Spyware: are malware that collect information about a person or organization without their knowledge and send the results to the hacker without permission. Often the goal is to be able to sell the collected data to third parties, but also to steal banking or personal data. As a rule, since they are not installed in the firmware or kernel, they are easy to remove once discovered.

- Backdoors: is a covert method of bypassing authentication in a computer, product, or embedded device (routers etc.). Backdoors are commonly used to establish remote access to a computer or access encrypted files and from there can be used to access, corrupt, delete, or transfer psychic data. They can be part of a program or installed directly into firmware or operating systems.

- Cryptojacking: Is a type of malware belonging to grayware. Similar to spyware or adware, it exploits the computational power of the machine that is infected to do cryptocurrency mining without the user's knowledge. The performance of the

machine suffers significant slowdowns and the longevity of hardware components can decrease very quickly.

Chapter 2

Social Engineering

The term "social engineering" refers to psychological manipulation through trickery or deception designed to get people to perform particular actions or reveal confidential information. The purpose is often to collect data or credentials, which are then used to gain access to a computer or an entire system for the purpose of fraud. It is important to note, however, that the meaning of the term does not have to be a negative one, as social engineering is widely used not only by cybercriminals but also by law enforcement, private investigators, find people, and journalists. We will see later in the text how social engineering is a necessary practice, for its good and its bad, used extensively by agencies that have always hunted cybercrime and beyond, from the Italian Secret Service to the FBI and Interpol.

2.1 Introduction to social engineering

To date, the techniques provided by social engineering are the most commonly used to commit cybercrimes through the intrusion and subsequent infection of computers and IT infrastructure. Because social engineering attacks are composed of a combination of social interactions and technology exploits, cybersecurity experts from private companies and the government struggle to implement effective countermeasures. The human factor remains the most difficult vulnerability to fix, since a simple patch is not enough as in a software system. In the vast majority of cases, the hacker or attacker never comes face to face with the victim, although this situation remains possible when active attack techniques are perpetrated, which we will discuss later. It is also important to note that victims of social engineering are often not properly prepared to deal with such threats, and this is often the cause of their success. Common targets such as corporate employees could be attacked by a hacker without even realizing it, becoming aware of it now that the damage has been done. These attack techniques have a wide variety of application being independent of the type of operating system, software or platform used. Whether it is a Windows, macOS or Linux operating system, the most vulnerable party remains as mentioned earlier the user, which is why it automatically becomes the target of greatest importance. In such cases, both antivirus and systems updated to the latest version are unable to provide adequate protection. In information security, as in IT security, every single area is exposed to risk when it comes to social engineering. As much as there is a lot of talk these days about artificial intelligence and the risks it poses, a hypothetical system protected solely by an AI, where even the highest level admin access is by prior authorization of the AI, would be the safest system from attacks related to Social Engineering. An AI, if programmed in such a way, would not have all those behavioral vulnerabilities common to human beings. This,

and much more, is the art of social engineering. The art of subjugation, of exploiting trivial details and nuances to one's advantage.

Tactics commonly used in social engineering. Source: Norton

2.2 Historical background

The history of the practice of social engineering can be interpreted in two different ways. Although technology and computers have evolved to generate the InfoSec-based social engineering concept only in recent decades, people have been using the principles of human psychology to manipulate others for hundreds of years. Social engineering is a practice as old as the dawn of humanity. For as long as coveted information has existed, so have people who want to exploit it. Just think of Greek mythology where Odysseus, after a ten-year-long war against the Trojans, changed tactics and using the famous Trojan horse (from which malware originates its name) managed to conquer the city from within, ending the war.

The term "social engineering" was first coined by Dutch industrialist J.C. Van Marken in 1984. Van Marken suggested that specialists in "human problems" were needed as much as there was a need for engineers to deal with all those technical problems. In 1911, even before Van Marken coined the term, Edward L. Earp used the term "Social Engineer" as a term to encourage people to handle social relationships similarly to the way one approaches a machine. It is then only since modern times that social engineering becomes a reference (and earns its official definition) to the process of manipulating persons to obtain information, usually followed by a cyber attack. Prior to that time, social engineering used many of the same attack vectors that are used to this day, but with the lack of an evolution (and many new methods) that would come shortly thereafter with the introduction of telephones into the homes of the ordinary citizen, email and then text messaging and so on. Because social engineering attacks are often conducted on complicated and interconnected

devices, it is difficult to draw a definite evolutionary line from the 1990s to the early 2000s. In addition, many of these attacks are conducted stealthily and are difficult to disclose, even after being carried out. This results in a lack of awareness of the attack by the victim and/or company, who may discover it in the future as well as remain totally unaware of it for years.

2.3 Famous social engineers

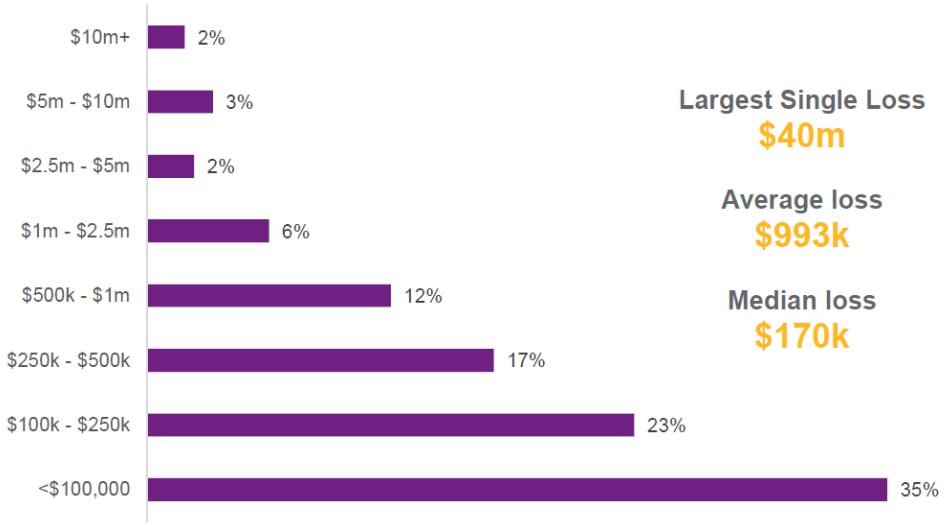
As complicated as it is to keep track of the various cases of social engineering, we are aware of what are considered the first cases of social engineering in the computer and Internet age. In 1982, a 15-year-old young man created what can be recorded as one of the first cases of social engineering for the purpose of spreading a virus. "Elk Cloner" was a worm-type virus created as a joke, but the fact that it spread via floppy disk under the guise of being a video game classifies it as one of the earliest cases of social engineering ever recorded. Although this young man is still unnamed today, we can list those who are instead considered the first social engineers of the modern era: Frank Abagnale Jr. is a cybersecurity consultant known for his past as a check forger, impostor, and fraudster. Impersonating various identities including pilot, doctor, doctor, lawyer and many others, he is famous for managing to escape from police custody twice, all before he turned 22. After years of escape and eventual arrest, Abagnale began working for the FBI as an undercover agent.

Susan Headly was a "phreak" or phone hacker, active between the 1970s and 1980s. She is known for her great social engineering skills, which enabled her to breach several military computer systems and even get past several checkpoints at the U.S. military base Area 51. She has also claimed in the past to have been able to obtain relevant information about the working schedules of intercontinental ballistic missile launch sites and possibly other related information of great importance. In more recent times, James Linton, is a British hacker and social engineer who in 2017 was able to use OSINT (open source intelligence) and several phishing attacks to scam several very important targets, including several CEOs of very important national banks, as well as several members of the Trump administration.

Combining past and present, we can then talk about what is considered the greatest social engineer ever: Kevin Mitnick. Born in 1963 in California, Mitnick is a cybersecurity consultant, author, and hacker. In the mid-1990s, however, he was the most wanted hacker in the world. At the young age of 12, Mitnick convinced a bus driver to let him tell him where he could buy a drill press for a "school project." After finding an unused transfer slip in a landfill near a bus company's garages, he was able to travel for a long time for free throughout Los Angeles. At the age of 16, Mitnick managed to penetrate the internal network of the Digital Equipment Corporation and copied its software, a crime for which he was arrested and sentenced to twelve months in prison. Toward the end of his supervised release, Mitnick managed to hack into the systems of Pacific Bell, a subsidiary of AT&T. After an arrest warrant was issued, Mitnick began a two-and-a-half-year escape. During his time on the run Mitnick managed to break into the computer systems of more than forty extremely important companies, not for financial gain but solely for defiance.

2.4 Statistics, social engineering in today's world

As we have said, social engineering relies less on hacking computers and more on manipulating people. Despite this, social engineering plays a key role in the success of a hacker attack. Nowadays, 98 percent of hacker attacks use some kind of social engineering technique. Once trust is gained, other attacks then follow. Whether it is identity theft, credential theft, or malware distribution, it is social engineering that serves as the bridge. On average, during a business year, a given organization is normally targeted by social engineering attacks more than 700 times. Considering that there are normally 260 working days in a year, this means that 2.7 attacks occur per day. On the database breach front, social engineering is the most commonly used approach to gain access to targeted systems. It is often easier to trick an employee of a company into giving him or her login credentials than to use a brute force attack. The result is that between 70 percent and 90 percent of database breaches occur through social engineering. Going back to talking about phishing, a favorite social engineering technique, this technique is used 25% of the time when a database breach occurs via social engineering. We also know that in 2021 approximately 83% of organizations in the United States were victims of at least one successful phishing attack, via email; this is a 43% increase over 2020, when Google removed as many as 2.1 million fake web pages suspected of phishing from its search results. Unfortunately, despite the great importance of the topic, only 27 percent of companies provide employees with training on how to recognize certain types of phishing and how to protect sensitive data.



Losses, in dollars, of several financial institutions in 2020 due to social engineering. Source: Willis Towers Watson Claims database.

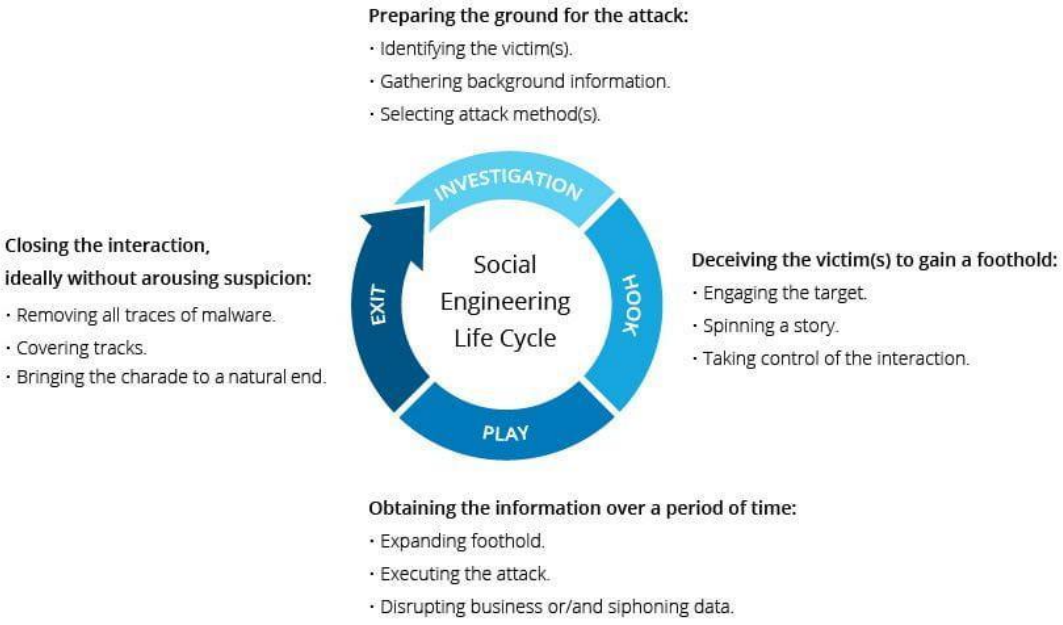
2.5 Terminology and concepts

All social engineering techniques rely on specific attributes of human decision making, known as cognitive biases. These biases are sometimes called "human hardware bugs" and are exploited in various combinations to create attack techniques. Typically, the process always relies on the hacker gaining the target's trust or exploiting their ignorance and then

using it to gain access to the sensitive information they were after. Social engineering relies heavily on the six principles of influence established by Robert Cialdini. Cialdini's theory of influence explains key principles as authority, intimidation, consensus, scarcity, urgency, and familiarity. The authority principle tells us that the attacker could pose as an authority figure (police, department head etc.) to increase the attacker's chances of success. The intimidation principle is based on instilling fears of bad consequences if certain actions, intended by the malicious actor, are not completed. The consent principle explains how people generally tend to perform certain actions if they see other people doing the same thing. For example, a passerby on the street will look up to the sky if he sees dozens of other people doing the same. The scarcity principle tells us in a simple way that a perceived scarcity generates demand. The famous phrase "while supplies last" capitalizes on the sense of scarcity. The urgency principle, similar to the scarcity principle, is used by the malevolent actor as a psychological attack on the victim, aiming to confuse him or her by generating anxiety. Finally, the familiarity principle explains how people are more easily persuaded by those who like them. In this case, the stereotype of good physical appearance can also be very important.

2.6 The life cycle of social engineering

Just as in software development and risk management, many cyber attacks follow a lifecycle approach, with a cycle of inputs and outputs that constantly improves the process. Social engineering is no different and even has some dedicated lifecycle models. In its simplest form, the social engineering lifecycle follows four basic phases: the initial one is called the *investigation* phase, followed by the *hook* phase, then the *play* phase, and finally the *exit* phase.



The life cycle of social engineering. Source: Imperva

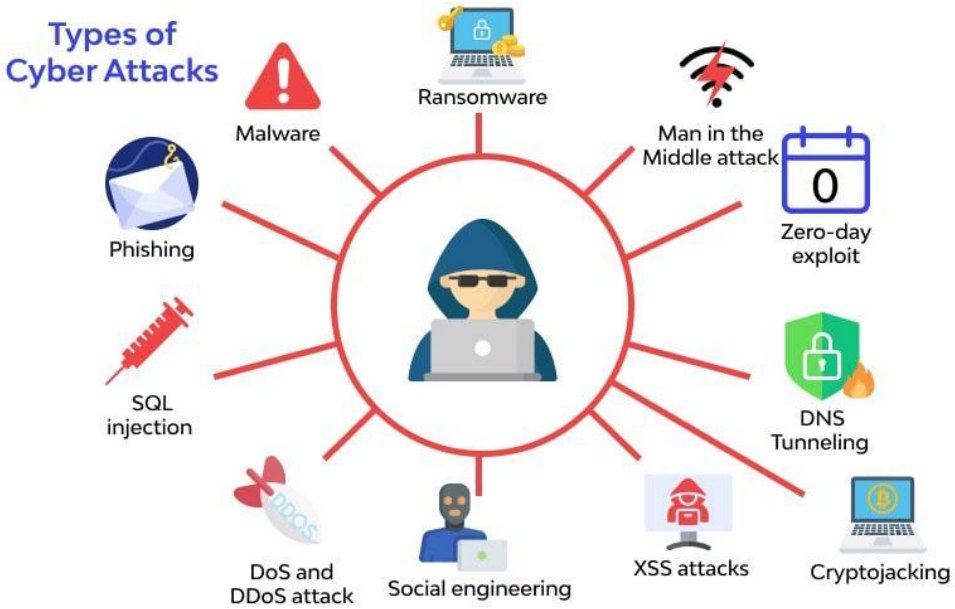
The first phase is the investigative phase, where the attacker focuses on reconnaissance and data collection. This phase is critical to the success of the planned attack and can also be the phase with the longest time duration. During this period, the attacker focuses not only on figuring out what possible physical or cyber security holes can be exploited, but also seeks to understand which individuals can be exploited and how. With the right information, the attacker can determine which attack vectors to use, possible passwords required, responses to expect from various actors, and refine his or her objectives. At this stage, it is also very important to define a clear objective so as to understand what information is important and what can be ignored. During intelligence gathering, small disconnected pieces of intelligence can then form a complete puzzle if joined together correctly. To complete this phase, the attacker often uses OSINT, google dorking, and various other investigative techniques, from social search to physical reconnaissance if necessary. The second phase called the "hook," is the phase where the attacker "throws the bait" i.e., tricks the chosen victims and implements the social engineering attack. For example, if after the first phase the attacker chose to proceed down the path of phishing, this phase might include contacting victims through emails forged to gain their trust, or gathering additional information. The attacker might then make contact with a company's human resources team, and given the extensive research done in the previous phase, might be able to create extremely realistic emails, exactly as if they had been sent by some other member within the team or company. The third phase, "play," is where all the information obtained and relationships established with the victims are exploited. This is when the attacker uses both information and relationships to actively infiltrate the target. In this phase, the attacker focuses on maintaining the pace of compliance established in the second phase without arousing suspicion. Exploitation can occur through the disclosure of seemingly unimportant information or access granted/transferred to the attacker. To give a few examples of successful exploitation, one can think of login credentials disclosed to the attacker via telephone, opening a malware-infected e-mail, inserting a malware-infected USB stick into a company computer, or even just the act of holding open the door or otherwise allowing the attacker to enter the facility. At the end of this phase, the attacker's objective should now be complete, whatever it was. The final phase, "exit," is the exit phase. This indicates the end of the life cycle. The social engineer will try to remove all traces of his presence and end his charade. Everything the attacker has acquired or learned during the process is then used during a new attack cycle to more effectively scam another victim. Social engineering and unwitting users provide a vast attack surface that can be easily exploited. It is therefore necessary to do all you can to be prepared and protect yourself from the scammers of the digital age, and as we have seen, awareness of these dangers is still vastly ignored by the bulk of businesses and individuals alike.

Chapter 3

Examples of malware and social engineering attacks

3.1 The connection between malware and social engineering

In the previous chapters we have seen in depth both malware and social engineering. It has also been mentioned how malware and social engineering work together, the latter being the conduit that is often used to bring malware to its final destination. Before providing several examples related to both topics mentioned, we want to emphasize the importance of the symbiosis between malware and social engineering and how the two strongly interface. As much as they may exist separately, the true strength of different attack vectors arises when well-created malware is combined with well-researched and applied social engineering techniques. In the vast majority of cases, cyber criminals use a combination of social engineering techniques and malware implementation methods so as to maximize the chances of infecting the target machine (or machines): social engineering techniques help to attract the attention of the potential victim, while malware implementation techniques increase the chance in succeeding in penetrating it. From here we also notice the big known difference in mention between the two topics. On the one hand, malware exploits all that are computer security flaws, while social engineering, on the other hand, exploits all possible human weaknesses, which then turn into equally important flaws. If mitigating a hacker attack composed of malware is difficult in its own right, being able to mitigate a well-managed attack through the use of social engineering becomes even more complicated.



Different types of attack vectors. Source : Wallarm

Before showing the different possible attack vectors, let's give a clear definition: in cybersecurity, an attack vector is a specific path, method, or scenario that can be exploited to break into a computer system, compromising its security. An attack vector can be exploited manually, automatically, or through a combination of manual and automatic activities. We also want to define another important term so as not to get confused: an attack surface is the total number of attack vectors that an attacker can use to manipulate a network or computer system or to extract data.

3.2 Classic attack vectors

We saw in the first chapter a distinction between malware and attack vectors. Once we have seen the previous definition, it should then be clear how the two are different: on the one hand we have a malicious program that performs certain actions once executed, and on the other hand we have a particular procedure that once exploited allows the execution of malware or enables an intrusion into a system. Malware is thus one of many attack vectors, but we will not list them since they have already been thoroughly explained in previous chapters. The following is a list, excluding malware, of the most common attack vectors used by hackers:

- *Compromised credentials*: usernames and passwords are still the most common type of login credentials and continue to be periodically exposed in data leaks, phishing attacks, and malware. When lost, stolen or exposed, credentials provide the attacker with immediate access. For this reason, organizations are investing in tools that can continuously monitor exposed data and leaked credentials. Two-factor authentication (2FA), multi-factor authentication (MFA), and biometric scans remain the best solution to protect against this type of attack vector.

- *Weak* credentials *and passwords*: in this case, login credentials are easy to guess or breach by brute force, an attack that uses a dictionary and modifications of different words to try to guess the password by sending multiple login requests. Passwords reused for different services also pose a security risk. For security reasons to date, there are many services that require passwords chosen to be at least eight characters long and contain symbols and numbers.

- *Weak or missing encryption*: Common encryption methods such as SSL certificates and DNSSEC can prevent MiTM (man in the middle, which we will discuss later) type attacks, and can protect confidential data during their transmission. Weak or missing encryption for data means it can be sniffed by a potential hacker or exposed in a data leak or breach.

- *Zero-day exploit*: A zero-day (also known as 0-day) is a vulnerability in software or hardware previously unknown to those who should be interested in its mitigation, such as the vendor of the software or hardware in question. Until the vulnerability is mitigated, hackers can exploit it to adversely affect programs, data, other computers, or a network. Typically, the only way to mitigate a zero-day is to release a patch after the product is placed on the market. It is the vendor who is responsible for ensuring that the flaw is made known since the patch is of extreme importance.

- *Misconfigured devices*: Security misconfigurations result from failure to properly implement security controls on devices, networks, cloud applications, firewalls and other systems. They can include anything from default administration credentials to open ports, from unused Web pages to unprotected files. A good example is a remote desktop protocol (RDP) that is working properly but still has the initial username and administration password. This type of passive attack vector is a problem that affects the organization itself and can lead to data breaches, unauthorized access, and other serious security incidents.

- *DDoS Attacks*: Distributed Denial of Service attacks are types of network attacks that are enacted against other resources in a network such as data centers, servers, websites or web applications and can limit the functionality and availability of the system itself. In this case, the hacker sends an inordinate amount of messages to the resource, which being unable to handle the huge number of requests slows down or even crashes, making it inaccessible to clients. Botnets previously created by the malicious actor are often exploited to do this. CNDs and proxies are possible methods to mitigate risks related to DDoS attacks.

- *SQL injections*: structured query language is a programming language used to communicate with databases. Many servers containing sensitive data use this language to manage their databases. An SQL injection uses malicious SQL code to force the server to expose data that otherwise could not be retrieved. This risk becomes very important if the database contains personal credentials, credit card numbers, or other important private data. Proper database configuration is mandated to stay safe from this type of attack vector.

- *Server and software security* vulnerabilities: new vulnerabilities are discovered and added to the CVE every day. If a developer has not released a patch for a possible generic vulnerability, be it a zero-day or other, it becomes difficult to defend against attacks that exploit these types of problems. Many discovered flaws can affect different systems-just think, for example, of the recent log4j flaw. The CVE is the "common vulnerabilities and exposures," a publicly disclosed list of vulnerabilities launched in 1999 by MITRE to identify and categorize vulnerabilities in software and firmware.

- *Cross-site scripting*: this type of attack, also known as XSS, involves injecting malicious code into a Web site, but it is not the site itself that is being attacked; rather, the goal is to target visitors to the site. A common way attackers can use cross-site scripting attacks is to inject malicious code into a comment, such as by embedding a link to JavaScript code containing malware in the comments section of a blog.

3.3 Social engineering attack vectors

We have seen several types of classic attack vectors that exploit different types of generic vulnerabilities, be they zero-day or weak credentials. However, these types of attack vectors do not fall into the category of vectors used by social engineering because they exploit

computer flaws and misconfigurations. Instead, here we see what are the classic attack vectors used by social engineers and how the different techniques focus not so much on computer flaws but on human weaknesses. As we mentioned earlier, malware is one attack vector, but it can fall into different categories. This is because different social engineering attack vectors can then rely on malware to complete the cyber intrusion, but of course the same thing can happen after using a classic vector seen above.

One of the big differences between the two categories of vectors is that those related to social engineering can be physical procedures, where the attacker performs actions in person and not remotely using a computer. The following is a list of the most common attack vectors used by social engineers and hackers to accomplish their malicious objective:

- *Scareware*: as the name implies, scareware is malware that is intended to scare the user into action, and to act quickly. It often comes in the form of pop-ups or e-mails indicating the need to "act now" to get rid of viruses or malware on one's device. If ignored, scareware is generally harmless, since it is simply pop-ups or emails that cannot actually compromise a system. Should the user fall for the pitfall, however, they will later download other potentially very dangerous malware.

- *Email hacking and spamming contacts*: it is in our nature to pay more attention to the messages we receive from people we know and trust. And social engineers know this all too well: they take over email accounts and spam the relevant contact lists with scams and phishing messages. In this case it is very difficult to mitigate the attack since the source is someone we trust. The result can be a mass distribution of the malware that the hacker has created or a series of scams that bounces from person to person bringing possible serious economic damage to both the individual and a possible company.

- *Access tailgating*: also known as piggybacking, access tailgating occurs when a social engineer physically follows an authorized individual into an area to which they do not have access. This can be an act as simple as holding a door open for someone else. Once inside, they have ample opportunity to access devices containing important information, which can be further exploited if implanted with USB sticks containing self-executing malicious code.

- Phishing: Phishing is perhaps the most famous method of stealing sensitive information or data from an unwitting victim. Let's briefly describe how it typically works: a cybercriminal, in this case the phisher, sends a message to a target asking for a type of information or to perform an action that might be useful in later carrying out a more significant crime. The request can also be as simple as encouraging the user to download an attachment (containing keylogger-type malware) or verify a mailing address or even a security code. It is worth noting that there are many forms of phishing from which social engineers can choose, all with different means of targeting the user. Spam phishing often takes the form of one large "blanket" email, not necessarily targeting a single user. Spear phishing, on the other hand, targets individual users, perhaps impersonating a trusted contact.

Whaling, yet another form, targets celebrities or high-level executives by referring to them as whales or whales, given their importance.

It is also important to mention the different forms of delivery in which phishing can occur. *Vishing*, or voice phishing, is when a phone call where sensitive data is revealed from a victim to the attacker is recorded. In *smishing* (i.e., sms phishing), text messages containing malicious links are sent, leading to malware downloads once clicked. In *angler phishing,* the social engineer poses as a customer service worker to intercept communications and important data. Then there are cases of *in-session phishing*, which occurs when one is already on a platform or account and is asked, for example, to log in again. In this case, the new login will transmit login credentials to the social engineer. In essence, we have seen how phishing is one of the most widely used and evolved forms of carrying out scams and how it can present itself in many different ways from each other.

- *DNS spoofing*: also known as cache poisoning, DNS spoofing occurs when a browser is manipulated so that online users are redirected to malicious Web sites intent on stealing sensitive information. In other words, DNS spoofing is when the cache is poisoned with malicious redirects. This attack vector is most common when the wifi network being used is public, such as that of a bar or hotel.

- *Bating: Bating* (solicitation) is based on the premise that someone will take the bait, that is, offer something desirable in front of a victim, hoping he or she will take the bait. This occurs most often on peer-to-peer sites such as social media, where someone might encourage a victim to download a video game, important documents, or even a movie and then infect the victim with malware via the previously downloaded file. This type of attack also has a physical counterpart, called a dead drop. In this case, a hacker might leave on the floor, perhaps near the entrance to an office or café, a usb flash drive with malware inside. The malicious files might further be categorized with interesting names such as "credit card data" or "important documents." A target who takes the bait will pick up the flash drive and proceed to insert it into his or her machine, infecting it and potentially putting the entire network to which it is connected at risk.

- *Pretexting*: pretexting is the use of an interesting pretext, or ruse, to capture someone's attention. Once the story, usually fabricated, has caught the person's attention, the social engineer tries to trick the potential victim into providing something of value. Often the social engineer poses as a legitimate source, such as a police officer or an electrical inspector at an apartment building business.

- *Watering hole* attacks: A watering hole attack is a sweeping attack that infects a single web page with malware. The web page is almost always on a popular site to ensure that the malware can reach as many victims as possible.

- *Quid* pro quo: quid pro quo means a favor for a favor, basically "I give you this and you give me something else." In the case of social engineering, the victim provides sensitive information such as account logins or payment methods and then the

social engineer does not return his or her part of the bargain. To provide an example in the world of online buying and selling, one might find a normally very expensive product at an extremely low price. One might send a deposit of money to the seller, but the seller will never ship anything and will disappear with the money once received.

- *Physical* breach: as the name indicates, physical breaches occur when a cybercriminal is in plain sight and physically presents himself as a legitimate source to steal data or confidential information. This could be a colleague or IT worker (perhaps a disgruntled former employee) who acts as if he or she is helping the user solve a problem on his or her device. In reality, the goal is to steal the user's account logins, or as much information as possible.

3.4 Noteworthy IT flaws

So far we have talked about different attack vectors and what are the possible vulnerabilities they exploit. It is also important to note, however, that the choice of vector depends on the type of vulnerability to be exploited. Prominent among these, given their importance, are some vulnerabilities discovered in recent years that have affected thousands if not millions of machines including computers and servers. The Heartbleed bug, discovered in April 2014, appeared as one of the largest flaws in Internet history, and compromised the security of as many as two-thirds of the world's servers, at the time more than half a million servers. Heartbleed, discovered by a cyber security company named Codenomicon and a Google researcher named Neel Mehta, is a serious vulnerability in the popular OpenSSL cryptographic software library. This vulnerability allows information protected, under normal conditions, by SSL/TLS encryption used to secure communications over the Internet to be stolen. SSL/TLS ensures the security and privacy of Internet communications for applications such as Web, e-mail, instant messaging (IM) and some virtual private networks (VPNs). The Heartbleed bug allows anyone on the Internet to read the memory of systems protected by vulnerable versions of OpenSSL software. This compromises the secret keys used to identify service providers and to encrypt traffic, user names and passwords, and the content itself. As a result, it allows attackers to spy on communications, steal data directly from services and users, and even impersonate those services and users. The bug specifically is found in OpenSSL's implementation of the heartbeat extension of TLS/DTLS (Transport Layer Security Protocols) (RFC6520), from which it takes its name. Interestingly, this is not an SSL/TLS protocol design flaw, but an implementation problem, i.e., a human programming error. To mitigate the flaw, a new version of OpenSSL was released, but this means that if the vulnerable version was still in use by some entity, it would still be vulnerable today.

The logo assigned by Codenomicon to the leak. Source: Codenomicon

A few years later, on April 14, 2017, the hacker group known as Shadow brokers leaked an exploit, originally developed by the U.S. National Security Agency (NSA). EternalBlue, the name of the exploit in question, exploits a vulnerability in Microsoft's Server Message Block (SMB) protocol implementation. The vulnerability is due to the fact that the SMB server version 1 (SMBv1) in various versions of Microsoft Windows mismanages specially crafted packets from remote attackers, allowing them to execute code remotely on the target computer. To clarify, the SMB protocol is a standard, generally secure system that creates a connection between client and server by sending responses and requests. When printing a document, a person can use their own computer, the client, to send a request to a colleague's computer, the server, with a request to print the document. The client and server communicate via the SMB protocol. The NSA did not alert Microsoft to the existence of EternalBlue for a period of five years, until a breach by the agency itself, at the hands of the aforementioned group, forced the agency to do so. Microsoft blames the agency for the existence of EternalBlue and its consequences and although the flaw is based on a Windows vulnerability, The NSA has refused to discuss it in detail. The leak of this exploit combined with the NSA's long silence led to a global hacker attack in May 2017 where the WannaCry ransomware was spread, infecting more than 230000 computers. This attack spread through computers unpatched against the EternalBlue flaw running Microsoft Windows. As in classic ransomware, users' files were encrypted and a ransom in Bitcoin was demanded for their return. When the ransomware spread beyond Europe, computer systems in more than 150 countries were crippled. The WannaCry ransomware attack had a significant financial impact worldwide. It is estimated that this cybercrime caused $4 billion in losses worldwide. Had it not been for the continued use of outdated computer systems and lack of education about the need to update software, the damage caused by this attack could have been avoided. In even more recent years, on November 14, 2021, a very serious

vulnerability was discovered in the code of a software library used for logging, known as Log4Shell. This is a zero-day vulnerability in Log4j, a popular logging framework written in Java belonging to Apache, which results in the execution of arbitrary code. It is estimated that this flaw was present in more than 100 million instances globally. Security experts consider Log4Shell one of the most serious threats in recent years. This consideration is due to two factors: the huge number of vulnerable systems and the ease with which an attacker can compromise a network. Log4j first logs messages in the software and then analyzes them for errors. Its logging capabilities allow it to communicate with other functions within systems, such as directory services, and it is this that creates the opening for the vulnerability. The main attack is to send Log4j messages that instruct the system to download and execute malware from a remote server, thus granting the attacker greater access to the victim's system. This can, in turn, lead to complete network compromise and theft of sensitive information, as well as the possibility of sabotage. The Log4j library has been in use since 2001, and it appears that the flaw has existed for 13 years. Surprisingly, the vulnerability was discovered by Chen Zhaojun of Alibaba (China's largest e-commerce site) in the popular video game Minecraft: Java edition, where players discovered that inserting a line of malicious code into the game's chat causes it to be logged by Log4j so that commands can be executed. If a malicious user is able to insert a string of code into a form that is logged by Log4j, even if it is a name field used for accessing cloud services, it is possible to exploit this vulnerability. Windows, Linux, and Mac environments are all equally vulnerable. Since December of the same year as its discovery, most vendors have released security updates that address the Log4j flaw within their applications, and Apache itself has released fixes and updated versions that remediate the vulnerability.

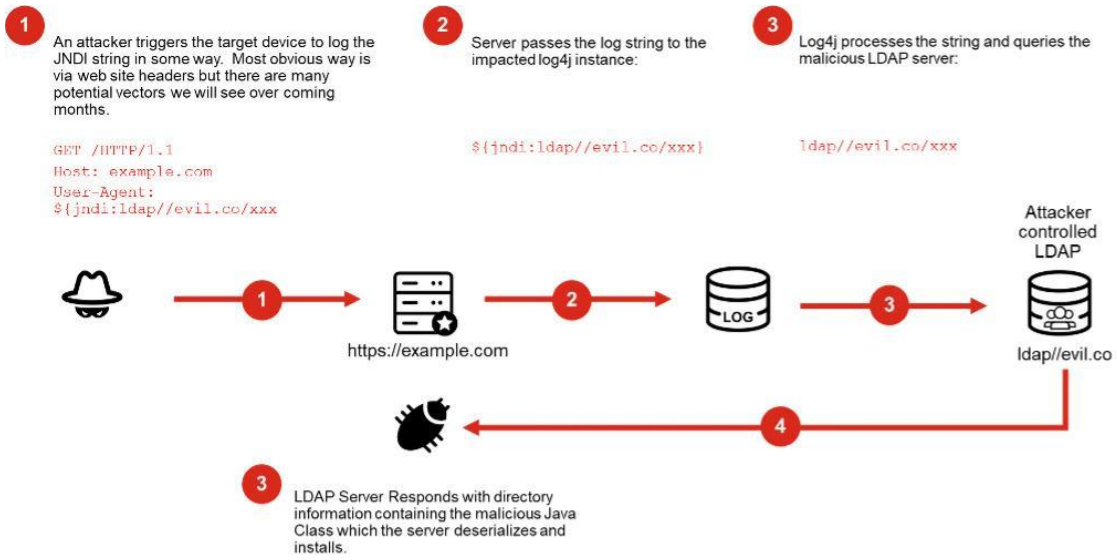


Diagram demonstrating the operation of the log4j flaw. Source: Fortinet

Chapter 4

Preventive measures and information security

In this chapter we want to talk about what are the possible preventive measures and defensive techniques that a user or enterprise can use to defend against hacker attacks and all the dangers that arise from security holes and problems, including possible social engineering attacks. We will talk about antivirus, surveillance systems, monitoring, sinkhole dns (etc.) but also about the attention the user and corporate employee should pay to certain things, i.e., explaining "good practices" on how to stay secure.

4.1 Preventive measures for individuals

When it comes to private individuals, i.e., individual users who use their devices in the comfort of their homes and not for work purposes, they automatically enjoy an advantage over any employees of companies: not being the target of hacker attacks aimed directly at the company, which could make any employee a potential target for obtaining credentials or information. The private individual who uses IoT devices for playful purposes must still be careful to properly secure the various platforms he or she uses. As an example, the more smart devices the user uses, the more care must be taken to make sure there are no security holes. State-of-the-art devices dedicated to smart homes, from surveillance cameras, thermostats and lights that can be adjusted via smartphones to devices such as Alexa, can prove to be a danger if not configured correctly. Research conducted in 2021 reveals that these new smart homes are highly vulnerable to different types of attacks. Among the reported cases of attacks on smart homes we see hackers remotely controlling lights and smart TVs, unlocking IoT-enabled doors, and remotely turning on and transmitting video from smart cameras. As if this were not enough, keep in mind that a compromised smart device in a home network, even a simple light bulb, can lead to a complete penetration of the home network.

Therefore, it is critically important to make sure that all possible credentials associated with different devices installed in the home are not left set to the default ones and that the various passwords are not easy to guess. Another good practice regarding credentials, is not to reuse the same passwords for too many services. In case one password is stolen, it would be easy for a hacker to gain access to several other user accounts and devices. To continue the discussion of possible preventive measures for individuals, we then have a key feature that should be used and maintained at all times, namely the firewall. A firewall is network security software that monitors and filters incoming and outgoing network traffic according to previously established security policies. In its most basic form, a firewall is essentially the barrier between a private internal network and the public Internet. The main purpose of a firewall is to allow non-dangerous traffic to enter and to exclude dangerous traffic. In addition to being built-in for Windows and macOS devices, all routers you use today have a firewall preinstalled. This is done because we do not want to filter out possible dangerous traffic only from our computer, but from our entire local network, which is accessed by the very router we use as an access point to the Internet. Another

essential security measure for the average user is the use of anti-virus software. Anti-virus software (abbreviated as AV software), also known as anti-malware if you want to use the term correctly, is a computer program used to prevent, detect and remove any malware. Antiviruses were originally developed to detect and remove computer viruses, hence the name. However, with the proliferation of other computer threats, antivirus software began to protect against other computer threats, a large proportion of other types of malware. Some products also include protection from malicious URLs, spam and phishing. It must be kept in mind, however, that an antivirus can also lead to some problems when using it. From being careful about the renewal terms of a license that you have purchased to not downloading "rogue AVs" (malware that claims to be antivirus but is not), antivirus can also lead to problems caused by false positives. By false positives we mean when the software identifies a non-threatening file as malware. If the antivirus is configured to immediately quarantine or delete a file deemed dangerous, a common behavior for antivirus dedicated to Windows, a false positive could actually be a file critical to the operation of the operating system that, if removed, could render it unusable. One must also keep in mind that antiviruses are meant to be used in conjunction with firewalls, since the former are concerned with keeping files under control while the latter are concerned instead with the security of Internet traffic, but do not support the removal of infected files. A good practice for any user of IoT devices is to ensure that they are always updated to their latest version and that all patches, if available, are installed. Patches are software and operating system updates that fix security vulnerabilities in a program or product. Usually this particular type of update is issued directly by the vendor of the product or software, and is treated as a common update. We have seen in previous chapters how large security holes (think of the recent Log4shell) were almost immediately fixed through patches and security updates, and so it is easy to see how small, common updates can actually be very important. Always staying with the good practices to follow for the security of one's devices, it is important to know how to also behave when they are being used. Ignoring spam and not falling into pitfalls is crucial, but this can only happen if the user is aware of what the risks are and how to act about them. Therefore, avoiding clicking on links received from strangers and not trusting almost anyone becomes imperative. It is also very important to keep backups of files that you consider important or of the entire hard drive of your computer, in case it is corrupted or locked via ransomware by a hacker. One last practice that may be complicated for some not-so-experienced users, but very useful, is virtualization. Not everyone needs to go this route, but one often visits dodgy websites, one must expect to be bombarded with spyware and malware. Although the best way to avoid browser-derived intrusions is to steer clear of unsafe sites, virtualization allows you to run your browser in a virtual environment, such as Parallels or VMware Fusion, that ignores the operating system to keep it more secure. With this extra layer of security, often called a "sandbox," whatever happens to the system inside the virtual machine is of little consequence, because it can always be shut down and wiped, without worrying much since it should not contain any sensitive or important data, held instead on the actual operating system of our machine.

4.2 Preventive measures for enterprise

When it comes to businesses, whether they are small or large, various systems are used to prevent possible hackers from accessing the company's network or machines. In this case,

it is important to note that the machines, the network and even the individuals employed by the company, are vulnerable. It is therefore critical to be able to protect all of these things together. To do this, businesses normally adopt what are called Information Security Policies, a set of guidelines that help a business protect the data of their customers and employees simultaneously. These policies also serve to ensure that the business does not violate any legal or ethical standards. In addition, they provide a framework for the company to make sure it is prepared for any security breaches. Maintaining information security is a difficult task. It requires constant vigilance and attention, and it can be challenging to keep up with the latest developments in the field. There are many different ways to maintain information security, and all companies should take steps to protect their data. One of the first ones we see is the same as seen above for individuals, but with even more focus: changing passwords regularly. We have already pointed out that default passwords should always be customized for each individual user. However, the strength of passwords in terms of numbers and special characters is only the first line of defense. Usernames and passwords are the easiest gateway for hackers to access the devices and internal networks of companies, consequently these companies tend to have policies whereby each user's passwords must be changed at least once a month. Another important factor that is usually ignored in the private sphere, however, is the choice of the correct ISP (Internet Service Provider). A castle is only as strong as its walls, and the same goes for the ISP of companies. It may not be obvious, but not all ISPs are built nor do they function in the same way. When choosing an Internet provider, one should not limit oneself to speed and price. There is a plethora of service providers on the market, so it is wise to choose an Internet package that has built-in security features. After evaluating an ISP's online security, it comes down to its convenience and connection speed. It is best to find one that meets all three criteria. Let us also return to the firewall discussion, but in more detail. In the case of a business, different types of firewalls are often implemented, and different switches and hardware devices dedicated solely to packet control and filtering are also used. Normally an enterprise does not use just one type of firewall, but several types: the first type we see are the classic "packet filtering firewalls," the most basic type and used by individuals. They operate at the network level and control IP addresses of sources and destinations, protocols, ports and destination ports of connections based on a priori defined rules. Moving on to the second category we have firewalls that operate at the session level and are called circuit-level gateways. These verify established TCP (Transmission Control Protocol) connections and keep track of active sessions. They are quite similar to packet filtering firewalls in that they perform a single check and use minimal resources, but they operate at a higher level of the Open Systems Interconnection (OSI) model. First, they determine the security of an established connection: when an internal device initiates a connection with a remote host, circuit-level gateways establish a virtual connection on behalf of the internal device to keep the identity and IP address of the internal user hidden. They are cost-effective, simplistic, and have minimal impact on network performance.

Turning then to the third type of firewall called stateful inspection firewalls, we have a firewall that operates at an additional step beyond the previous one. Stateful inspection firewalls verify and track established connections and perform packet inspection to provide better and more comprehensive security. They work by creating a state table with source IP,

destination IP, source port and destination port once a connection is established. They dynamically create their own rules to allow expected inbound network traffic, instead of relying on a set of rules coded based on this information. They quickly discard data packets that do not belong to a verified active connection. Then there is a fourth and final type of firewall, called application-level gateways or even more commonly proxy firewalls. These are implemented at the application level via a proxy device. Instead of accessing the internal network directly, the connection is established through the proxy firewall. The external client sends a request to the proxy firewall. After verifying the authenticity of the request, the proxy firewall forwards it to one of the internal devices or servers on behalf of the client. Alternatively, an internal device may request access to a Web page, and the proxy device forwards the request while concealing the identity and location of the devices and network. This talk of policies, secure credentials, and firewalls should also be supplemented with other best practices that all companies should follow. For example, all personal and financial data of a company's customers should be encrypted. If you then want to achieve an additional layer of security, this applies to any kind of sensitive data that is hosted within the company's databases or systems. Another way to ensure that employees remain safe from certain attack vectors, among which we mainly see the types related to social engineering, is to restrict access to the Internet and block unnecessary sites. Restricting access to certain information online reduces the chances of security breaches, so it is a good idea to make sure that only necessary users have access to certain data. Similarly, blocking the viewing of certain sites reduces the possibility of virus and spyware-carrying sites being opened within the corporate network. Finally, we cannot again fail to mention the section regarding patches and updates. It is imperative that an enterprise's IT systems are always updated to the latest version to avoid exploitation of possible flaws in older versions. To conclude we also want to mention again that employees' lack of awareness of potential dangers remains one of the greatest risks of security intrusions. Users and employees must be aware of the risks and threats to the systems and information they use. Users need to be trained on how to recognize attempts to access sensitive information through email, phone calls, or other means.

4.3 The process of Information Security

Any organization that relies on computer networks and the Internet must prioritize Information Security, or Information Security. This term simply refers to the strategies, products and procedures implemented by security experts to prevent unauthorized access or insider threats to sensitive information, whether stored in the cloud or on physical storage devices. The information security expert is responsible for ensuring that this data is not altered or modified in any way. Information Security is thus a process that moves in phases, building and strengthening along the way. Although this process involves many strategies and activities, these can be grouped into three distinct phases: prevention, detection and response. Each of these phases requires strategies and activities that move the process to the next stage. The dynamic growth of new threats and vulnerabilities requires extremely rapid adjustment of the methodologies of the prevention, detection and response cycle, and a change in one phase affects in some way the subsequent phases and thus the entire process. For effective management of the latter, each phase must be designed with great appropriate capabilities and oversight to ensure its maturity The

ultimate goal of the Information Security process is to protect three unique attributes of information:

- Confidentiality: sensitive information should be seen only by those people authorized to view it. The information may be confidential because it is proprietary information of the organization, or it may be personal information of customers that must be kept confidential because of legal responsibilities.
- Integrity: Information must not be corrupted, degraded or modified. Measures must be taken to isolate information from accidental or intentional modification.
- Availability: information must always be available for viewing by authorized personnel when necessary.

Hacker attacks compromise systems in ways that affect one or all of these attributes, and this must be avoided. An organization is able to protect these attributes through proper planning, which can also greatly reduce the risks of an attack and greatly increase detection and response capabilities should an attack take place. Let us now look at the processes in detail.

4.3.1 Prevention

Threat prevention generally refers to tools that perform threat detection and prevention actions, such as endpoint detection and response, or cybersecurity policies and strategies that prioritize prevention techniques. During the prevention phase, it is then necessary to design and implement security policies, controls, and processes. All of these are interrelated and need to be developed at an early stage. The information security policy is the cornerstone on which everything else is based. So the first step is to create security policies that determine what needs to be protected and how. This is followed by awareness programs that educate employees on the importance of security, the use of such measures, and their responsibilities related to the topic. The last step is then to implement access control. Not all users should have access to the system and the information contained within. Access should be restricted and granted on different levels to those who need it. This is usually done by issuing identifiers and verification methods that employees can use to access what they need.

4.3.2 Detection

Threat detection is the practice of 'analyzing the entire security ecosystem to identify any malicious activity that could compromise the network. If a threat is detected, mitigation measures must be taken to neutralize it before it can exploit the vulnerabilities present. When it comes to detecting and mitigating threats, speed is of the essence. Security programs must be able to detect threats quickly and efficiently. A company's defense programs are ideally able to block most threats because they have often been seen before and are therefore considered "known." However, there are other "unknown" threats that an organization aims to detect. This means that the organization has not encountered them before, perhaps because the attacker is using new methods or technologies. Various techniques and programs are used to try to detect any threat while ignoring its type: threat

intelligence is a way of examining signature data from attacks that have already been seen and comparing it with corporate data to identify threats. It is particularly effective in detecting known threats, but not unknown ones.

Threat intelligence is often used to great effect in the SIEM (Security Information and Event Management) technologies we will discuss extensively later, antivirus, IDS (Intrusion Detection System) and web proxy. UEBAs (Users and Entity Behaviour Analytics) are used in conjunction with SIEMs, through which an organization is able to figure out what an employee's normal behavior is: what kind of data they access, what time they log on, and where they are physically located, for example. Unexpected behavior will set off alarm bells, which will open an investigation into the event to determine the reasons for it. A final method of detection, which derives heavily from prevention, is the creation of traps, so-called "honeypots." Some targets are too tempting for an attacker. Security teams know this, so they set traps in the hope that the attacker will take the bait. When an attacker aims for this bait, an alarm goes off to let the security team know that there is suspicious activity in the network that needs to be investigated.

4.3.3 Answer

Incident response (called Incident Response or IR) refers to an organization's processes and technologies to detect and respond to cyber threats, security breaches, or cyber attacks. The goal of incident response is to prevent cyber attacks before they occur and to minimize costs and business disruptions resulting from any unprevented cyber attacks. Ideally, an organization defines its incident response processes and technologies in a formal incident response plan (called an IRP) that specifies exactly how different types of attacks are to be identified, contained, and resolved. An effective incident response plan can help cybersecurity teams identify and contain cyber threats and restore affected systems more quickly, reducing lost revenue, regulatory penalties, and other costs associated with these threats. The two best-known Incident Response frameworks were developed by NIST and SANS to provide IT teams with a foundation on which to build their incident response plans. The following are the steps of the SANS framework, which we focus on as more detailed.

1. *Preparation*: No organization can organize an effective response to incidents at a moment's notice. It is necessary to have a plan for preventing and responding to events. This involves organizations taking a comprehensive inventory of their IT infrastructure, including networks, servers, and endpoints, and assessing its importance. To assess importance, organizations must determine which IT assets contain critical or sensitive information. In addition, they must create a baseline for normal activities through monitoring. As part of the preparation, security teams must also create guidance on how to handle common types of incidents and identify which types of incidents require in-depth investigation.

2. *Identification*: Identification involves collecting data from IT systems, security tools, publicly available information, and people inside and outside the organization, and identifying signs that an incident may occur in the future (precursors) and data showing that an attack has happened or is happening now (indicators). The analysis

involves identifying a baseline or normal activity for the affected systems, correlating related events, and checking for deviations from normal behavior.

3. *Containment*: If an incident is identified, the next step is containment: security teams must work to isolate the attack and prevent it from spreading. This may involve segmenting the network under attack as part of short-term containment. Once short-term measures have been taken, security teams can focus on long-term solutions, which may involve rebuilding entire systems.

4. *Eradication*: Eradication refers to the actual removal of malware or other artifacts introduced by the attacks and the complete restoration of all affected systems. The SANS eradication process first involves re-imaging, which is a complete wipe of the affected system's hard drives to ensure the removal of any malicious content. An attempt is then made to understand what caused the incident to prevent future compromise. After that, basic security best practices are applied, such as updating old versions of software and disabling unused services. Finally, next-generation anti-malware or antivirus software is used to scan the affected systems and ensure that all malicious content is removed.

5. *Recovery*: This phase involves restoring affected systems that were disabled during the time of the incident. Security teams must test and monitor affected systems to ensure that attacks are not repeated and normal functionality is achieved.

6. *Lesson Learned*: Shortly after the attack, teams need to look back and evaluate how the incident was handled and analyze how to improve the incident response process for future incidents. This means writing comprehensive documentation about an incident, publishing reports of the incident, and identifying how to improve the security team's performance. A meeting is then conducted with team members to discuss the incident and cement lessons that can be learned and applied immediately.

This concludes the process of responding to a cyber threat. We have seen how the Information Security process, despite its corpus, is fundamental to the proper protection of an organization's information system. We will talk in a moment in more detail about the applications that are used.

Chapter 5

SIEMs as a security measure

One security measure used by organizations around the world, in conjunction with other tools, is the use of SIEMs. In the field of information security, SIEMs are an abbreviation for "security information and event management," and it is a particular subfield where products and services such as "security information management" (SIM) and "security event management" (SEM) are combined, thus giving rise to SIEM systems. In this chapter

we want to introduce these systems and explain what they are, how they work, what their capabilities are, and generally provide a sufficient knowledge base to delve into their design, installation, configuration, and maintenance, which will be recounted in the next chapter.

5.1 What are SIEMs and how they came about

As we just mentioned, by combining security information management (SIM) and security event management (SEM), we get security information and event management (SIEM). This provides real-time monitoring and analysis of events, as well as tracking and recording of security data for compliance or audit purposes. Simply put, SIEM is an information security solution that helps organizations recognize potential security threats and vulnerabilities before they have a chance to disrupt business operations. SIEM detects anomalies in user behavior and uses artificial intelligence to automate many of the manual processes associated with threat detection and incident response, becoming a staple of modern security operations centers (SOCs) for security management and compliance use cases. Over the years, SIEM has matured to become more than the simple log management tools that preceded it. Today, SIEM offers advanced user and entity behavior analysis (UEBA) through the power of artificial intelligence and machine learning. So we are looking at a highly efficient data orchestration system for managing evolving threats, as well as for regulatory compliance and reporting.

To understand where we have gotten to today with the use of SIEMs, it is first important to understand how we got to this point. Like most modern security systems for technologies, the history of SIEMs is not very long. In fact, the first commercial SIEM product dates back less than two decades to the beginning of the new millennium. The arrival of the first generation of SIEM platforms ushered in a new dawn in the field of data security, combining security event management with security information management for the first time, something not yet seen before. However, these SIEM 1.0 platforms had the huge problem of scaling only vertically, which severely limited their growth. As larger and larger hardware is required to handle data loads, getting to an I/O management limit becomes inevitable, at which point scalability breaks down. The second generation of SIEM, released about 2011, arrived just in time, and not surprisingly, the main difference from SIEM 1.0 is scalability, moving away from the centralized database and instead using big data to enable horizontal scalability. SIEM 2.0 also enabled improved reporting and dashboards, as well as querying historical data for the first time. However, while scalability was SIEM 2.0's greatest strength, it also eventually became its curse. Not because it did not work, but because it simply moved the problem further down the operational pipeline. Before SIEM, security professionals were essentially blind, unable to see at the data level what was happening in their IT environments. The first generation of SIEM gave them sight, but the second generation took it away again by presenting more data than they could handle. It also failed to innovate the alerting aspects of SIEM, leaving teams to depend on preconfigured alerts that, at best, correlated only with certain elements. In its new and most recent version introduced circa 2015, SIEM 3.0, analytics is added for the first time, through the application of machine learning. What makes SIEM 3.0 different from previous versions, and for the first time truly viable, is the shift from preconfigured alerts to a risk-based

approach. Alerts remain valuable when looking for simple, known facts, but in modern security management teams are just as likely to encounter unknown zero-day threats. Analytics-based security monitoring applies statistical techniques to massive amounts of data to build operational models, which are baselines for each individual user and entity in the environment. This technique is known as user and entity behavior analytics (UEBA).

5.1.1 What are UEBAs

Given their importance and strong use within systems, it is important to explain what we are talking about when we talk about UEBA systems. UEBA stands for User and Entity Behavior Analytics and was formerly known as user behavior analytics (UBA). UEBA uses large datasets to model the typical and atypical behaviors of people and machines within a network. By defining these baselines, it is able to identify suspicious behaviors, potential threats and attacks that traditional antivirus might miss. This means that UEBA systems are able to detect non-malware-based attacks because they analyze various behavioral patterns. UEBA also uses these patterns to assess the threat level, creating a risk score that can help guide the appropriate response. Increasingly, UEBA uses machine learning to identify normal behaviors and flag risky deviations that suggest insider threats, lateral movement, compromised accounts, and attacks. Returning to the discussion of SIEMs, in addition to noticing suspicious network behaviors, SIEMs have evolved to include user and entity behavior analysis and thus integrate UEBA systems within them.

5.2 How SIEMs work and what are their capabilities

At the most basic level, all SIEM solutions perform some level of data aggregation, consolidation, and sorting functions to identify threats and meet data compliance requirements. Although some solutions vary in capability, most offer the same basic feature set:

Log management: SIEM captures event data from a wide range of sources across an organization's entire network. Logs and flow data from users, applications, resources, cloud environments and networks are collected, stored and analyzed in real time, giving IT and security teams the ability to automatically manage event logs and network flow data in one centralized location. Some SIEM solutions also integrate with third-party threat intelligence feeds to correlate internal security data with previously recognized threat signatures and profiles. Integration with real-time threat feeds allows teams to block or detect new types of attack signatures not previously known.

Event correlation and analytics: Event correlation is an essential part of any SIEM solution. Using advanced analytics to identify and understand intricate data patterns, event correlation provides insights to quickly detect and mitigate potential threats to corporate security. SIEM solutions significantly improve mean time to detection (MTTD) and mean time to response (MTTR) for IT security teams by offloading the manual workflows associated with in-depth analysis of security events.

Incident monitoring and security alerts: Because they enable centralized management of on-premise and cloud-based infrastructure, SIEM solutions can identify all entities in the IT environment. This enables SIEM technology to monitor security incidents among all

connected users, devices and applications, classifying anomalous behavior as soon as it is detected in the network. Using predefined and customizable correlation rules, administrators can be alerted immediately and take appropriate action to mitigate the problem before it materializes into more significant security issues.

Compliance management and reporting: SIEM solutions are a popular choice for organizations subject to various forms of regulatory compliance. With the automated data collection and analysis it provides, SIEM is a valuable tool for collecting and verifying compliance data across the entire enterprise infrastructure. SIEM solutions can generate real-time compliance reports for PCI-DSS, GDPR, HIPPA, SOX and other compliance standards, reducing the burden of security management and detecting potential breaches early enough to address them. Many SIEM solutions come with pre-built, off-the-shelf add-ons that can generate automated reports designed to meet compliance requirements.

5.3 The benefits of using SIEMs

No matter what the size of a given organization, taking proactive measures to monitor and mitigate cybersecurity risks is essential. SIEM solutions offer companies several benefits and have become a key component in streamlining security workflows to date. Among the various benefits given by their use, we immediately see the importance of advanced real-time threat recognition. Infrastructure-wide monitoring solutions significantly reduce the time it takes to identify and respond to potential network threats and vulnerabilities, helping to strengthen the security posture as the organization expands. We also see excellent control of regulatory compliance. SIEM solutions allow compliance audits and reports to be centralized across an entire enterprise infrastructure. Advanced automation simplifies the collection and analysis of system logs and security events, reducing the use of internal resources and meeting rigorous compliance reporting standards. Reviewing the machine learning discussion, we thus see automation driven by artificial intelligence. Next-generation SIEM solutions integrate with powerful orchestration, automation, and security response (SOAR) capabilities that save IT teams time and resources in managing enterprise security.

The consequence of everything we have seen so far is thus an improvement in organizational efficiency. Because of the improved visibility of IT environments it provides, SIEM can be an essential factor in improving interdepartmental efficiency. With a single, unified view of system data and an integrated SOAR, teams can communicate and collaborate efficiently when responding to perceived events and security incidents. Given how rapidly the cybersecurity landscape is changing, organizations must rely on solutions that can detect and respond to both known and unknown security threats. Using integrated threat intelligence feeds and AI technology, SIEM solutions can successfully mitigate modern-day security breaches, such as classic social engineering attacks, among which phishing always returns, but also SQL injections, DDoS attacks and data exfiltration. Another huge benefit given by using SIEM solutions is the ability to conduct digital forensic investigations once a security incident occurs. It is then easy for organizations to efficiently collect and analyze log data from all digital assets in one place. This way they can recreate past incidents or analyze new ones to investigate suspicious activities and implement more effective security processes. Thus, we have seen how the use of SIEM solutions offers

several advantages, mainly centered on making tasks that were previously tedious and time-consuming both in terms of resources and timing easier.

5.4 Terminology and components

To give correct and complete definitions, it is necessary to be clear about the terminology that is used to describe SIEMs, the capabilities they offer, and the components they integrate. In addition to clarifying several previously explained terms that might be easy to confuse, we will introduce new terms useful for understanding the infrastructure of SIEM systems at a greater level of detail. In addition to terminology, we also want to analyze what components can be part of a SIEM system. So let us begin with a roundup of terms to which we can assign a clear meaning and function.

- When we talk about *device,* or device, we use this generic term to describe servers, firewalls, switches, workstations etc. The term network device, or network device, goes into more detail and refers to all and only those devices that interconnect the network such as firewalls, routers, switches, but excludes servers and workstations instead. The term CMDB stands for configuration management database. The CMDB lists all the devices that are sending logs to the SIEM. Each device in the CMDB shows its health status as well as the current number of events per second (EPS).

- A data collector, or data collector, is a device or server that collects different types of data including logs, events, network flows etc. The data collector then proceeds to send the above toward the SIEM system. Syslog is a standard logging protocol that allows a device to send logs to a listening server. The listening server can be a log server that then forwards to the SIEM, or it can be the SIEM system host directly. The forwarded logs contain a code that specifies their nature and a level of severity. There are several types of syslog including syslog-ng, rsyslog or syslogd. An event is a specific entry in the log file that represents a particular event, such as a blocked connection or a login failure.

- It is also important to define a strong difference between *events* (and thus logs) and *flows,* or streams, which represent network activity by normalizing IP addresses, ports, byte and packet counts, and other data into stream records, which are in fact records of network sessions between two hosts. Flows are specific to SIEM QRadar, which we take as an example below.

- *Rule* - SIEM analyzes event attributes and correlates logs with other SIEM devices. The logs are compared with rules, which look for a pattern of events that matches specific criteria. When a pattern is found, an incident is triggered.

- An *incident* is a unique instance of a rule. Incidents provide the definition of the rule and the events that triggered it.

- A *false positive* is when you activate a rule that does not represent a true security incident. For a more in-depth look at false positives, see the "False Positives" section.

- An *exception* adds a condition to a rule to prevent it from being triggered when specific conditions are met. For example, a vulnerability scanner that runs regularly would generate an excessive amount of tickets even though the traffic is legitimate. An exception would be added to reduce the rate of false positives created by the vulnerability scanner.
- *EPS*, or events per second that a device sends to SIEM. Changes in EPS can indicate that a device needs to be checked for configuration or security issues.

Let us now discuss the components of a SIEM system. The architectures of these systems may vary by vendor, but generally the essential components that make up a SIEM are always the same. It must be kept in mind that a SIEM system is a collection of distinct elements that build and create a system, rather than a single application. These components, which we also saw and explained earlier, are:

- *Log management*: this component deals with the collection, management and storage of received data. SIEM captures both event data and contextual data. The SIEM architecture basically collects event data from organized systems such as installed devices, network protocols, storage protocols, and streaming protocols.
- *Normalization of logs*: The SIEM receives incoming event and context data, but this data is still unreadable because it is too corpus. Normalization is a necessary process that deals with the transformation of event data into useful security information. This procedure includes the removal of irrelevant data from the received data by a filtering procedure. Most importantly, only relevant data are retained for future examination.
- *Sources of logs*: The devices mentioned earlier, both network and non network, all generate logs. This procedure is essentially about how organizations feed logs into the SIEM for security.
- *Correlation of data*: Data must be presented in a relevant and organized way because they are acquired from different devices. The correlation function helps to present a broader picture of data collected from multiple points.
- *Real-time monitoring*: Users receive real-time information about any type of security breach. As a result, the threat can be tracked and eliminated in a timely and effective manner.
- *Automation*: Any event can be responded to automatically with SOAR (Security, Orchestration, Automation, and Response), which eliminates the need for security analysts.
- *Dashboards*: SIEM dashboards make it easy for security analysts to understand changes in data patterns. As a result, a security analyst can quickly and readily notice any irregularities in the network.

- *Reporting*: Other administrators can use SIEM's reporting tool to generate various reports, reducing uncertainty about their reporting activities. SIEM generates reports quickly by storing all log data in database tables.

5.5 Use cases of SIEMs

We have already discussed the capabilities, terminology and components of SIEM systems. Let us now look at the implementation use cases of these systems. Security researcher Chris Kubecka, a cyberwarfare expert, identified the following use cases by presenting them at the 28C3 (Chaos Communication Congress) hacking conference:

- Visibility and anomaly detection of SIEM systems can help prevent various types of zero-days or polymorphic code vulnerabilities. This arises mainly because of the low detection rates of antivirus against this type of rapidly evolving malware.
- Log analysis, normalization and categorization can occur automatically, regardless of the type of computer or network device, as long as it is capable of sending a log. This is another important feature of SIEM systems because it allows them to ingest huge amounts of data that would be, as they were in early versions of these systems, too large for security teams to handle.
- Visualizing data with a SIEM using security events and log anomalies can help detect patterns. These patterns can then be refined and used by the security teams themselves to improve in both effectiveness and speed, especially in detecting security anomalies.
- Protocol anomalies, which may indicate misconfiguration or a security problem, can be identified with a SIEM using pattern detection, alerts, baseline, and dashboards. This allows a SIEM system to detect potential misconfigurations of devices, which could then lead to major security holes within an organization.
- SIEM systems are capable of detecting covert and malicious communications and encrypted channels. This capability proves very useful, especially in the case of insider threats, which can thus be foiled quickly and without damage to the company.
- SIEMs can even accurately detect cyber wars, identifying both attackers and victims. Several countries around the world have interests in this type of activity, and already several cyber wars are active at this time. SIEMs thus prove to be an essential tool even for governmental entities, whether or not they are involved in this type of activity.

Chapter 6

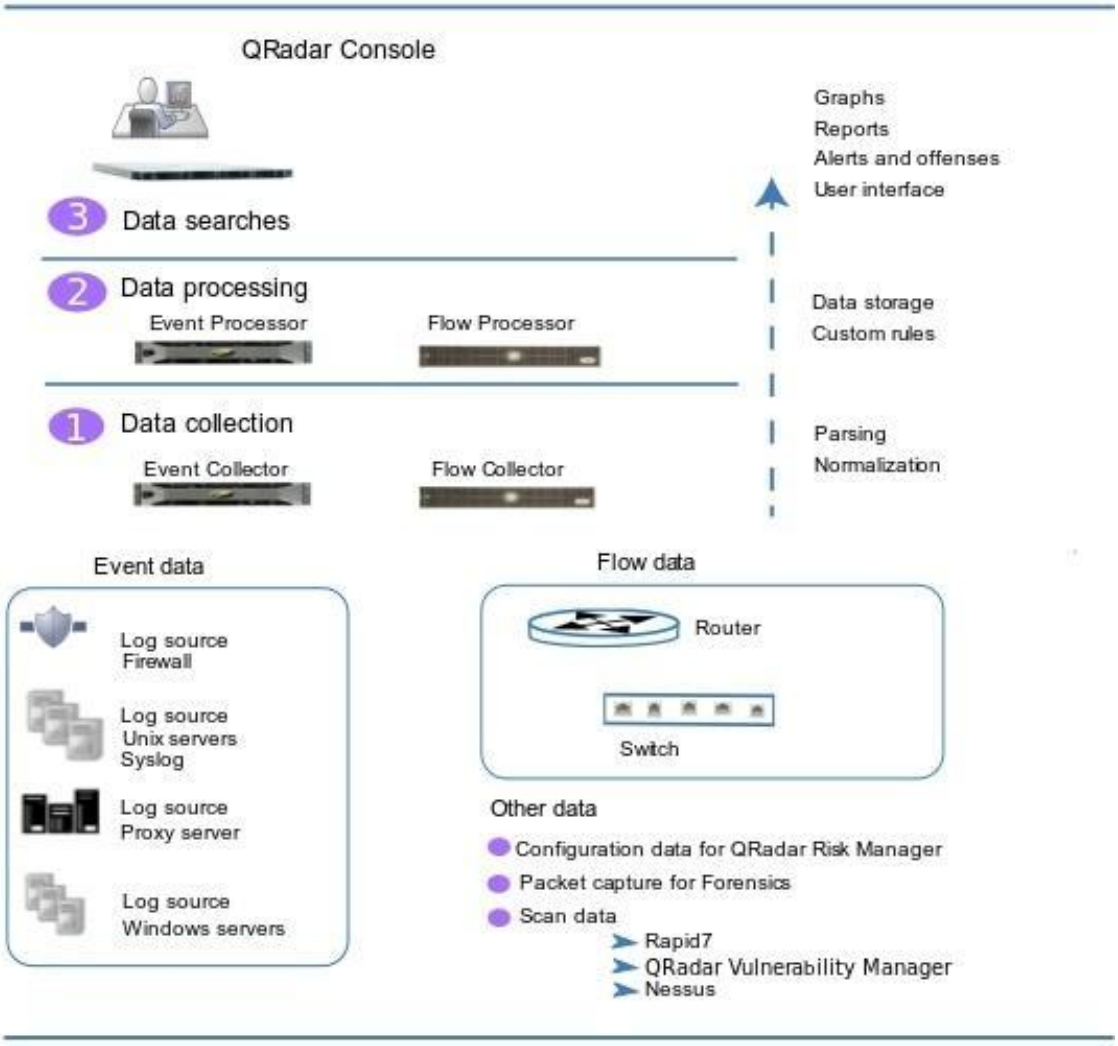
Design, installation, configuration and maintenance of a SIEM system

In this chapter we will proceed with a demonstration of the design, installation, configuration, and maintenance of a SIEM system. The system chosen, as part of the internship with the University of Parma security team, is QRadar SIEM Community Edition provided by IBM. This particular edition of QRadar is a free, full-featured, low-memory, low EPS version that includes a perpetual license. This version is limited to 50 events per second and 5,000 network streams per minute, supports apps, but is based on a smaller footprint for non-business use. Despite these limitations it still provides a solid base to work on and the workflow related to design and next steps remains the same as the full edition.

6.1 The architecture of IBM QRadar

First let's look at how the SIEM architecture is structured: QRadar collects, processes, aggregates, and stores network data in real time. It then uses this data to manage network security by providing real-time information and monitoring, alerts and offenses, and responses to network threats. It is a modular architecture that provides real-time visibility of the IT infrastructure, which can be used for threat detection and prioritization. It is possible to scale QRadar to meet the needs of log and flow collection and analysis, and it is also possible to add integrated modules to the platform, such as QRadar Risk Manager, QRadar Vulnerability Manager, and QRadar Incident Forensics, but these are available in the full version of the product.

The operation of the QRadar security intelligence platform consists of three layers and applies to any QRadar deployment structure, regardless of its size and complexity. The diagram below shows the layers that make up the QRadar architecture.



The architectural layers of QRadar

The three levels seen represent the main functionalities of any QRadar system.

6.1.1 Data collection:

Data collection is the first level, where data such as events or flows are collected from the network. The All-in-One appliance can be used to collect data directly from the network or collectors such as QRadar Event Collectors or QRadar QFlow Collectors can be used to collect event or flow data. The data are analyzed and normalized before moving to the processing layer. When the raw data is analyzed, it is normalized to present it in a structured and usable format.

QRadar's main functionality focuses primarily on collecting event data and collecting flows, also known as flows and events. It is necessary to delineate a distinction between the two: event data represent events occurring at a given time in the user's environment, such as user logins, e-mails, VPN connections, firewall denials, proxy connections, and any other event that you want to record in device logs. Flow data, on the other hand, is information

about network activity or sessions between two hosts in a network, which QRadar translates into flow records. QRadar translates or normalizes the raw data into IP addresses, ports, byte and packet counts, and other information into flow records, which effectively represent a session between two hosts.

6.1.2 Data processing:

After data collection, the second level or data processing layer is where event data and flow data are run through the Custom Rules Engine (CRE), which generates infractions and alerts, and then the data are written to the archive.

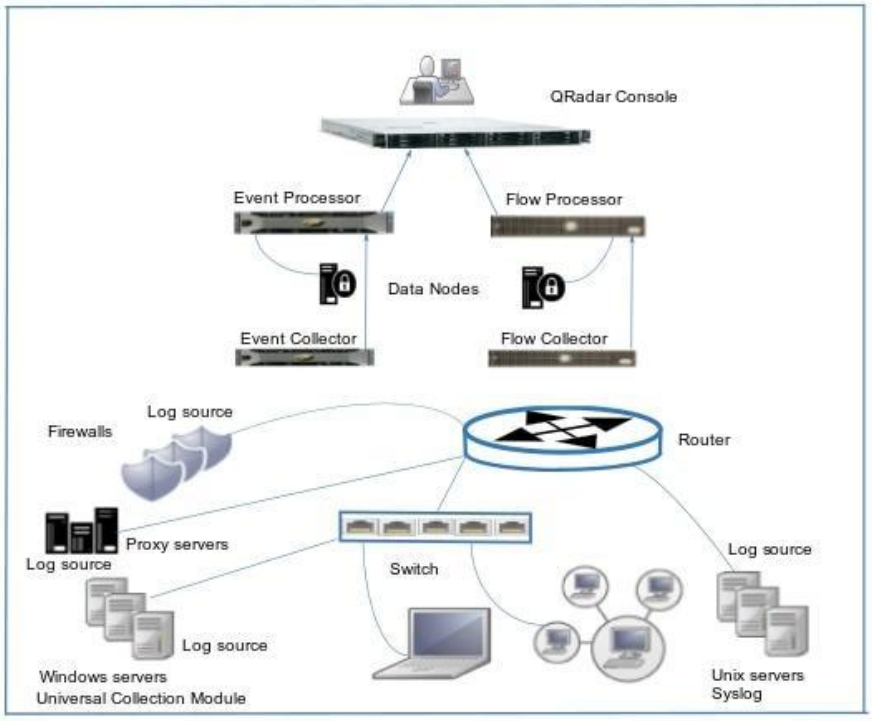
Event data and flow data can be processed by an All-in-One appliance without the need to add event processors or flow processors. If the All-in-One appliance's processing capacity is exceeded, it may be necessary to add event processors, stream processors, or any other processing appliance to handle the additional requirements. More storage capacity may also be needed, which can be handled by adding Data Nodes.

6.1.3 Data searches:

In the third, or top level, the data collected and processed by QRadar are available to users for research, analysis, reporting, alerts or crime investigation. Users can search and manage security administration activities for their network from the QRadar Console user interface. In an All-in-One system, all data is collected, processed and stored on the All-in-One appliance. In distributed environments, the QRadar Console does not perform event and stream processing or archiving. Instead, the QRadar Console is used primarily as a user interface, and users can use it for searches, reports, alerts, and surveys.

6.2 Choice of distribution

When you want to deploy a SIEM system in an organization you must first start with the deployment of the system itself. IBM QRadar's architecture supports deployments of various sizes and topologies, from a single-host deployment, in which all software components run on a single system, to multiple hosts, in which appliances such as Event Collector and Flow Collector, the Data Nodes, App Hosts, Event Processors, and Flow Processors, have specific roles and are located on different machines. Before planning the implementation, consider the following questions: How does the company use the Internet? Is upload being used as much as download? How many events per second (EPS) and flows per minute (FPM) do you need to monitor? How much information needs to be stored and for how long? The most widely used type of deployment for a medium-sized organization, on which we focus, is the All-in-One appliance. An All-in-One appliance includes the capabilities of collection, processing, storage, monitoring, search, reporting, and breach management, all on the same machine. The following diagram shows the components of QRadar that can be used to collect, process, and store event and flow data in the QRadar implementation.



The components of QRadar

The Event Collector collects event data from network log sources and sends it to the Event Processor. The Flow Collector collects flow data from network devices, such as a switch SPAN port, and sends it to the Flow Processor. Both processors provide the data to the collectors and the QRadar Console. The processor appliances can store the data, but they can also use Data Nodes to store the data. The QRadar Console appliance is used for monitoring, searching data, creating reports, managing infractions, and administering the QRadar implementation.

In our case, within the University, the decision was made to choose the deployment of an All-in-One appliance within the mathematics faculty, which has several powerful machines for computation and research. One machine was therefore dedicated entirely to the QRadar appliance. It was also decided to create on a different machine a honeypot, using Artillery, to be connected later to QRadar. The purpose of Artillery is to provide a combination of honeypot, file-system monitoring, system strengthening, real-time threat intelligence gathering, and server health and monitoring tool; to create a comprehensive way to protect a system. Project Artillery was written to be an addition to a server's security and make it very difficult for attackers to penetrate a system. This choice was made because since it is not possible to configure dozens of machines and workstations to send logs to the SIEM, the problem can be remedied by having a single machine draw attention from the others and communicate with the SIEM to manage the security of the entire network. In a more realistic case, all machines connected to the network would have to send logs toward the SIEM, but here we deal with a simplified case. Thus, a design was obtained where the

honeypot machine acts as the flow and event collector, where QRadar is instead the flow and event processor, and then can display the normalized data on the dashboard.

6.3 Installation

Installation of the system mentioned above then begins with installation of the QRadar appliance on a dedicated machine, which uses a recent distribution of Ubuntu Desktop as its operating system. The Community Edition (CE) of QRadar is provided as an .ova file, which can be used by virtualization applications such as VMware Workstation or Oracle VirtualBox. We then proceed to create a virtual machine, importing the previously downloaded .ova file, via VirtualBox at the command line assuming that we are connected to the machine in question via SSH.

```
$ vboxmanage import QRadar.ova
0\%...10\%...20\%...30\%...40\%...50\%...60\%...70\%...80\%...90\%...100\%
Interpreting /home/user/Downloads/QRadar.ova...
OK.
Disks:
  vmdisk1 250 4362338304
http://www.vmware.com/interfaces/specifications/vmdk.html#streamOptimized
QCE-jan22-disk1.vmdk 3441993728 -1

Virtual system 0:
 0: Suggested OS type: "RedHat_64"
    (change with "--vsys 0 --ostype <type>"; use "list ostypes" to list all
possible values)
 1: Suggested VM name "vm 1"
    (change with "--vsys 0 --vmname <name>")
 2: Suggested VM group "/"
    (change with "--vsys 0 --group <group>")
 [...]
12: Hard disk image: source image=QCE-jan22-disk1.vmdk, target
path=QCE-jan22-disk1.vmdk, controller=10;port=0
    (change target path with "--vsys 0 --unit 12 --disk path";
    change controller with "--vsys 0 --unit 12 --controller <index>";
    change controller port with "--vsys 0 --unit 12 --port <n>";
    disable with "--vsys 0 --unit 12 --ignore")
0\%...10\%...20\%...30\%...40\%...50\%...60\%...70\%...80\%...90\%...100\%
Successfully imported the appliance.
```

Through the `vboxmanage list vms` command we can now see the virtual machine created:
`"vm" {e860195d-1c0e-44dc-99a3-0fa6105ec828}`

The assigned machine name is not important, so you can leave it unchanged. At this point it is necessary, if using a locally hosted virtual machine with a local IP address, to forward port 8444 to port 443 to access QRadar in the web browser and forward port 2222 to port 22 to use ssh to connect to the QRadar machine. However, this step is not necessary in case you were to use a bridged virtual machine networking setting, as in our case. With bridged type networking, the virtual machine is accessible from the network and it is sufficient to

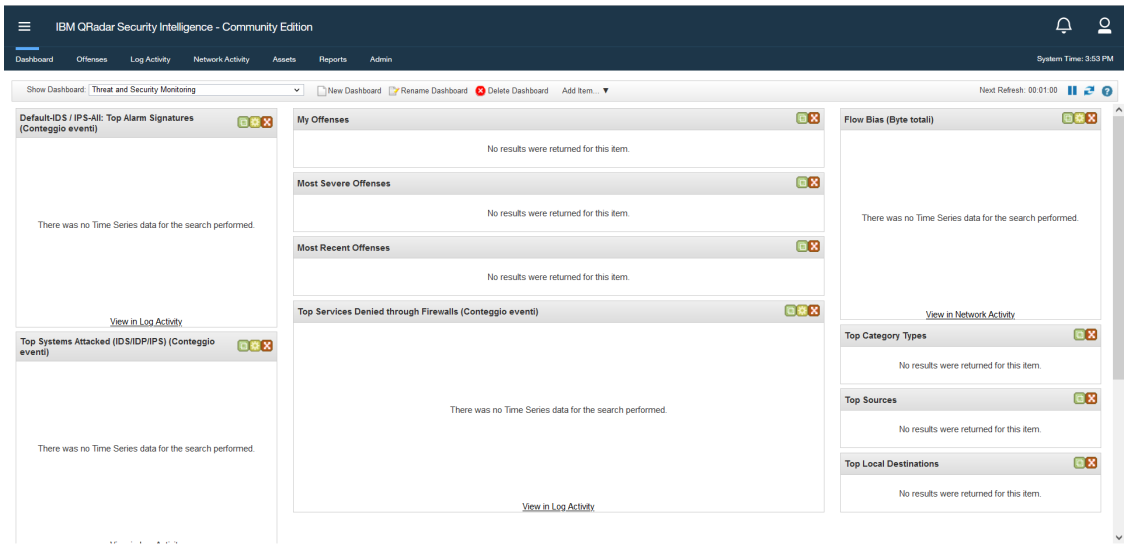
assign it a static IP via DHCP, so that at each reboot the IP address always remains the same. If the network is NAT type, on the other hand, it shares the host's network connection by assigning the virtual machines an IP address from a private network and translates the guest's network requests. In this way, the host appears as a single system to the network. Installation then proceeds by booting the virtual machine via:

```
vboxmanage startvm vm
```

By connecting to the latter, logging in as root, setting a password and finally then entering the command `./setup .` Once this step is completed simply accept the terms and conditions and the installation will begin its course. Once the installation is complete, the machine will start the necessary services and the dashboard via web interface will be available at the IP address of the host machine.

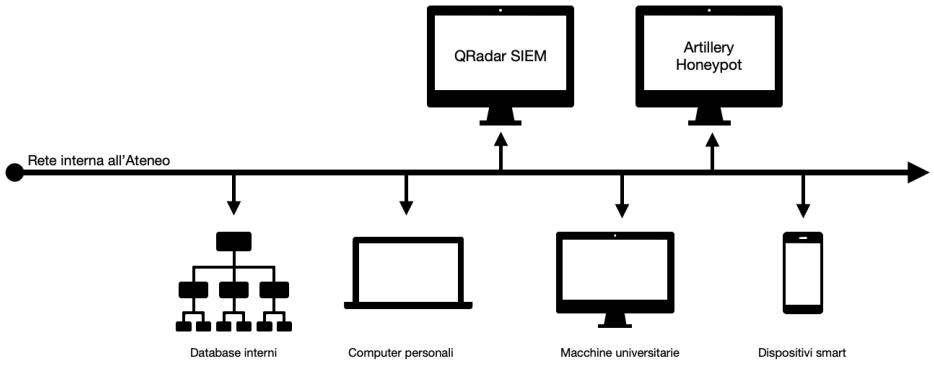
The QRadar login screen

Once logged in, through an admin account and no longer the root account which will remain unused for security reasons, we see a presentation of the QRadar web interface. We can immediately notice two important tabs from which we will view the data we are interested in: log activity and network activity. These two sections are dedicated respectively to events, those obtained from the logs of external machines sending information and internal to QRadar, and network flows, obtained instead from routers, switches, TAPs in the network, but also machines configured to send this type of information.



The QRadar Dashboard

Now let's move on to the installation of the honeypot: as mentioned earlier we use Artillery and its installation is very simple. Starting from the machine on which you want to install the application, we use git to clone the repository: `sudo git clone https://github.com/BinaryDefense/artillery` Next, we enter the directory created (cd artillery) and continue by writing `./setup`. This way Artillery will be installed in `/var/artillery` and all its features can be modified from the configuration file named "config" present in the installation directory. The result of our installations leads to a topographical map of the network of the type visible in the figure below.



Example Athenaeum network topography

For a general average-sized organization, the installation process turns out to be the same in terms of installing the QRadar SIEM appliance. Regarding the other components, such as firewalls, switches, servers, honeypots and in general any machine whose specific task is to communicate with the SIEM for one reason or another, they will have their own installation

to follow. Generally, each workstation is set up to send events to QRadar while switches and routers are configured to send network flows.

6.4 Configuration

The configuration of the QRadar system, the one required for its operation, is done automatically. Basically, to get access to QRadar SIEM and its web interface, there are no specific configuration steps to follow, other than the initial setup of the appliance. In this case, however, we mean the configuration part as the steps necessary to set up communication between the various machines. Before proceeding, however, it is necessary to mention a problem that plagues the current version of QRadar SIEM Community Edition and other full versions of the appliance: changes made to QRadar on December 31, 2020 can (and in fact do) impact the functionality of the product. The result is that even after properly configuring communication between the machines from which you want to extract data and QRadar itself, the web interface will not show any data. A patch has been released by IBM that allows the problem to be resolved. Simply be connected to the QRadar host machine via SSH and run one of the following commands, dependent on the version you are using. Once this is done, a simple restart of the web interface will solve the problem.

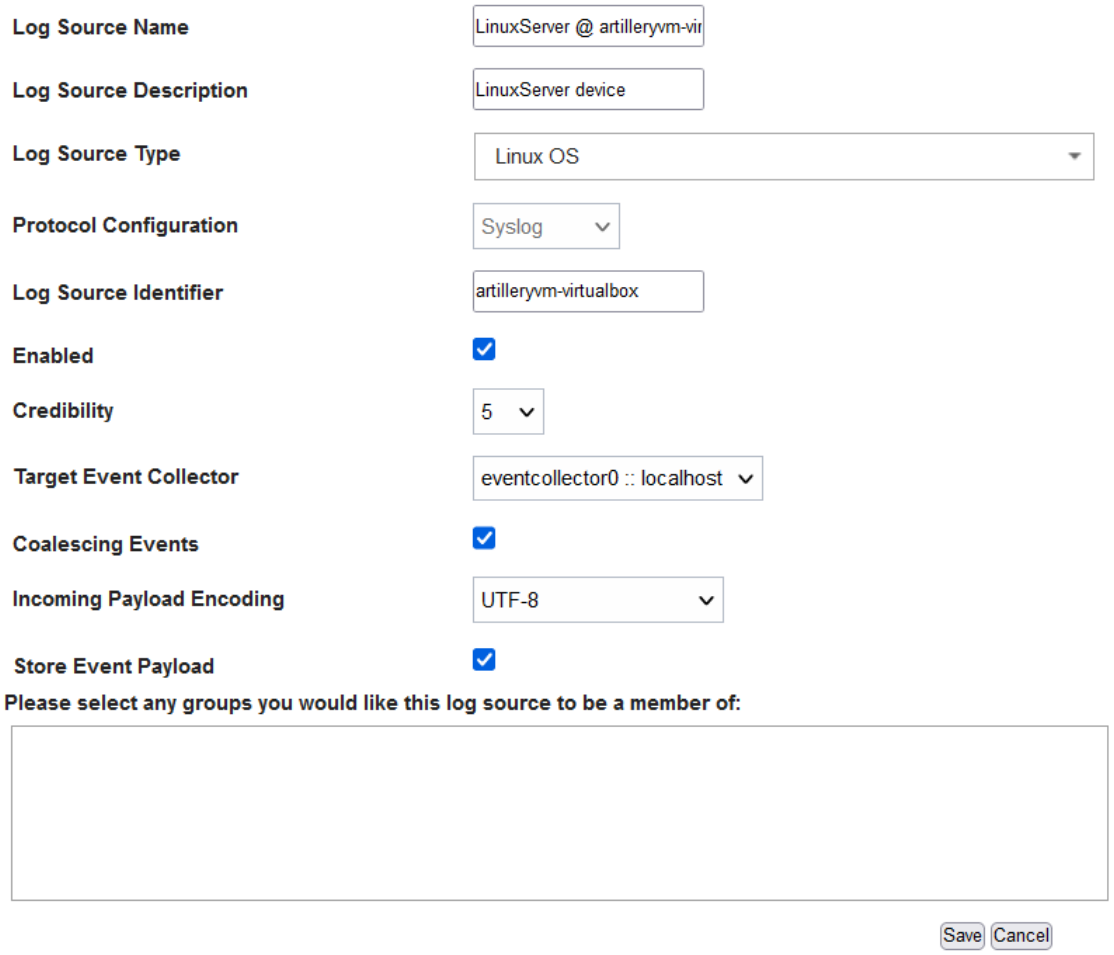
Let's look at the version for QRadar Community Edition:

```
\if [ -f /opt/qradar/ecs/license.txt ] ; then echo -n "QRadar:Q1 Labs
Inc.:0007634bda1e2:WnT9X7BDFOgB1WaXwokODc:12/31/20" >
/opt/qradar/ecs/license.txt ; fi ; if [ -f
/opt/ibm/si/services/ecs-ec-ingress/current/eventgnosis/license.txt ] ; then
echo -n "QRadar:Q1 Labs Inc.:0007634bda1e2:WnT9X7BDFOgB1WaXwokODc:12/31/20" >
/opt/ibm/si/services/ecs-ec-ingress/current/eventgnosis/license.txt ; fi ; fi
; if [ -f /opt/ibm/si/services/ecs-ep/current/eventgnosis/license.txt ] ;
then echo -n "QRadar:Q1 Labs
Inc.:0007634bda1e2:WnT9X7BDFOgB1WaXwokODc:12/31/20" >
/opt/ibm/si/services/ecs-ep/current/eventgnosis/license.txt ; fi ; if [ -f
/opt/ibm/si/services/ecs-ec/current/eventgnosis/license.txt ] ; then echo -n
"QRadar:Q1 Labs Inc.:0007634bda1e2:WnT9X7BDFOgB1WaXwokODc:12/31/20" >
/opt/ibm/si/services/ecs-ec/current/eventgnosis/license.txt ; fi ; fi ; if [
-f /usr/eventgnosis/ecs/license.txt ] ; then echo -n "QRadar:Q1 Labs
Inc.:0007634bda1e2:WnT9X7BDFOgB1WaXwokODc:12/31/20" >
/usr/eventgnosis/ecs/license.txt ; fi ; fi ; if [ -f
/opt/qradar/conf/templates/ecs_license.txt ] ; then echo -n "QRadar:Q1 Labs
Inc.:0007634bda1e2:WnT9X7BDFOgB1WaXwokODc:12/31/20" >
/opt/qradar/conf/templates/ecs_license.txt ; fi ; fi
```

It is now possible to continue with the configuration of the machines so that they establish a connection between them. This means configuring QRadar so that it can receive events and streams from the other machines, or in our case Artillery, and the other machines so that they can send the same events and streams to the QRadar appliance.

6.4.1 Configuring QRadar and Artillery for Events

Configuring the two applications to communicate with each other requires two steps, one for each of the two appliances. First you need to enable QRadar to receive the above events, and to do that, you need to add a source for this data i.e., a log source. This means heading to the admin panel, then data sources, log sources and click add to add a new log source.



Adding a log source

We must first provide a name for the source, then provide the IP address from which we expect to receive the content. We also need to select the type of source, which in this case uses the syslog protocol (RSyslog to be precise), so it will be a Linux OS type. Once this step is completed, back in the Admin tab, we simply click on deploy changes and wait for the changes to take effect. QRadar is now properly configured to receive events (or logs). It is important to note that this process is to be repeated for each different log source that you want to configure.

Regarding the configuration of Syslog and Artillery for sending log files, we want the machine to forward both Artillery and machine logs to QRadar. To do this, we will have the Artillery logs piped to the system logs, which in turn will be forwarded to SIEM via the

RSyslog protocol. We then need to connect via SSH to the application host machine and then check that RSyslog, the chosen communication protocol, is present on the system. To do this we can type:

```
$ systemctl status rsyslog
```

```
 rsyslog.service - LSB: enhanced syslogd
     Loaded: loaded (/etc/init.d/rsyslog; generated)
     Active: active (exited) since Tue 2023-02-14 11:00:35 CET; 26min ago
       Docs: man:systemd-sysv-generator(8)
    Process: 1785 ExecStart=/etc/init.d/rsyslog start (code=exited,
status=0/SUCCESS)

Feb 14 11:00:35 ***** systemd[1]: Starting LSB: enhanced syslogd...
Feb 14 11:00:35 ***** systemd[1]: Starting LSB: enhanced syslogd.
```

Once we have verified the presence of the protocol, we need to edit the file containing the syslog configuration, using:

```
sudo nano /etc/rsyslog.conf
```

We add the following lines to the bottom of the text file:

```
#####QRADAR CONNECTION#####
module(load="imtcp")
input(type="imtcp" port="514")
#module(load="imudp")
#input(type="imdup" port="514")
*.*@@192.168.50.252:514 #connection type TCP
#*.*@192.168.50.252:514 #Connection type UDP
```

This change in the protocol configuration file will cause the system logs to be redirected to the QRadar appliance, which has been configured to receive the above events on port 514. Finally, it is necessary to access the Artillery configuration file (found in /var/artillery) in order to pipe the application logs back to the localhost direction so that they are then sent to SIEM. The last step is to perform a restart of the syslog communication protocol via console, systemctl restart rsyslog, and to start the Artillery service via ./restart-server. To check for proper honeypot functionality we can type

```
$ sudo netstat -nlp | grep python
```

and we should see that different Python processes are listening on different ports:

```
tcp 0 0.0.0.0:5900 0.0.0.0:* LISTEN 9020/python3
tcp 0 0.0.0.0:110 0.0.0.0:* LISTEN 9020/python3
tcp 0 0.0.0.0:10000 0.0.0.0:* LISTEN 9020/python3
tcp 0 0.0.0.0:8080 0.0.0.0:* LISTEN 9020/python3
tcp 0 0.0.0.0:21 0.0.0.0:* LISTEN 9020/python3
tcp 0 0.0.0.0:1433 0.0.0.0:* LISTEN 9020/python3
tcp 0 0.0.0.0:1337 0.0.0.0:* LISTEN 9020/python3
tcp 0 0.0.0.0:25 0.0.0.0:* LISTEN 9020/python3
```

```
tcp 0 0.0.0.0:44443 0.0.0.0:* LISTEN 9020/python3
tcp 0 0.0.0.0:1723 0.0.0.0:* LISTEN 9020/python3
tcp 0 0.0.0.0:445 0.0.0.0:* LISTEN 9020/python3
tcp 0 0.0.0.0:3389 0.0.0.0:* LISTEN 9020/python3
tcp 0 0.0.0.0:135 0.0.0.0:* LISTEN 9020/python3
tcp 0 0.0.0.0:5800 0.0.0.0:* LISTEN 9020/python3
```

If all has gone well, accessing the QRadar web interface now and clicking on the "Log Activity" tab, a series of events should now appear both internal to the machine and from the Artillery host machine.

Event Name	Log Source	Even Coun	Time	Low Level Category	Source IP	Source Port	Destination IP
Linux login messages Message	Artillery	1	Feb 14, 2023, 12:44:...	Stored	192.168.50.16	0	192.168.50.16
Linux login messages Message	Artillery	1	Feb 14, 2023, 12:44:...	Stored	192.168.50.16	0	192.168.50.16
Linux login messages Message	Artillery	1	Feb 14, 2023, 12:44:...	Stored	192.168.50.16	0	192.168.50.16
Linux login messages Message	Artillery	1	Feb 14, 2023, 12:44:...	Stored	192.168.50.16	0	192.168.50.16
Linux login messages Message	Artillery	1	Feb 14, 2023, 12:44:...	Stored	192.168.50.16	0	192.168.50.16
Unknown log event	SIM Generic Log DSM-7 :: local...	1	Feb 14, 2023, 12:44:...	Unknown Generic Log Event	192.168.50.1	0	192.168.50.1
Unknown log event	SIM Generic Log DSM-7 :: local...	1	Feb 14, 2023, 12:44:...	Unknown Generic Log Event	192.168.50.1	0	192.168.50.1
Information Message	System Notification-2 :: localhost	1	Feb 14, 2023, 12:44:...	Information	127.0.0.1	0	127.0.0.1
Information Message	System Notification-2 :: localhost	1	Feb 14, 2023, 12:44:...	Information	127.0.0.1	0	127.0.0.1
Information Message	System Notification-2 :: localhost	1	Feb 14, 2023, 12:44:...	Information	127.0.0.1	0	127.0.0.1
Information Message	System Notification-2 :: localhost	1	Feb 14, 2023, 12:44:...	Information	127.0.0.1	0	127.0.0.1
Information Message	System Notification-2 :: localhost	1	Feb 14, 2023, 12:44:...	Information	127.0.0.1	0	127.0.0.1
Information Message	System Notification-2 :: localhost	1	Feb 14, 2023, 12:44:...	Information	127.0.0.1	0	127.0.0.1
Information Message	System Notification-2 :: localhost	1	Feb 14, 2023, 12:44:...	Information	127.0.0.1	0	127.0.0.1
Information Message	System Notification-2 :: localhost	1	Feb 14, 2023, 12:44:...	Information	127.0.0.1	0	127.0.0.1
Information Message	System Notification-2 :: localhost	1	Feb 14, 2023, 12:44:...	Information	127.0.0.1	0	127.0.0.1

Activity log sheet

Another way to test that the two machines are communicating is to use the logger command, to generate logs that will then be sent from RSyslog to the QRadar machine. You must first, however, put the QRadar machine on listen, from the console, using tcpdump. The commands will then be:

On the QRadar host machine: `sudo tcpdump -i enp0s17 host 192.168.50.16`

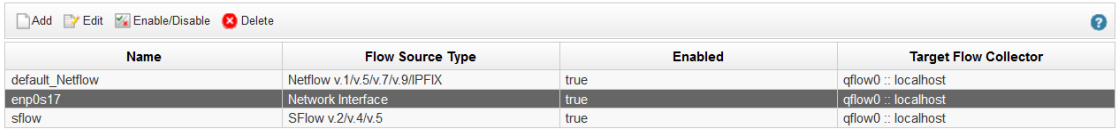
On the artillery machine: `logger -n 192.168.50.252 -P 514 Test`

In output from the QRadar console, when we run the command several times, we see that messages are being received, and this confirms that the two machines are communicating.

```
E.....@.@.FS..2...2.........<13>1 2023-02-15T12:02:10.640449+01:00
*************** - - [timeQuality tzKnown="1" isSynced="1"
syncAccuracy="421500"] Test
12:02:09.578836 IP (tos 0x0, ttl 64, id 19298, offset 0, flags [DF], proto
UDP (17), length 163)
    ********.54669 > localhost.localdomain.514: SYSLOG, length: 135
        Facility user (1), Severity notice (5)
        Msg: 1 2023-02-15T12:02:10.961475+01:00 ******** ****- - [timeQuality
tzKnown="1" isSynced="1" syncAccuracy="421500"] Test
E...Kb@.@.....2...2......../<13>1 2023-02-15T12:02:10.961475+01:00
************ - - [timeQuality tzKnown="1" isSynced="1" syncAccuracy="421500"]
Test
```

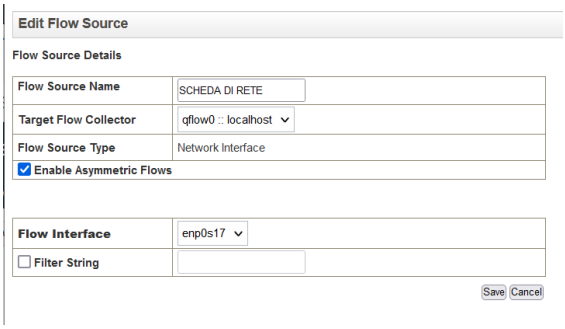
6.4.2 Configuration of network flows

In addition to configuring events we know that it is also possible to configure network flows so that connections are monitored by QRadar and appear in the "Network Activity" section. As seen above there are several methods to collect these types of flows, from the ability to use protocols such as SFlow or JFlow, to being able to monitor connections directly from the network card used by the appliance. Let's look at how to populate the network activity using the network card: you must first again head to the Admin panel and click on Flows, and then Flows Sources. A new tab will open where you can configure the source for the network flows.



Name	Flow Source Type	Enabled	Target Flow Collector
default_Netflow	Netflow v.1/v.5/v.7/v.9/IPFIX	true	qflow0 :: localhost
enp0s17	Network Interface	true	qflow0 :: localhost
sflow	SFlow v.2/v.4/v.5	true	qflow0 :: localhost

Addition of a flow source



Using the network card for monitoring

Once this step is completed, simply, as we did before, use the deploy command to make the changes, and once completed, we can head to the network activity page, which will now be populated with all the information that our network card detects. It is possible to add several sources, especially if there is a need to monitor several subnets, but to avoid redundancy we have avoided this route.

Flow Type	First Packet Time	Source IP	Source Port	Destination IP	Destination Port	Protocol	Application	Source Bytes	Destination Bytes	Source Packets	Destination Packets	ICMP Type/Code	Flow Source
	Feb 14, 20...	192.168.50.252	39414	192.168.50.1	53	udp_ip	Misc.domain	136 (C)	159 (C)	1	1	N/A	localhost
	Feb 14, 20...	192.168.50.252	38278	192.168.50.1	53	udp_ip	Misc.domain	136 (C)	159 (C)	1	1	N/A	localhost
	Feb 14, 20...	192.168.50.252	34207	192.168.50.1	53	udp_ip	Misc.domain	90 (C)	140 (C)	1	1	N/A	localhost
	Feb 14, 20...	192.168.50.252	55945	192.168.50.1	53	udp_ip	Misc.domain	85 (C)	162 (C)	1	1	N/A	localhost
	Feb 14, 20...	192.168.50.1	N/A	224.0.0.1	N/A	igmp	Other	192 (C)	0	3	0	N/A	localhost
	Feb 14, 20...	192.168.50.252	59376	192.168.50.1	53	udp_ip	Misc.domain	85 (C)	162 (C)	1	1	N/A	localhost
	Feb 14, 20...	192.168.50.16	60926	192.168.50.252	514	udp_ip	Misc.Syslog	115 (C)	0	1	0	N/A	localhost
	Feb 14, 20...	192.168.50.252	54375	192.168.50.1	53	udp_ip	Misc.domain	136 (C)	159 (C)	1	1	N/A	localhost
	Feb 14, 20...	192.168.50.252	41290	192.168.50.1	53	udp_ip	Misc.domain	85 (C)	162 (C)	1	1	N/A	localhost
	Feb 14, 20...	192.168.50.16	45263	192.168.50.252	514	udp_ip	Misc.Syslog	106 (C)	0	1	0	N/A	localhost
	Feb 14, 20...	192.168.50.252	54022	192.168.50.1	53	udp_ip	Misc.domain	136 (C)	159 (C)	1	1	N/A	localhost
	Feb 14, 20...	192.168.50.252	40498	192.168.50.1	53	udp_ip	Misc.domain	136 (C)	159 (C)	1	1	N/A	localhost
	Feb 14, 20...	192.168.50.252	38649	192.168.50.1	53	udp_ip	Misc.domain	136 (C)	159 (C)	1	1	N/A	localhost
	Feb 14, 20...	192.168.50.16	48337	192.168.50.252	514	udp_ip	Misc.Syslog	132 (C)	0	1	0	N/A	localhost
	Feb 14, 20...	192.168.50.252	41906	192.168.50.1	53	udp_ip	Misc.domain	136 (C)	159 (C)	1	1	N/A	localhost
	Feb 14, 20...	192.168.50.252	42255	192.168.50.1	53	udp_ip	Misc.domain	136 (C)	159 (C)	1	1	N/A	localhost
	Feb 14, 20...	192.168.50.252	56223	192.168.50.1	53	udp_ip	Misc.domain	136 (C)	159 (C)	1	1	N/A	localhost
	Feb 14, 20...	192.168.50.252	41745	192.168.50.1	53	udp_ip	Misc.domain	85 (C)	162 (C)	1	1	N/A	localhost

Network activity tab

6.5 Maintenance

Let us now look at the responsibility part for common maintenance and administration activities. Most maintenance or custom administration scenarios are defined as an activity that can be performed in the user interface or any customization that is not covered in the IBM Documentation for QRadar. If you need to perform a configuration or process that is not described in the documentation, it is probably a custom configuration and not a product issue, such as adding cron jobs or editing files. Most maintenance work related to QRadar, however, is limited to taking the configurations we saw earlier and applying them for different services. This is mainly done in case you want to expand your enterprise network and thus add machines, routers, TAP devices, or others, so that they too can communicate with QRadar and provide useful information to analysts.

Let's look at a list of maintenance work that can occur on different occasions when you are maintaining a complex SIEM such as QRadar:

- Removal of event data or streams other than those configured in the preservation buckets.
- Administrators asking QRadar Support to open ports, update iptables, or add cron jobs to appliances.
- Case requests for modifying files or implementing workarounds for features not included in the IBM Documentation.
- Modification of the user interface.
- Creation of custom scripts to perform tasks for administrators.
- Requests to update or manage the implementation of QRadar, such as:
- Cleaning unused log sources.
- Adding and configuring managed hosts for administrators when no errors are reported.
- Review, cleaning or maintenance of reference set data.
- Requests to monitor disk space, delete data, or move data between QRadar appliances.
- Performing network hierarchy updates for administrators.
- Requests to test working log sources for audit or data validation purposes.
- Creating static paths.
- Maintain or update IP addresses or complete DNS changes for administrators.
- Maintenance of rules.

- Custom Event Properties (CEP) updates or maintenance.
- Scheduling, editing reports, or updating report layout for administrators.
- Removing WinCollect agents or updating WinCollect configuration parameters in the user interface.
- Troubleshooting implemented by third-party vendors or implemented by Security Experts Labs.

An example of maintenance that we saw in the Athenaeum internship was the creation of a rule that attributes the output of a given Artillery log, specifically where it is written in the output that an attack on a range of ports has been detected, to an offense, that is, a risk of compromise to the system.

6.6 Experimentation and attacks

For the experimentation part, after performing all the steps seen above, it was chosen to monitor the network to check its normal and secure situation. The result was satisfactory in that the university network was found to be secure and free of any kind of attack over a period of a few months. Therefore, it was chosen to simulate several attacks on the installed honeypot and to see how the presence of the aforementioned can be detected by QRadar and then act accordingly and mitigate the attack. These attacks are intended to simulate the three basic steps that a hacker would follow in the case of a real attack:

- Reconnaissance
- Machine penetration
- Editing internal system files

The idea was to take a machine internal to the network and therefore free of suspicion, and then run a full scan (hosts, ports, services) of the honeypot from it. This procedure is intended to simulate the reconnaissance phase that an attacker would perform in preparation for a possible attack on the university's infrastructure. It was chosen to use a machine internal to the university network because of the latter's protection by a VPN. To perform the attack, a tool commonly used by hackers for reconnaissance called NMAP was chosen, which we will see in a moment in detail.

A second type of attack that was intended to be tested against the honeypot is that of a connection via SSH from the "infected" machine. After applying the good practice of changing the honeypot's SSH service port from the classic port 22 to something more secure, such as 22117, a connection was attempted on the monitored port.Artillery monitors several ports and connection types, including ports 22 and 21 for SSH connections. This second type of attack is intended to see and understand the reaction of the system in case a potential hacker, having done the reconnaissance and seen that the machine accepts SSH connections, tries to connect perhaps hoping that the password is simple and to attempt a potential brute-force, which could grant him access.

The third type of attack we wanted to test is the modification of files internal to the honeypot machine in case it is penetrated. This corresponds to the step of tampering with the system's internal files once penetration has occurred, such as adding a backdoor inside the machine. Artillery monitors certain internal directories, which can be changed at will, so that the analyst can be notified if any unpermitted changes occur. Let us next look at the attacks in detail.

6.6.1 NMAP

Nmap ("Network Mapper") is an open-source tool for network exploration and auditing. It is designed to quickly scan large networks, but is also suitable for use toward individual hosts. Nmap uses "raw" (raw, unformatted) IP packets in various ways to determine which hosts are available on a network, what services (application name and version) are offered by these hosts, what operating system (and what version of the operating system) is running, what type of firewalls and packet filters are being used, and many other features. Although Nmap is commonly used for security audits, many systems engineers and network administrators find it useful for all day-to-day tasks such as inventorying machines on the network, managing scheduled service updates, and monitoring hosts or their uptime. In addition to a table of interesting ports, Nmap can provide additional target information such as resolved DNS names (reverse DNS names), likely operating system in use, device type, and physical address (MAC address).

6.6.2 Results, detection, countermeasures

The first type of attack we want to test is the one we have associated with the reconnaissance phase. In this phase, an attacker performs several scans to try to figure out what certain vulnerabilities a particular machine may have. One commonly used tool is the one we saw above, which is NMAP. Scanning using NMAP is done in a simple way. Once the tool is installed, simply run the command with the parameters you want to use. In this case:

```
sudo apt-get install NMAP sudo NMAP -sU -V 192.168.50.251
```

The result of the scan appears to us soon after:

```
Completed SYN Stealth Scan at 15:22, 0.11s elapsed (1000 total ports)
Nmap scan report for artilleryvm-virtualbox (192.168.50.251).
Host is up (0.00062s latency).
Not shown: 986 closed ports
PORT STATE SERVICE
21/tcp open ftp
22/tcp open ssh
25/tcp open smtp
110/tcp open pop3
1433/tcp open ms-sql-s
1723/tcp open pptp
5060/tcp open sip
5061/tcp open sip-tls
5800/tcp open vnc-http
5900/tcp open vnc
```

```
8080/tcp open http-proxy
10000/tcp open snet-sensor-mgmt
16993/tcp open amt-soap-https
44443/tcp open coldfusion-auth
MAC Address: ************* (Oracle VirtualBox virtual NIC)
Read data files from: /usr/bin/../share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.35 seconds
           Raw packets sent: 1001 (44.028KB) | Rcvd: 1001 (40.084KB)
```

In addition to the result, however, a suspicious event immediately appears on the QRadar console, which we see is sent from the Artillery host machine.

Event Name	Log Source	Even Coun	Time	Low Level Category	Source IP	Source Port	Destination IP	Destina Port	Username	Magnitude
Health Metric	Health Metrics-2 :: localhost	1	Feb 21, 2023, 2:55:01 PM	Information	127.0.0.1	0	127.0.0.1	0	N/A	
Health Metric	Health Metrics-2 :: localhost	1	Feb 21, 2023, 2:55:01 PM	Information	127.0.0.1	0	127.0.0.1	0	N/A	
Health Metric	Health Metrics-2 :: localhost	1	Feb 21, 2023, 2:55:01 PM	Information	127.0.0.1	0	127.0.0.1	0	N/A	
Health Metric	Health Metrics-2 :: localhost	1	Feb 21, 2023, 2:55:01 PM	Information	127.0.0.1	0	127.0.0.1	0	N/A	
Health Metric	Health Metrics-2 :: localhost	1	Feb 21, 2023, 2:55:01 PM	Information	127.0.0.1	0	127.0.0.1	0	N/A	
Unknown log event	SIM Generic Log DSM-7 :: local...	1	Feb 21, 2023, 2:55:01 PM	Unknown Generic Log Event	192.168.50.1	0	192.168.50.1	0	N/A	
Linux login messages Message	LinuxServer @ artilleryvm-virtu...	1	Feb 21, 2023, 2:55:01 PM	Stored	192.168.50.251	0	192.168.50.251	0	N/A	
Linux login messages Message	LinuxServer @ artilleryvm-virtu...	1	Feb 21, 2023, 2:55:01 PM	Stored	192.168.50.251	0	192.168.50.251	0	N/A	
Linux login messages Message	LinuxServer @ artilleryvm-virtu...	1	Feb 21, 2023, 2:55:01 PM	Stored	192.168.50.251	0	192.168.50.251	0	N/A	

Suspicious event detected by Artillery

By highlighting the message and opening the related tab we can see how it is highlighted that the honeypot has detected a connection:

Event Description	Linux login messages Stored Event						
Magnitude	(7)	Relevance	10	Severity	3	Credibility	10
Username	N/A						
Start Time	Feb 21, 2023, 2:55:01 PM	Storage Time	Feb 21, 2023, 2:55:01 PM	Log Source Time	Feb 21, 2023, 2:55:01 PM		
Domain	Default Domain						

Source and Destination Information

Source IP	192.168.50.251	Destination IP	192.168.50.251
Source Asset Name	192.168.50.251	Destination Asset Name	192.168.50.251
Source Port	0	Destination Port	0
Pre NAT Source IP		Pre NAT Destination IP	
Pre NAT Source Port	0	Pre NAT Destination Port	0
Post NAT Source IP		Post NAT Destination IP	
Post NAT Source Port	0	Post NAT Destination Port	0
Source IPv6	0:0:0:0:0:0:0:0	Destination IPv6	0:0:0:0:0:0:0:0
Source MAC	00:00:00:00:00:00	Destination MAC	00:00:00:00:00:00

Payload Information

utf | hex | base64

Wrap Text

```
<10>Feb 21 14:55:01 artilleryvm-virtualbox Artillery[WARN]: message repeated 62 times: [ Artillery has detected an attack from 192.168.50.16 for a connection on a honeypot port 5060]
```

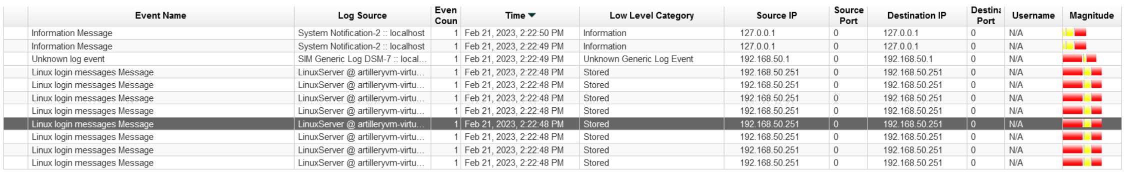
Details of the suspicious event

It is easy then to see that the scan, despite not being a real connection attempt, was detected immediately by the honeypot which then sent the logs toward QRadar so that the analyst could view them. All of these alerts, if desired, can be turned via rules into offenses so that it is quick and easy to detect "real" problems from all of the common logs you may receive.

The second type of attack we discussed earlier is that of an attempt to connect via SSH to the honeypot machine by the hacker, who after reconnaissance notices that port 21 and 22 are conventionally open. By simply entering the IP address and port, the hacker in question then tries to connect to the machine by executing: `sudo ssh 192.168.50.251 -p 21` But the connection is blocked:

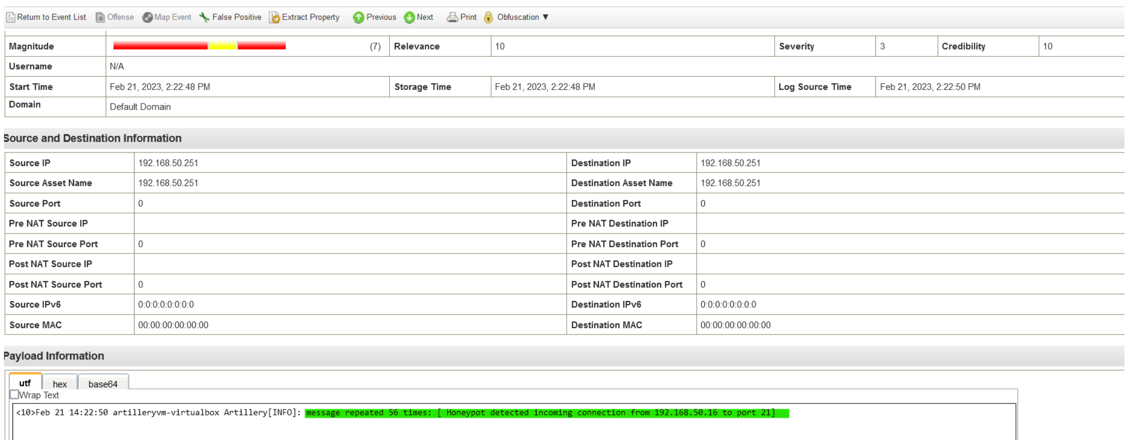
```
|kex_exchange_identification: banner line contains invalid characters
```

It is possible to see in the Log Activity section of the QRadar web interface that logs with high magnitude related to the attempted access have popped up. Opening one of the generated logs we get a description of the attack that tells us how Artillery detected an attempted connection to the IP address and port.



Event Name	Log Source	Even Coun	Time	Low Level Category	Source IP	Source Port	Destination IP	Destina Port	Username	Magnitude
Information Message	System Notification-2 :: localhost	1	Feb 21, 2023, 2:22:50 PM	Information	127.0.0.1	0	127.0.0.1	0	N/A	
Information Message	System Notification-2 :: localhost	1	Feb 21, 2023, 2:22:49 PM	Information	127.0.0.1	0	127.0.0.1	0	N/A	
Unknown log event	SIM Generic Log DSM-7 :: local...	1	Feb 21, 2023, 2:22:49 PM	Unknown Generic Log Event	192.168.50.1	0	192.168.50.1	0	N/A	
Linux login messages Message	LinuxServer @ artilleryvm-virtu...	1	Feb 21, 2023, 2:22:48 PM	Stored	192.168.50.251	0	192.168.50.251	0	N/A	
Linux login messages Message	LinuxServer @ artilleryvm-virtu...	1	Feb 21, 2023, 2:22:48 PM	Stored	192.168.50.251	0	192.168.50.251	0	N/A	
Linux login messages Message	LinuxServer @ artilleryvm-virtu...	1	Feb 21, 2023, 2:22:48 PM	Stored	192.168.50.251	0	192.168.50.251	0	N/A	
Linux login messages Message	LinuxServer @ artilleryvm-virtu...	1	Feb 21, 2023, 2:22:48 PM	Stored	192.168.50.251	0	192.168.50.251	0	N/A	
Linux login messages Message	LinuxServer @ artilleryvm-virtu...	1	Feb 21, 2023, 2:22:48 PM	Stored	192.168.50.251	0	192.168.50.251	0	N/A	
Linux login messages Message	LinuxServer @ artilleryvm-virtu...	1	Feb 21, 2023, 2:22:48 PM	Stored	192.168.50.251	0	192.168.50.251	0	N/A	
Linux login messages Message	LinuxServer @ artilleryvm-virtu...	1	Feb 21, 2023, 2:22:48 PM	Stored	192.168.50.251	0	192.168.50.251	0	N/A	
Linux login messages Message	LinuxServer @ artilleryvm-virtu...	1	Feb 21, 2023, 2:22:48 PM	Stored	192.168.50.251	0	192.168.50.251	0	N/A	

Event for connection attempt

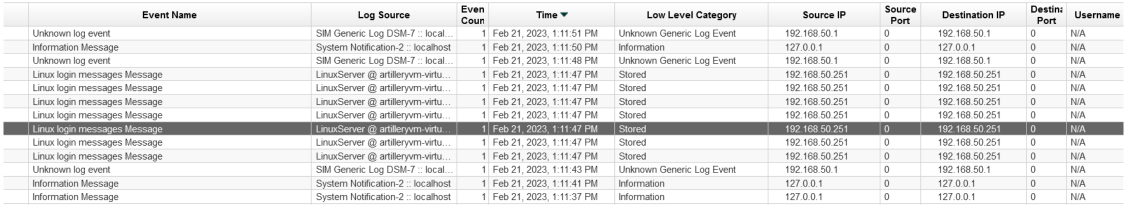


Connection attempt details

The third type of attack we have seen is the modification of system files. In this case we assume that the hacker has managed to connect to the machine via SSH and now wants to install a backdoor into some directory. After setting up Artillery to monitor the /var/www/ directory, to simulate the attack they simply created a file inside the directory using the touch command:

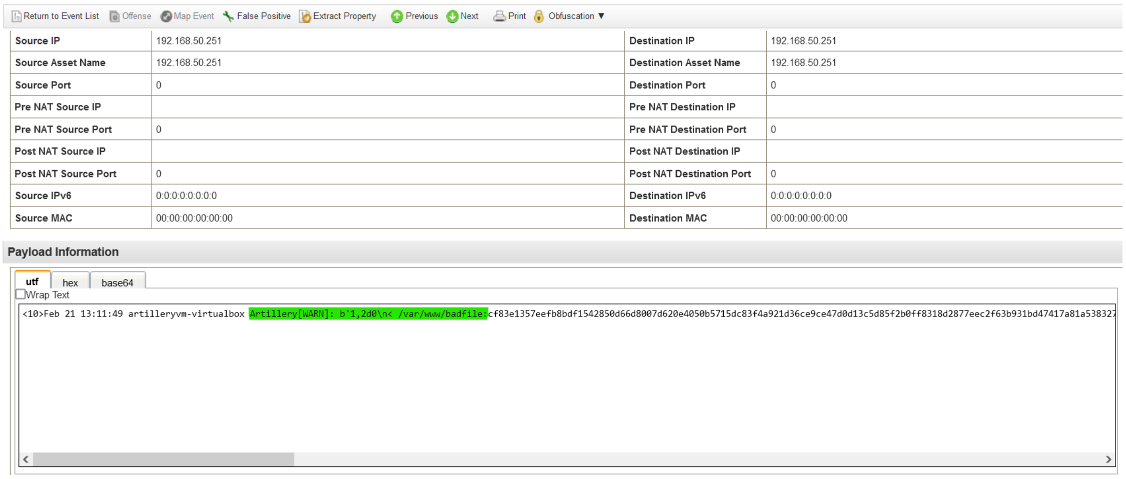
```
sudo touch bad_file
```

After a few moments you can see from the QRadar web interface in the Log Activity section the problem. By double-clicking on the event we can see in detail what happened: we see how Artillery sends a warning and tells us that inside `/var/www` a change has been noticed and we are also specified the name of the created file, in this case `bad_file`.



	Event Name	Log Source	Even Coun	Time	Low Level Category	Source IP	Source Port	Destination IP	Destina Port	Username
	Unknown log event	SIM Generic Log DSM-7 :: local...	1	Feb 21, 2023, 1:11:51 PM	Unknown Generic Log Event	192.168.50.1	0	192.168.50.1	0	N/A
	Information Message	System Notification-2 :: localhost	1	Feb 21, 2023, 1:11:50 PM	Information	127.0.0.1	0	127.0.0.1	0	N/A
	Unknown log event	SIM Generic Log DSM-7 :: local...	1	Feb 21, 2023, 1:11:48 PM	Unknown Generic Log Event	192.168.50.1	0	192.168.50.1	0	N/A
	Linux login messages Message	LinuxServer @ artilleryvm-virtu...	1	Feb 21, 2023, 1:11:47 PM	Stored	192.168.50.251	0	192.168.50.251	0	N/A
	Linux login messages Message	LinuxServer @ artilleryvm-virtu...	1	Feb 21, 2023, 1:11:47 PM	Stored	192.168.50.251	0	192.168.50.251	0	N/A
	Linux login messages Message	LinuxServer @ artilleryvm-virtu...	1	Feb 21, 2023, 1:11:47 PM	Stored	192.168.50.251	0	192.168.50.251	0	N/A
	Linux login messages Message	LinuxServer @ artilleryvm-virtu...	1	Feb 21, 2023, 1:11:47 PM	Stored	192.168.50.251	0	192.168.50.251	0	N/A
	Linux login messages Message	LinuxServer @ artilleryvm-virtu...	1	Feb 21, 2023, 1:11:47 PM	Stored	192.168.50.251	0	192.168.50.251	0	N/A
	Linux login messages Message	LinuxServer @ artilleryvm-virtu...	1	Feb 21, 2023, 1:11:47 PM	Stored	192.168.50.251	0	192.168.50.251	0	N/A
	Linux login messages Message	LinuxServer @ artilleryvm-virtu...	1	Feb 21, 2023, 1:11:47 PM	Stored	192.168.50.251	0	192.168.50.251	0	N/A
	Unknown log event	SIM Generic Log DSM-7 :: local...	1	Feb 21, 2023, 1:11:43 PM	Unknown Generic Log Event	192.168.50.1	0	192.168.50.1	0	N/A
	Information Message	System Notification-2 :: localhost	1	Feb 21, 2023, 1:11:41 PM	Information	127.0.0.1	0	127.0.0.1	0	N/A
	Information Message	System Notification-2 :: localhost	1	Feb 21, 2023, 1:11:37 PM	Information	127.0.0.1	0	127.0.0.1	0	N/A

Event for observed directory change



Source IP	192.168.50.251	Destination IP	192.168.50.251
Source Asset Name	192.168.50.251	Destination Asset Name	192.168.50.251
Source Port	0	Destination Port	0
Pre NAT Source IP		Pre NAT Destination IP	
Pre NAT Source Port	0	Pre NAT Destination Port	0
Post NAT Source IP		Post NAT Destination IP	
Post NAT Source Port	0	Post NAT Destination Port	0
Source IPv6	0:0:0:0:0:0:0:0	Destination IPv6	0:0:0:0:0:0:0:0
Source MAC	00:00:00:00:00:00	Destination MAC	00:00:00:00:00:00

Details of changes made

Conclusion

During the testing phase we saw how a good system configured to defend a network can detect and block an intrusion attempt fully automatically. In the first test we immediately noticed the reconnaissance performed by the hacker and, if we wanted, we could have configured Artillery to apply an immediate ban on this kind of offense. This would have made the system even more secure, but for simulation purposes we decided against it. After receiving the notification, the security measures an analyst could have taken are different: having the hacker run a scan, one could do the same thing to verify the results and fortify the system at the weaknesses found, if necessary. In the second test, we instead blocked an incoming SSH connection and notified the attempt. An immediate ban could have been triggered here as well, and care should be taken to check that all good security practices for SSH connections are in place. In a network protected by other overlying layers of security (such as a VPN) one would instead have to take care to understand where the connection came from and how it was possible for the hacker to attempt the connection. Continuing backtracking style, one would look for flaws and proceed to apply fixes if necessary. Regarding the third attack, where the situation falls into incident responding, the steps to follow are as we saw in the text in the dedicated section (4.3.3). In general, however, you now have a notification of an intrusion, and thanks to Artillery, the perpetrator can be stopped immediately. In this text we have looked at malware, social engineering and their attack vectors. We have talked extensively about information security, its process and how

to keep safe but also how hackers exploit all the tools at their disposal to compromise machines and networks, whether these are private, corporate or governmental. It is extremely difficult to keep up with the evolution of technologies, and this shows us how it is even more difficult to stay up-to-date and ready to protect systems that are constantly growing and a source of data that must remain protected, away from the hands and eyes of those who would exploit them for malicious purposes. Although security measures are constantly evolving, the battle to attack and defend information is likely to continue, probably hand in hand, for decades to come. What we have been able to demonstrate during our testing phase is how SIEMs simplify security management for organizations by filtering huge amounts of security data and prioritizing software-generated security alerts. SIEM software enables organizations to detect incidents that might otherwise go unnoticed, such as simple system scans or connection attempts, as we saw in the previous chapter. In addition to all the defensive systems and methods that have been mentioned in this text, it is now clear that SIEM systems offer something that a security team absolutely needs in order to process a huge amount of data and make it actionable.

If you would like to see the bibliography as well, please refer to the Italian version.
There you will be able to find all the citations and the complete bibliography for this text.

Thank you for taking the time to read my dissertation.