ONTOLOGY-BASED MODEL FOR SECURITY ASSESSMENT: PREDICTING CYBERATTACKS THROUGH THREAT ACTIVITY ANALYSIS

Pavel Yermalovich and Mohamed Mejri

Faculté des Sciences et de Génie, Université Laval, Québec City, Canada

ABSTRACT

The prediction of attacks is essential for the prevention of potential risk. Therefore, risk forecasting contributes a lot to the optimization of the information security budget. This article focuses on the ontology and stages of a cyberattack. It introduces the main representatives of the attacking side and describes their motivation.

KEYWORDS

Cyberattack, cyberattack prediction, ontology, cyberattack ontology, information security, cybersecurity, IT security, data security, threat activity.

1. INTRODUCTION

The use of information is inextricably linked with its security [1] which is founded on confidentiality, integrity, and accessibility. Each information security component has its own vulnerabilities. The exploitation of vulnerabilities enables the third party to breach the security [2], either entirely or partially (partial breach of confidentiality, integrity or obtainment of access to the information).

Threat modeling assists in identifying the most vulnerable infrastructure areas. This method is used at all project implementation stages [3].

The number of identified vulnerabilities, including Day Zero, is constantly growing. The same refers to the number of information security experts and hackers. The latest can identify and exploit Day Zero [4] vulnerability that is not yet known to the global community. In this case, modeling may result in inaccurate or false security assessments since the conducted analysis is based solely on common vulnerabilities. The proactive scan would not ensure 100% protection [5] against Day Zero`s vulnerability. This may necessitate the strengthening of safety rings for some links less prone to attacks. However, there may not be enough time to strengthen weaker links.

Today there are different systems for analyzing logs [6], including NIDS (Network Intrusion Detection System) [7]. These systems rely on already known and established parameters to reveal the activity distinct from the "normal" level. This "normal" level is set by the information security specialist based on his/her experience. However, this method has several disadvantages. First, it relies on the experience of the specialist setting up a particular security system. After the successful system installation and configuration, one would only start receiving system alerts for further investigation of the incident. In the beginning we do not know if it is a real attack or a

DOI: 10.5121/ijnsa.2020.12301

non-standard situation planned by the security specialist during a configuration. The analysis of such data can take much time.

It is vital to ensure prompt and adequate response in case of a successful attack. It worth noting that Command and control check, also known as C&C or C2, is a complicated process. The probability of detecting communication between the infected server and the management infrastructure is very low in cases when the infected server uses non-standard (unconventional) options to receive commands, such as tweets, ICMP tunnel, short-range RF protocols (Bluetooth) [8]. For this reason, we need to have tools that can predict an attack or recognize it at the commitment stage.

This paper comprises of 5 sections. Section 1 introduces ontology components to describe a cyberattack. Section 2 covers ontology components, different types of cyberattacks, attack patterns, components of a successful cyberattack, classification of hackers, theory of human needs, and motivation. Section 3 presents the general decomposition of the probability of an attack on any of the security properties. Section 4 provides tools for establishing the probability of an attack and Section 5 summarizes the ideas aimed at improving the current risk prediction methods.

1.1. Motivation

The system threat modeling unveils a probabilistic image of an attack plan. Unfortunately, the simulation is not time-related and we cannot predict the exact attack commitment period.

The periodic information system scanning for known vulnerabilities identifies only the list of system vulnerabilities. This list cannot ensure an accurate risk assessment in all cases. Thus, for SIEM (Security Information and Event Management), it is important to have a list arranged according to the importance of primary actions and reactions. In the SIEM context, it is vital to ensure the correct classification of primary responses according to the vulnerability class. First, it is required to use the results of vulnerability assessment covering the most important assets to ensure their protection against identified critical vulnerabilities.

Currently, there are training developments for Artificial Intelligence (AI) that are formed through the analysis of traffic logs. The application of this approach enables real-time identification of an attack with a certain probability.

Today it is almost impossible to determine the precise attack commitment time and its vector. This confirms the relevance of "prediction of attacks" aimed at identifying the levels prone to risks at every moment. Thus, it is proposed to extend risk prediction [9] to all the existing data (risk indicators history).

To ensure maximum protection of the system from real threats and existing vulnerabilities compatible with certain risk level, it is necessary to recognize the attack stage properly. This recognition is based on the attack ontology. Thus, in this article, we would undertake an attempt to lay the foundation for the prediction of attacks based on the understanding of their vectors [10].

1.2. Our Contributions

The ontology of cyberattacks is based on the aggregated data about them. This knowledge helps to understand the motivation and capabilities of threats. The decomposition of risk into components allows identifying the absolute risk level based on reliable external data, such as OWASP, CVSS, etc.

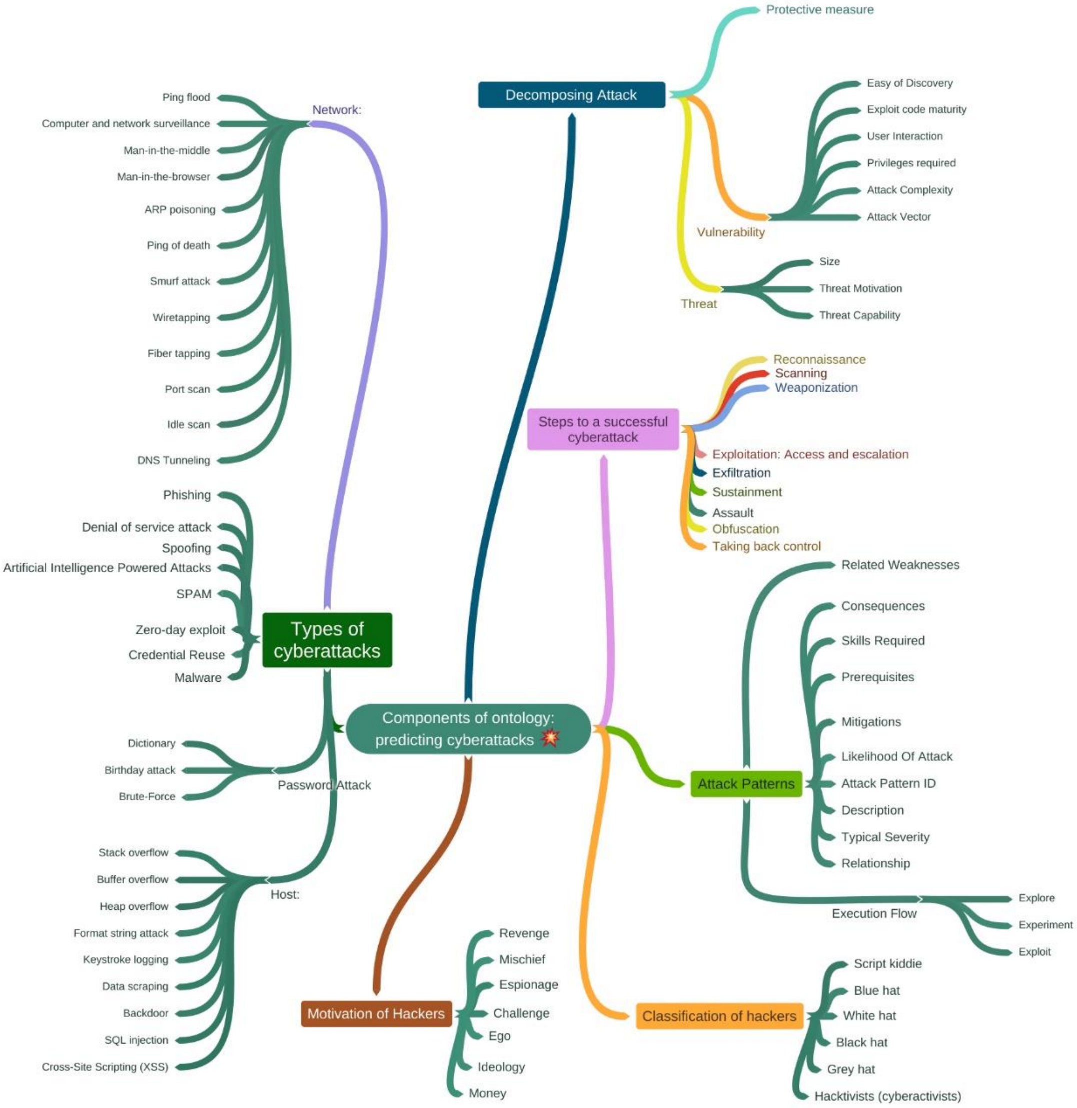

Figure 1. Ontology components

2. COMPONENTS OF ONTOLOGY

2.1. Definitions

In this article, we would only introduce the terms that are essential for its understanding. All other terms would be clarified following their introduction.

In the context of computer networks, an attack (cyberattack) is classified as an attempt to expose, alter, disable, destroy, steal or gain unauthorized access or make unauthorized use of an asset

[11]. Depending on the context, user error may also be categorized as an attack, albeit not intentional.

Attack vector [12] - is a method or path used by the intruder to gain access to the target (asset). The definition used in this article is different from the definition in the CVSS [13].

2.2. Types of Cyberattacks

The commons types of cyberattacks are listed below [14]:

1. Denial of service attack
2. Spoofing
3. Phishing
4. SPAM
5. Password Attack
   (a) Birthday attack
   (b) Brute-Force
   (c) Dictionary
6. Zero-day exploit
7. Credential Reuse
8. Malware
9. Artificial Intelligence Powered Attacks
10. Network:
    (a) Computer and network surveillance
    (b) Man-in-the-middle
    (c) Man-in-the-browser
    (d) ARP poisoning
    (e) Ping flood
    (f) Ping of death
    (g) Smurf attack
    (h) Wiretapping
    (i) Fiber tapping
    (j) Port scan
    (k) Idle scan
    (l) DNS Tunneling
11. Host:
    (a) Buffer overflow
    (b) Heap overflow
    (c) Stack overflow
    (d) Format string attack
    (e) Keystroke logging
    (f) Data scraping
    (g) Backdoor
    (h) SQL injection
    (i) Cross-Site Scripting (XSS)

More details in Common Attack Pattern Enumeration and Classification (CAPEC) [29])

2.3. Attack Patterns

All attack patterns are comprising its ontology, which is understood as a peculiar scheme of presenting an attack taking into account various characteristics. One such approach is an application of CAPEC [10]. CAPEC helps by providing a comprehensive dictionary of the known attack patterns employed by adversaries to exploit known weaknesses in cyber-enabled capabilities. Let us consider the attack characteristics described in CAPEC.

2.3.1. Attack Pattern ID

Name of the attack and its identification number in the knowledge base.

2.3.2. Description

Description of the attack for a detailed understanding of its application context.

2.3.3. Probability of Attack

Presented levels from low to high determined based on the characteristics described in Section 3.1.

2.3.4. Typical Severity

The impact of carrying out a particular attack; separate concept, different from the analysis of the asset at which this attack is directed. Presented levels are from low to high.

2.3.5. Relationship

The connection between the actual attack pattern and other patterns or high-level categories. This relationship is defined as ChildOf and ParentOf. It gives insight to similar items that may exist at higher and lower levels of abstraction. In addition, relationships such as CanFollow, PeerOf, and CanAlsoBe are defined to show similar attack patterns that the user may want to explore. The source [15] shows the views that this attack pattern belongs to and top-level categories within that view.

2.3.6. Execution Flow

The execution flow consists of three phases: Explore, Experiment, and Exploit.

- Explore is an equivalent to entry points finding.

- Experiment is defined as a scenario in which the adversary injects the entry points identified in the Explore Phase with response splitting syntax and variations of payloads to be acted on in the additional response. He/she records all the responses from the server that include unmodified versions of his/her payload.

- Exploit shows which exploit is required for carrying out an attack.

The exploit analysis process typically consists of the following steps:

- Analysis of the exploit through reverse engineering, forensic analysis, and analyzing any available patches by vendors of the target software. This step is not specific to attack pattern analysis and is generally performed to understand exploits and develop countermeasures such as antivirus definitions and spyware removal tools. Once the inner workings of the exploit are revealed, actual attack pattern analysis can start.

- Determination of whether the exploit is an instantiation of any existing attack pattern. Often it is not a clear and unambiguous decision. A careful analysis and comparison must be performed. In most cases where the exploit is discovered in the wild, it will be an existing attack pattern and the analysis will stop here. Otherwise, a new attack pattern which has been discovered would need to be analyzed and documented as described below.
- Determination of the functionality in the software that contained the vulnerability. The functionality could be a file parser, format converter, cookie handler, or anything else. One needs to determine whether the exploit attacks a vulnerability or weakness in the particular functionality or whether the same issue could exist in the software even if the targeted functional component is removed. If the exploit targets specific functionality, then it is recommended to generalize the attack. The vulnerability could be identified in any binary file processor, or it could exist only in MP3 playing software.

- Determination of how the software vulnerability was exploited. Examples include providing a maliciously crafted file to the software, leveraging a race condition, providing separator characters in the input, or bypassing client-side input filtering. This step helps to identify how the targeted functionality determined in the previous step was exploited.

- Determination of skill level or knowledge the attacker would need to execute such an attack.There might be different skill levels and knowledge required to generate certain results. For example, exploiting a buffer overflow to crash a system may require very limited knowledge, but actually executing malicious code on the target host to gain control of it might require sophisticated skills.

- Determination of the resources required to execute an attack. Does the attack simply require an attacker manually entering commands at a terminal, or thousands of hosts have to be compromised before using them to attack the main target? Would the execution of the attack require support by a well-funded organization? It is important to determine the resources required to execute an attack, as it helps to determine the likelihood of an attack and prioritize mitigations during the actual software development.

- Determination of the motivation of the attacker that generates this type of exploit. Why would an attacker choose this type of attack in particular? Having various technical (e.g., executing a buffer overflow) and non-technical (e.g., social engineering) means of achieving a goal, attackers tend to select the easiest ones. Keeping that in mind, one has to determine what makes a particular attack attractive. How does the attacker want to start the attack? What does the attacker want to accomplish? This discussion should be mostly technical since business consequences will obviously depend on the particular software and deployed environment. The consequences may include execution of arbitrary code on the target host, denial of service, obtaining privileged access to target host, etc.

2.3.7. Prerequisites

Description of the necessary vulnerabilities inciting to the commission of an attack.

2.3.8. Required Skills

Description of the level of knowledge required for carrying out an attack.

2.3.9. Consequences

Different individual consequences associated with the attack pattern. The Scope identifies the security property that is violated, while the Impact describes the negative technical impact that arises if an adversary succeeds in committing the attack. Probability provides information on how likely the specific consequence is expected to be seen relative to the other consequences in the list. For example, there may be a high probability that a pattern will be used to achieve a certain impact, but a low probability that it will be exploited to achieve a different impact.

2.3.10. Mitigations

Potential cases helping to reduce the probability of an attack, or its impact.

2.3.11. Related Weaknesses

Relationship associating a weakness with the attack pattern. Each association implies a weakness that must exist for a given attack to be successful. If multiple weaknesses are associated with the attack pattern, then any of the weaknesses (but not necessarily all) may be present for the attack to be successful. Each related weakness is identified by a CWE identifier [16].

2.4. Components of a Successful Cyberattack

The attack is divided into various stages, presented in Figure 2. These are the main components of a successful cyberattack [17]:

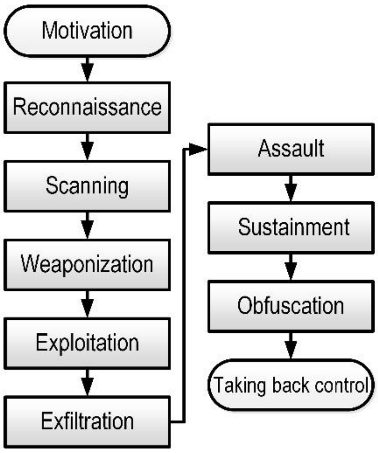

Figure 2. Successful cyberattack

2.4.1. Reconnaissance

The objective of reconnaissance is to check the situation before taking action. Before launching an attack, hackers identify a vulnerable target and explore the best ways to exploit it. Anyone can become the initial target, for example, executive, admin, or third party supplier. The attackers simply need a single entry point to get started. Targeted phishing emails are a common method used during the active reconnaissance to check who might take the bait.

2.4.2. Scanning

Once the target is identified, the next step is the identification of weak points that allow attackers to gain access to it. This is usually accomplished by scanning the organization’s network with tools easily found on the Internet. This step usually goes slowly and may last several months.

2.4.3. Weaponization

The weaponization may be manifested in many forms, including web application exploitation, watering hole attacks, compound document vulnerabilities (delivered in PDF, Office or other document formats), off-the-shelf or custom malware (downloaded for reuse or purchased). These are generally prepared with opportunistic or very specific intelligence. The intruder creates remote access malware weapons, such as a virus or worm, tailored to one or more vulnerabilities, coupling a remote access Trojan with an exploit into a deliverable payload. Increasingly, data files such as Microsoft Office documents or Adobe PDF files have been used as a weapon platform, spawning attacks on other computers.

2.4.4. Exploitation: Access and escalation

Now when the weaknesses in the target network are identified, the next step will be gaining secret access via using of exploit and escalating to moving through the network. In almost all such cases, privileged access is required, since it allows attackers to move freely within the environment. Rainbow Tables and similar tools help intruders steal credentials, escalate privileges to admin, and then get into any system on the network that is accessible via the administrator account. Once the attackers gain elevated privileges, the network is effectively taken over and “owned” by them.

2.4.5. Exfiltration

After obtaining the freedom to move around the network, attackers have a chance to access systems containing the organization’s most sensitive data. They may extract it for any purpose. However, the intruders' activity is not limited by stealing, since they can also modify or erase files on the compromised systems.

2.4.6. Sustainment

After gaining unrestricted access through the target network, the attackers are practicing sustainment (staying in place quietly). Pursuing this objective, the hackers may secretly install malicious software like rootkits enabling their further revisits. Using the previously acquired elevated privileges, they cease relying on a single access point and can come and go at any time.

2.4.7. Assault

Fortunately, this step does not accompany each cyberattack. The assault is classified as the stage of an attack when the things are becoming particularly nasty. At this time hackers might alter the victim’s hardware functionality, or may disable it entirely. The Stuxnet attack on Iran’s critical infrastructure is a classic example. During the assault phase, the attack ceases to be stealth. Consequently, since the attackers have already taken control of the environment, it is generally too late for the breached organization to undertake response measures.

2.4.8. Obfuscation

Usually, the attackers want to hide their tracks, but this is not universally the case – especially if the hackers want to leave a “calling card” behind to boast about their exploits. The purpose of trail obfuscation is to confuse, disorientate and divert the forensic examination process. The trail obfuscation covers a variety of techniques and tools including log cleaners, spoofing, misinforming, backbone hopping, zombified accounts, Trojan commands and more.

2.4.9. Taking back control

According to Mandiant [18], 97 percent of organizations have already been breached at least once. Perimeter security tools, like next-generation firewalls, offer little real protection against advanced, targeted attacks. The key to blocking a cyberattack is controlling privileged access. Each step beyond 2.4.4 in the process described above requires privileged credentials. At the same time, in the case of each successful cyberattack, privileged access was gained despite companies’ investments in “adequate security solutions”.

The privileged identity management can automatically discover privileged accounts throughout the networkand audit access to them. Each privileged credential is updated on a continuous basis. This negates the damage inflicted by advanced cyberattacks. Even if the intruder compromises a credential, it cannot be leveraged to leapfrog between systems, and extract data. The ability to control privileged access significantly mitigates potential cyberattacks.

As any ambitious endeavor, a successful cyberattack requires careful planning and precise execution. One thing that effective hacks have in common is the ability to remain covert right up until the moment they choose to strike by abusing illegitimately gained privileged access rights. Focussing on this element, and getting the security around privileged access tight, will stop attackers from gaining a crucial foothold within a target to rob, and exploit organizations.

2.5. Classification of Hackers

In this subsection, we review different groups of hackers, their motivation and objectives. This subsection also analyzes two theories of needs: Maslow’s theory of hierarchical needs and Herzberg’s motivation-hygiene theory. This analysis is important to predetermine an attacker’s group and its motivation (Figure 3). By predetermining the attacker’s group in advance, it is possible to forecast variants of attack vectors.

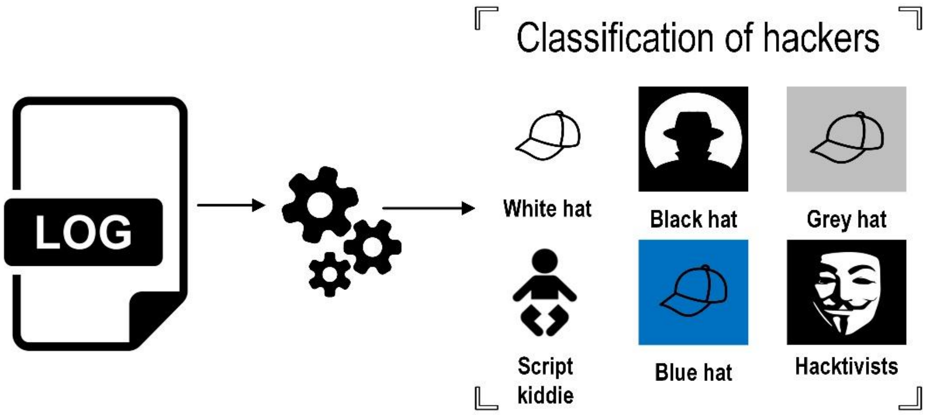

Figure 3. Classification of hackers

In the context of computer security, a "hacker" is a specialist who looks for ways to circumvent the software and hardware protection. The hacker may want to report the found flaws to the owner of the concerned system (to improve its security), take advantage, use them for a politically or socially motivated purpose (hacktivism) or simply consider a bypass (hacking) as a challenge" [19]. Many "underground" subgroups of different kinds use various terms to stand out from each other or try to exclude a specific group with which they disagree. Eric S. Raymond, author of New Hacker's Dictionary [20], proposes to use a term crackers while referring to underground members. Yet this category wants to be singled out from the rest. They even cite Raymond's views while presenting themselves to broader hacker culture, a point of view that Raymond vehemently rejected. Instead of a hacker dichotomy, they focus on a range of different categories, such as white hats, gray hats, black hats, and script kiddies. Unlike Raymond, they generally reserve the term cracker for more malicious activity. These subgroups can also be classified on the based on the legal status of their activities [21].

2.5.1. White hat

A white hat is an ethical hacker or computer security expert who performs intrusion tests and other testing methods to ensure the security of an organization's information systems. By definition, white hats notify the developers (authors) when vulnerabilities are found. This distinguishes them from black hats, who are malicious hackers. A hacker is a computer "smuggler". Both black and white hats are classified as hackers, while they not only trade the information systems, but also identify vulnerabilities that are not made public and never exploited ("Day Zero"). Until this step, one cannot differentiate between the hat colors. This raises the issue of disclosure of vulnerability, namely should it be made public or not. In absolute terms, white hats advocate a full disclosure, while black hats are for restricting an access to the information (to take advantage of these vulnerabilities for as long as possible). The distinction is also made between white hats, who will usually immediately disclose the vulnerability (often with the source code of a program called "exploit" to solve the bug), and the gray hats, who will generally give a reasonable time to companies to fix the problem before making the vulnerability public, and rarely publish the source code to exploit the security breach. However, malicious individuals can ensure that appropriate computer codes are made public by some white hats, to cause system failures, "mass-root", etc. These individuals are then called script kiddies.

2.5.2. Black hat

A black hat is, in computer slang, a malicious hacker, as opposed to white hats, who are hackers with good intentions. These terms originated in western movies, where a hero or sheriff wears a white hat, while a criminal wears a black hat. Black hats have clear preference for illegal actions. Their activities are rather broad, including: creation of viruses, Trojans, worms and spyware. This group uses computer skills to make a financial profit or harm individuals or organizations (in the latest case we are talking about cybercrime or cyberterrorism). More generally, they use their knowledge to identify things that are hidden from them. Their numbers are constantly growing due to the increasing value of information in the economic war. It is also known that some black hats are assisting the companies specializing in computer security. Thus, Sven Jaschan, the creator of a Sasser virus, was recruited by the German SME Securepoint in 2005. The black hats community is heterogeneous and its members do not always recognize each other because of differences in opinions, abilities or philosophy.

2.5.3. Grey hat

The term gray hat usually reffers to a hacker or a group of hackers who sometimes act ethically, and sometimes not. This term is used to refer to those between white hats and black hats. An ethical hacker is a professional who applies his/her knowledge adhearing to both legal and moral norms, while a non-ethical hacker by contrast has no dogma, does not follow morality, and acts illegaly to achieve his/her goal. A common example is a person who illegally accesses a computer system without destroying or damaging it (at least not voluntarily), and then informs developers of that computer system of the existence of security breach and possibly makes some suggestions in order to solve this issue. Despite these good intentions, it is still considered to be a crime in most countries. Indeed, there is evidence that people were convicted for filling security holes in a computer system they accessed illegally. This category also includes individuals who will discover a new security vulnerability in a software or computer system (including protocol design errors, equipment vulnerabilities, for example in case of routers) and notify the developer and users about them in order to find a solution. Gray hats are acting in the name of the ideology they consider fair, committing offenses not for their own profit, but for the purpose of fighting for a cause (examples include freedom of expression and protection of privacy for the gray hat hacker groups like LulzSec and Anonymous). The exploited community can also be considered one of the gray hat groups, although it has nothing common with any ideology or hacktivism.

2.5.4. Script kiddie

Script kiddie or lamer is a pejorative term of English origin referring to neophytes who, lacking the main skills in managing computer security, spend most of their time trying to infiltrate the systems using scripts or software developed by others. Despite their low qualification level, script kiddies sometimes represent a real threat to the system security. Indeed, besides the fact that they can incompetently alter something without wanting or knowing it, they are very numerous. They are also often stubborn enough to spend sometimes several days trying out all possible combinations of a password until they achieve their goal. It is very common that script kiddies themselves are becoming infected.

2.5.5. Blue hat

A blue hat is a computer security consultant who is responsible for checking bugs and correcting any "exploits" before the market launch of any operating system. The term is widely used by Microsoft to distinguish between its hackers and computer security engineers whose role is to find the vulnerabilities in Windows.

2.5.6. Hacktivists (cyberactivists)

Cyberactivism is the process of using Internet-based socialising and communication techniques to create, operate and manage activism of any type. A hacktivist infiltrates computer networks for militant purposes [23] and organizes technological punching operations: hacking, use of hijacking servers, replacement of homepages with leaflets (alterations), dissemination and theft of confidential data, etc. The majority of hacktivists claims that they belong to the so-called "Hacker" culture. They are proponents of the idea of open Internet with complete freedom of expression. They are generally libertarians and mainly concerned with oppression and human rights. The methods of hacktivism are widely used internationally during geopolitical or religious conflicts in order to censor political opponents [24]. Due to the illegal nature of hacking, hacktivists generally remain anonymous and those who appear in videos are using voice synthesizers and hiding their identity. Some hacktivists are becoming whistleblowers. They distribute the confidential data obtained as a result of hacking and denounce the actions they seek to reveal by making them known to the public. However, even despite this, they still fail to help governments, law enforcement agencies, businesses and the community to explore the nature of motivation behind their activities [24]. Although these determinants are known to indicate the propensity of an individual to commit a crime, what really motivates him to act so? It’s a question of real motivation.

2.6. Theory of Human Needs and Motivation

In 1954 Maslow, as summarized by Hunt [25], hypothesized five large classes of hierarchically arranged needs:

1. physiological needs;
2. security or safety needs;
3. social or membership needs;
4. esteem needs, further subdivided into self-esteem and esteem for others;
5. self-actualization or self-completion needs.

Herzberg’s motivation-hygiene theory of 1959 is classified as a two-factor theory [26]. The Herzberg model presents a dichotomy [26] known as the motivation-hygiene theory [27], i.e. "characteristics of the content of work", which stand for motivations responsible for satisfaction; and "characteristics of the workplace", that is hygiene responsible for dissatisfaction at work [25]. Motivational factors include achievement, recognition, advancement, the opportunity for growth, responsibility, and the work itself. Hygiene, on the other hand, extends to salary; interpersonal relationships with superiors; subordinates and peers; technical supervision; the policy and administration of the company; private life; working conditions; statute and job security. Table 1 below illustrates how Maslow’s and Herzberg’s models relate by conveying the notion of motivation. There is some degree of overlap of categories in the models. However, the main idea here is to show how the two theories "agree" with each other.

To sum up, it is worth to mention that although a "traditional crime" differs from a "cybercrime", they both share a common denominator: "crime"; therefore, the determinants and motivational factors of conventional crime and cybercrime would be the same, as they differ only in each medium. In addition, by assessing the motivation of cybercriminals, it is safe to predict that criminal action will be motivated either by "need" (Maslow model) or by "work content / environmental characteristics" (Herzberg model). On the lower level, Herzberg had classified hackers into seven categories, as in table 1 below. Furnell's works [28] then characterized these types of hackers based on their motivation [29]. Since the purpose of this part of the research is to emphasize the motivational aspects that it highlights, no attempt is made to define the categories of hackers singled out by him. The motivation categories used to classify the types of hackers mentioned above include challenge, ego, spying ambition, ideology, mischief, money, and revenge. These categories may be successfully integrated into the Maslow-Herzberg model, but not the other way around. Therefore, it is possible to conclude that the Maslow-Herzberg model is a more holistic and explanatory model. The main correspondences between the hacker class and its motivation are presented in Table 1 below.

Table 1. Hackers and their motivation

	White hat	**Black hat**	**Grey hat**	**Script kiddie**	**Blue hat**	**Hacktivists**
Challenge	+	+	+	+	+	
Ego	+	+	+	+	+	
Espionage		+	+			+
Ideology		+		+		+
Mischief		+	+			
Money	+	+	+			
Revenge		+	+	+		+

The attempt to define a hacker's class in advance can forecast its behavior based on the general behavior of the group to which he/she belongs. The behavior of a hacker may differ from the average behavior of one of the classes, which may indicate the emulation of a particular behavior [30]. This "pretense" can be considered as a disguise of the true hacker's objectives. To determine both assets (hacker's potential targets) and risks, it would be more appropriate to use the existing risk assessment techniques [31].

3. Decomposing Attack Probabilities into Probability Components

Let us consider the general decomposition of the probability of an attack on any of the security properties. The graphical representation of decomposition is shown in Figure 4.

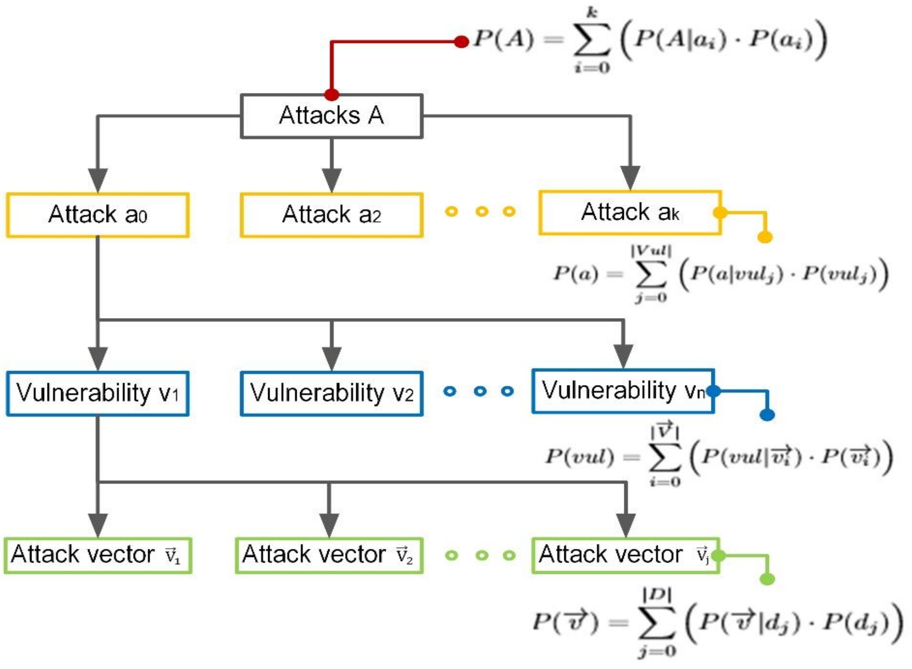
$P(A) = \sum_{i=0}^{k} \left(P(A|a_i) \cdot P(a_i)\right)$
Attacks A
Attack a0
Attack a2
Attack ak
$P(a) = \sum_{j=0}^{|Vul|} \left(P(a|vul_j) \cdot P(vul_j)\right)$
Vulnerability v1
Vulnerability v2
Vulnerability vn
$P(vul) = \sum_{i=0}^{|\vec{v}|} \left(P(vul|\vec{v_i}) \cdot P(\vec{v_i})\right)$
Attack vector $\vec{v}_1$
Attack vector $\vec{v}_2$
Attack vector $\vec{v}_j$
$P(\vec{v}) = \sum_{j=0}^{|D|} \left(P(\vec{v}|d_j) \cdot P(d_j)\right)$

Figure 4. Attack probability decomposition: exploiting the existing vulnerability using attack vector

3.1. Attacks

The formula presented below could be used to express the probability of an attack (a_i) in case of all the attacks A on one of the assets. It is applicable to any security component (availability, integrity or confidentiality):

$$P(A) = \sum_{i=0}^{k} \left(P(A|a_i) \cdot P(a_i)\right) \tag{1}$$

where,

- $P(A|a_i)$ - probability of an attack (a_i) in general cyberattack statistics [32].
- $P(a_i)$ - probability of an attack (a_i), Formula (2) is used.

Let us analyze the lower level to determine the probability of an attack a through exploiting vulnerabilities Vul.

$$P(a) = \sum_{j=1}^{|Vul|} \left(P\left(a\middle|vul_j\right) \cdot P\left(vul_j\right)\right) \tag{2}$$

where,

- $|Vul|$ - number of vulnerabilities enabling commitment of an attack a.
- $P\left(a\middle|vul_j\right)$ - probability of an attack a occurring in the presence of a threat of exploiting a vulnerability vul_j. Formula (3) is used to determine the threats.
- $P\left(vul_j\right)$ - probability of occurrence of a vulnerability vul_j event presented in Formula (4).

3.2. Threats

Formula (2) is used to determine the threats:

$$P\left(a\middle|vul_j\right) = \text{ThreatMotivation} \times \text{ThreatCapability} \times \text{Size} \tag{3}$$

where,

- Threat Motivation [33] - assesses motivation (2.6) of the threat agents group to find and exploit a vulnerability.
- Threat Capability [34] - probable level of resistance that a threat agent (2.5) is capable to demonstrate against an asset.
- Size [33] - characteristics of threat agents (developers, system administrators, intranet users, partners, authenticated users, anonymous Internet users).

3.3. Vulnerabilities

The probability of occurrence of a vulnerability vul_j event presented in Formula (2) depends on the total probability of exploitation of attackvectors $\overleftarrow{V}$ presented in Formula (4).

$$P(vul) = \sum_{i=0}^{|\overleftarrow{V}|} \left(P(vul|\overleftarrow{v_i}) \cdot P(\overleftarrow{v_i})\right) \qquad (4)$$

where,

- $|\overleftarrow{V}|$ - identifies the number of vulnerabilitiesvul for the attack a;
- $P(vul|\overleftarrow{v_i})$ - exploitability of vulnerability using attack vector $\overleftarrow{v_i}$ from the variety of attacks vectors for the vulnerability vul, $P(vul|\overleftarrow{v_i})$ presented in Formula (5). The exploitation of a vulnerability in the presence of a chance to exploit it through a certain method by the attack vector.
- $P(\overleftarrow{v_i})$ - attack vector probability can be tracked by analyzing Formula (6).

3.4. Exploitability

$P(vul|\overleftarrow{v_i})$defines the exploitation of a vulnerability by the attack vector in the presence of a chance to employ a certain method to reach this objective. The following formula is used by us to establish theexploitability degree:

$$P(vul|\overleftarrow{v_i}) = \text{AttackVector} \times \text{AttackComplexity} \times$$

$$\times\ \text{PrivilegesRequired} \times \text{UserInteraction} \times$$

$$\times\ \text{ExploitCodeMaturity} \times \text{EasyofDiscovery} \qquad (5)$$

where,

- Attack Vector [13] - Common Vulnerability Scoring System (CVSS) metric reflecting the context in which it is possible to exploit the vulnerability.
- Attack Complexity [13] - CVSS metric describing the conditions beyond the attacker's control that must be created to exploit the vulnerability.
- Privileges Required [13] - CVSS metric describing the level of privileges an attacker must possess before successfully exploiting the vulnerability.
- User Interaction [13] - CVSS metric capturing the requirement for a human user, other than an attacker, to participate in the successful compromise of the vulnerable component.
- Exploit Code Maturity [13] - CVSS metric measuring the likelihood of the vulnerability being attacked. It is typically based on the current state of exploit techniques, exploit code availability, or active "in-the-wild" exploitation. Public availability of easy-to-use exploit code increases the number of potential attackers by including those who are unskilled, thereby increasing the severity of the vulnerability.
- Easy of Discovery [33] - describes the degree of easiness for a group of threat agents targeting a particular vulnerability to get access to it (practically impossible, difficult, easy, automated tools available).

3.5. Attack vectors

Let us consider the formula for the probability of an attack vector (6), the result of which is used in Formula (4).

$$\mathrm{P}(\overleftarrow{v}) = \sum_{j=1}^{|D|} \left(P\left(\overleftarrow{v}\middle|d_j\right) \cdot P\left(d_j\right) \right) \qquad (6)$$

where,

- $|D|$ - preventive measures against the attack vector $\overleftarrow{v}$. Here we analyze individual protection components D in isolation, however, it is very important to consider different sets of protection components and their configuration and interactions.

- $P\left(\overleftarrow{v}\middle|d_j\right)$ - damage caused by attack vector $\overleftarrow{v}$ with valid protection measures d_j (return value of quality of protection against an attack vector) and statistical data.

- $P\left(d_j\right)$ - probability of applying this protection measure (yes or no).

For simplicity purposes, the component of formula $P\left(\overleftarrow{v}\middle|d_j\right)$ is visualized in Table 2.

Table 2. Visualization of component $P\left(\overleftarrow{v}\middle|d_j\right)$

	$\overleftarrow{\boldsymbol{v_1}}$	$\overleftarrow{\boldsymbol{v_2}}$	**…**	$\overleftarrow{\boldsymbol{v_\iota}}$	**…**	$\overleftarrow{\boldsymbol{v}_{\lvert\overleftarrow{V}\rvert}}$
d_1	$P(\overleftarrow{v_1}\lvert d_1)$	$P(\overleftarrow{v_2}\lvert d_1)$				
d_2	$P(\overleftarrow{v_1}\lvert d_2)$	$P(\overleftarrow{v_2}\lvert d_2)$				
…						
d_j				$P\left(\overleftarrow{v_\iota}\middle\lvert d_j\right)$		
…						
$d_{\lvert D\rvert}$						$P\left(\overleftarrow{v_{\lvert\overleftarrow{V}\rvert}}\middle\lvert d_{\lvert D\rvert}\right)$

4. PREDICTION OF A CYBER ATTACK BASED ON THE CATEGORIZATION OF THREATS

The prediction element was added by us in the above risk analysis methodology. In this article we will consider the prediction of the activity of threats (Section 3.2), as the basis for predicting attacks.

The threat activity analysis helps to establish the probability of an attack in the future period. Its graphical representation is displayed in Figure 5.

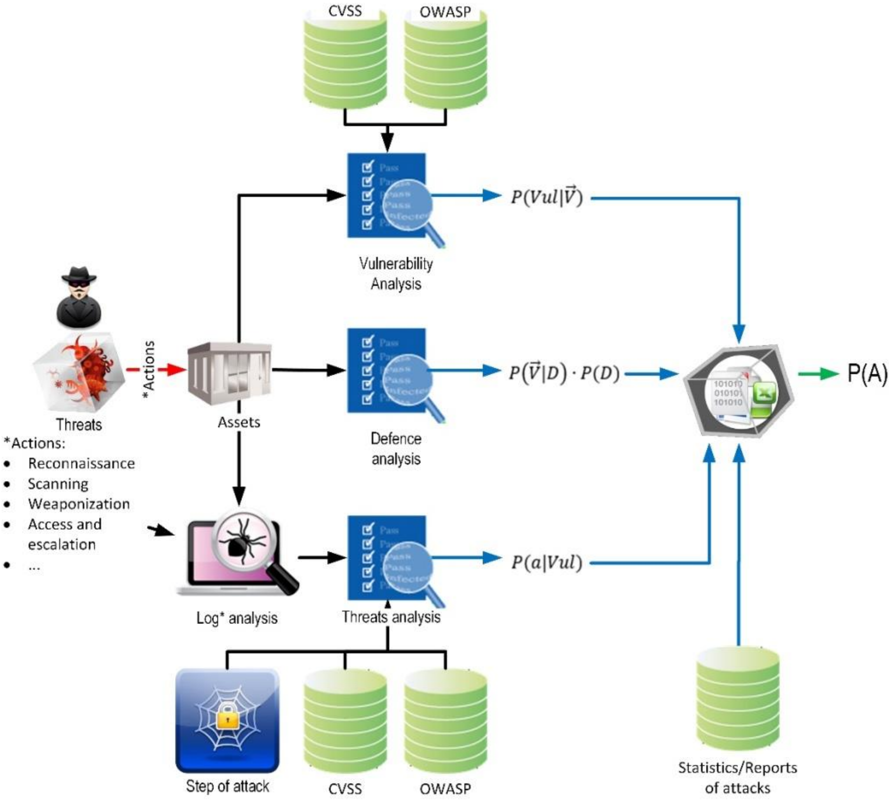

Figure 5. Threat activity analysis for establishing an attack probability

We decomposed a cyberattack into components bearing in mind that for each of these components, hypothetically, we can identify the mechanism for predicting subsequent probability values at a certain period of time. This means that we can calculate the level of risk for a given point in time in the future, based on historical data and their specifics.

We think that it would be most logical to look at the threats and link their activities with the time to identify the period at which the attack commitment is planned. The chances to explore the vulnerabilities depend on the attacker's skills and creativity. They are also predetermined by the mistakes made during the development. Frequently, it is almost impossible to predict all these parameters [35]. It is also impossible to ensure 100% protection of the system from cyberattacks. As discussed in chapter 2.4, the attack is always split in stages. Each of the stages requires a certain time for implementation. The consideration of this time is important for predicting the next stage of an attack. The attack commitment stages are presented in Figure 6. The establishment of an attack pattern in the logs at the Reconnaissance stage would help to save more time for the preparation of response measures.

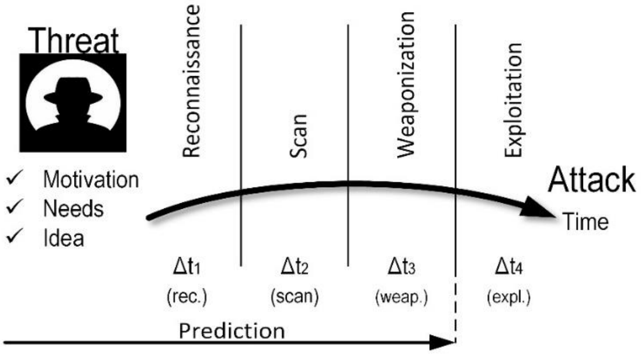

Figure 6. Cyberattack prediction stages

Each threat is preconditioned by certain motivation, budget and knowledge level. The speed of transition from the previous stage to the next one will depend on these parameters. To predict an attack, one must check the following attack stages: Reconnaissance 2.4.1, Scanning 2.4.2, Weaponization 2.4.3. The subsequent stages of attacks are viewed as the stages that do not need to be predicted. One only has to ensure an effective response to the information security incident which took place.

Let us rewrite Formula (2) taking into account the fixed stage of the attack in the log for the attack time of prediction:

$$P_{[t;t+\Delta t]}(a) = \sum_{j=1}^{|Vul|} P_{[t;t+\Delta t]}\left(a \middle| vul_j\right) \cdot P_{[t;t+\Delta t]}\left(vul_j\right) \quad (7)$$

where,

- $|Vul|$ - number of vulnerabilities enabling commitment of an attack a.
- $P_{[t;t+\Delta t]}\left(a \middle| vul_j\right)$–probability of an attack a occurring if there is a threat of exploitation of the vulnerability vul_j among other vulnerabilities of attack a_j.

For this reason,we should consider rewritingFormula (3) considering the time of prediction of the attack $a$and the vulnerability vul_j in the time interval $[t; t+\Delta t]$:

$$P_{[t;t+\Delta t]}(a|vul_j) = \sum_{\phi_i}^{|\text{Threats}|} \left[P_t(\phi_i) \cdot P_{[t;t+\Delta t]}\left(\left(a \middle| vul_j\right) \wedge (\phi_i \rightarrow \text{attack})\right)\right] \quad (8)$$

where,

- $|\text{Threats}|$ - number of threats logged.
- ϕ_i - attack phase i (Reconnaissance, Scanning, etc.).
- $P_t(\phi_i)$ - probability of being in the attack phase ϕ_i in the time t.
- $P_{[t;t+\Delta t]}\left(\left(a \middle| vul_j\right) \wedge (\phi_i \rightarrow attack)\right)$ - probability of a successful attack a in a given period of time $[t; t+\Delta t]$ if the attacker is in the attack phase ϕ_i.

The determination of the attack phase $P_t(\phi_i)$ could be ensured by analyzing the logs using a machine learning approach. This approach will contribute to a more accurate determination of the attack phase ϕ_i based on the history of actions recorded in the logs.

The definition of *log* comprises not only information about the actions of the protected infrastructure, but also the analysis of social networks [36][37], like Twitter [38]; Dark Web [39], etc.

The statistical data needs to be obtained to determine $P_{[t;t+\Delta t]}\left(\left(a\middle|vul_j\right) \wedge (\phi_i \rightarrow \text{attack})\right)$, which is the probability of completion of an attack from phase ϕ_i . The statistical data is required to determine the probability of an attack over time $[t; t + \Delta t]$ from the time the first attack pattern appears in the logs in the time $[t]$. This parameter $P_{[t;t+\Delta t]}$ affects the following components reviewed inSection2: Threat Motivation, Threat Capability, Size.

In this article we undertook an attempt to prove that risk level can be forecasted. The proper analysis of threats provides an opportunity to design proper response or emergency measures to repel the upcoming attacks or reduce their impact. This analysis helps to identify the exact probability (not a conditional estimate). The vulnerability assessments are based on CVSS data. However, the probability of threats calculated byapplying formula 8 is not quite accurate, since this value is based on the conditional probability of an event in the future, and not on specific values. Also, it is rather complicated to establish the exact values of the defence system quality parameters against the attack vector. Even not all security system manufacturers are able to provide such information.

5. FUTURE WORK

In future work, we plan to review machine learning algorithms in more detail to determine the attack phases based on logs.This would help to increase forecasting accuracy. We consider using a big data approach to select more informative parameters for predicting threats. In cases when the long-term resistance to specific threats is identified, the game theory will be used. To ensure successful countermeasures against the attack, it will be essential to obtain the knowledge on whether the system responds properly to the future attacks against it. This will help to reconsider and reconfigure its protection components P(v ↼ | d_j)listed in Section 3.5. Therefore, weadviseto consider the system stability prediction concepts in dynamics.

6. CONCLUSION

The article reviewed the main components for determining the probability of cyberattacks. Due to the popularity of this topic, many organizations offer different definitions when introducing the terms referred in to the cybersecurity field. This article attempted to define the key terms from this domain to ensure a more accurate understanding of the context. We unified different attack types and described various types of attackers, their motives, and capabilities. We believe that this information would contribute to identifying the severity of threats. Additionally, a mathematical approach was presented to determine the numerical indicators of the probability of cyberattacks. The obtained finding allows us to conclude that a more accurate prediction of future cyberattacks could be ensured upon relying on the successful experience of predicting the probability of attacks from the existing threats. Currently, a lot of systems are identifying the level of risk without analyzing its values. At the same time, such an analysis could contribute a lot to the improvement of corporate information security.

REFERENCES

[1] J. M. Stewart, M. Chapple, and D. Gibson, CISSP: Certified Information Systems Security Professional Study Guide. John Wiley & Sons, 2012.

[2] S. Barnum and A. Sethi, “Attack patterns as a knowledge resource for building secure software,” in OMG Software Assurance Workshop: Cigital, 2007.

[3] S. Myagmar, A. J. Lee, and W. Yurcik, “Threat modeling as a basis for security requirements,” in Symposium on requirements engineering for information security (SREIS), vol. 2005, 2005, pp. 1–8.

[4] “Vuln´erabilit´e zero-day,” http://bit.ly/2KDVSq2, accessed: 2018-02-25.

[5] “Proactive cyber defence,” http://bit.ly/2XaCPVO, accessed: 2018-02-25.

[6] S. Dinesh, S. Rao, and K. Chandrasekaran, “Traceback: A forensic tool for distributed systems,” in Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics. Springer, 2016, pp. 17–27.

[7] Y. Altman and A. Y. Keren, “System and method for automated configuration of intrusion detection systems,” Oct. 25 2016, US Patent 9,479,523.

[8] “Security hacker.” http://bit.ly/2IxhbGX, accessed: 2018-03-11.

[9] P. Yermalovich and M. Mejri, “Determining the probability of cyberattacks,” European Journal of Engineering and Formal Sciences, vol. 4, no. 1, pp. 46–63, 2020.

[10] “Common attack pattern enumeration and classification.” http://bit.ly/37s4xlo, accessed: 2020-02-13.

[11] “Information technology — Security techniques — Information security management systems — Overview and vocabulary,” International Organization for Standardization, Geneva, CH, Standard, Feb. 2018.

[12] D. Achmadi, Y. Suryanto, and K. Ramli, “On developing information security management system (isms) framework for iso 27001-based data center,” in 2018 International Workshop on Big Data and Information Security (IWBIS). IEEE, 2018, pp. 149–157.

[13] “Common vulnerability scoring system v3.1: User guide.” https://bit.ly/33JM6HK, accessed: 2019-10-18.

[14] “What are the most common cyber attacks?” http://bit.ly/2SpVFJ0, accessed: 2020-02-14.

[15] “Attack patterns. CISA is part of the department of homeland security.” http://bit.ly/2PQ1ygT, accessed: 2020-03-07.

[16] “CWE is a community-developed list of common software security weaknesses. it serves as a common language, a measuring stick for software security tools, and as a baseline for weakness identification, mitigation, and prevention efforts.” https://cwe.mitre.org/, accessed: 2020-02-19.

[17] “7 steps to a successful cyber attack.” http://bit.ly/2tYfPjM, accessed: 2020-02-14.

[18] S. Donaldson, S. Siegel, C. K. Williams, and A. Aslam, Enterprise cybersecurity: how to build a successful cyberdefense program against advanced threats. Apress, 2015.

[19] “Hacker.” http://bit.ly/2UhbrmW, accessed: 2018-03-11.

[20] E. S. Raymond, The new hacker’s dictionary. Mit Press, 1996.

[21] “Security hacker.” http://bit.ly/2GlN1Fe, accessed: 2018-03-11.

[22] M. Pinard, “L’hacktivisme dans le cyberespace: quelles r´ealit´es?” Revue internationale et strat´egique, no. 3, pp. 93–101, 2012.

[23] A. Mereuil and A.-M. Bonnefous, “Anatomie d’une cyber-attaque contre une entreprise: comprendre et pr´evenir les attaques par d´eni de service,” in Annales des Mines-G´erer et comprendre, no. 1. FFE, 2016, pp. 5–14.

[24] M. Ngafeeson, “Cybercrime classification: a motivational model,” College of Business Administration, The University of Texas-Pan American, vol. 1201, 2010.

[25] J. Hunt and J. Hill, “The new look in motivation theory for organizational research,” Human Organization, vol. 28, no. 2, pp. 100–109, 1969.

[26] P. Louart, “Maslow, herzberg et les th´eories du contenu motivationnel,” Les cahiers de la recherche, CLAREE Centre Lillois d’Analyse et de Recherche sur l’Evolution des Entreprises, 2002.

[27] F. Herzberg, “Motivation-hygiene theory,” J. Miner, Organizational Behavior I: Essential Theories of Motivation and Leadership, pp. 61–74, 2005.

[28] S. Furnell, “The problem of categorising cybercrime and cybercriminals,” in 2nd Australian information warfare and security conference, vol. 2001, 2001.

[29] S. Proulx and S. Couture, “Pratiques de coop´eration et ´ethique du partage `a l’intersection de deux mondes sociaux: militants du logiciel libre et groupes communautaires au qu´ebec,” JM Penalva, ´ed., Intelligence Collective. Rencontres, vol. 2006, pp. 137–152, 2006.

[30] B. Idiri, “M´ethodologie d’extraction de connaissances spatio-temporelles par fouille de donn´ees pour l’analyse de comportements `a risques: application `a la surveillance maritime,” Ph.D. dissertation, Ecole Nationale Sup´erieure des Mines de Paris, 2013.

[31] G. A. Zsidisin, L. M. Ellram, J. R. Carter, and J. L. Cavinato, “An analysis of supply risk assessment techniques,” International Journal of Physical Distribution & Logistics Management, vol. 34, no. 5, pp. 397–413, 2004.

[32] “Hackmageddon. information security timelines and statistics. cyber attacks statistics: Motivations behind attacks, attack techniques, targets.” https://bit.ly/2ohjjuT, accessed: 2019-10-05.

[33] “Owasp risk rating methodology,” http://bit.ly/2pp496W, accessed: 2019-10-02.

[34] M. U. Aksu, M. H. Dilek, E. ˙I. Tatlı, K. Bicakci, H. I. Dirik, M. U. Demirezen, and T. Aykır, “A quantitative cvss-based cyber security risk assessment methodology for it systems,” in 2017 International Carnahan Conference on Security Technology (ICCST). IEEE, 2017, pp. 1–8.

[35] P. Yermalovich and M. Mejri, “Formalization of attack prediction problem,” in 2018 IEEE International Conference” Quality Management, Transport and Information Security, Information Technologies”(IT&QM&IS). IEEE, 2018, pp. 280–286.

[36] A. Okutan, S. J. Yang, and K. McConky, “Predicting cyber attacks with bayesian networks using unconventional signals,” in Proceedings of the 12th Annual Conference on Cyber and Information Security Research, 2017, pp. 1–4.

[37] B. Munkhdorj and S. Yuji, “Cyber attack prediction using social data analysis,” Journal of High Speed Networks, vol. 23, no. 2, pp. 109–135, 2017.

[38] A. Hernandez-Suarez, G. Sanchez-Perez, K. Toscano-Medina, V. Martinez-Hernandez, H. Perez-Meana, J. Olivares-Mercado, and V. Sanchez, “Social sentiment sensor in twitter for predicting cyber-attacks using l1 regularization,” Sensors, vol. 18, no. 5, p. 1380, 2018.

[39] S. Sarkar, M. Almukaynizi, J. Shakarian, and P. Shakarian, “Predicting enterprise cyber incidents using social network analysis on dark web hacker forums,” The Cyber Defense Review, pp. 87–102, 2019.