Ronierison Maciel*, Jean Araujo, Carlos Melo, Paulo Pereira, and Paulo Maciel

Impact evaluation of DDoS and Malware attack using IoT devices

Abstract: Distributed Denial-of-Service (DDoS) attacks deny access to infrastructures of service providers. These attacks can arise anytime, anywhere, and with little or no warning at all. Most of the Small and Medium Business (SMBs) are not able to handle a significant outage, which may be fatal for these companies. These attacks generate damage to enterprises due to service provisioning interruption, which increases the chances of financial losses, and the system's unavailability. Therefore, to overcome these issues, the companies must possess a bandwidth higher than the attacker, redundant components in their infrastructure, regular backups, firewalls, other proactive and reactive mechanisms for threat monitoring. This chapter explores DDoS and Malware attacks that employ the Internet of Things (IoT) devices. Hierarchical modeling is commonly used to evaluate the availability of such systems. This chapter also assesses the DDoS attack impacts and Malware in IoT devices. It was proposed models based on attack trees that produce the system and components behavior to determine the DDoS and Malware attack effects on system availability; still, it was verified metrics of interest as the likelihood of an attack, attacker benefit, feasibility, and pain factor. The attack tree indicators show the impact of the concurrent attacks using vulnerable IoT devices on a computer system, which can cause a system's downtime. Using the attack tree analysis, we allow planning and improving the system's availability, maintainability, and reliability. The obtained results show that DDoS attacks orchestrated by IoT devices correlate negatively with Malware and affect the system's availability and services.

Keywords: DDoS, IoT, Attack Tree, Malware, Cybersecurity

***Corresponding author: Ronierison Maciel, Carlos Melo, Paulo Pereira, Paulo Maciel,** Centro de Informática, Universidade Federal de Pernambuco, Recife, Brazil,{rsm4, casm3, prps, prmm}@cin.ufpe.br

Jean Araujo, Unidade Acadêmica de Garanhuns, Universidade Federal Rural de Pernambuco, Garanhuns, Brazil, {jean.teixeira}@ufrpe.br

1 Introduction

The term Denial-of-Service (DoS) was initially presented by Gligor [33] in the context of operating systems. It has since been associated with many types of network attacks. The DoS attacks have been identified as one of the most severe threats to Internet services [45]. Now, the DoS attacks are denominated Distributed Denial-of-Service (DDoS) attacks; they are characterized by orchestrating a distributed attack, where many infected nodes are accessing one system [32].

The first DDoS was orchestrated in 1988 with a malware known as Morris worm[1], where at least 60,000 nodes were infected [63]. However, the first DDoS attack cataloged by Cyber Emergency Response Team (CERT) occurred ten years later, in 1998, and received the name of Smurf attack [74]. It was as unauthorized access, resulting in several problems to victims, such as customer and credibility losses, as well as financial issues and other factors that lead systems to downtime [62].

According to TechRepublic [68], in Q1 2019, the 100 Gbps or higher DDoS attack increased 96.7% compared to Q1 2018. The frequency of DDoS attacks increased more than 2.5 times between 2014 and 2017 [1]. Mischievous actions usually cause financial losses, for example, in data centers management, health systems, and real-time operating systems [59].

Attack Trees (AT) was cited by Schneier [70]. However, they seem to have more time, according to [29, 20]. Fault tree (FT) was described by H. A. Watson on the Bell Labs, under the U.S Air Force. The fault tree analysis (FTA) is a commonly used technique for analysis in risk and reliability studies [87, 24]. Furthermore, both methods have similarities.

This book chapter portrays the DDoS and Malware impact using the Internet of Things (IoT) on vulnerable devices; the book chapter is organized as follows; Section (2) addresses the main cybernetic threats, DDoS attacks principles, and the state-of-art in the security of IoT devices. Section (3) presents the motivation to modeling using AT formalism, and the step-by-step to construct an AT. Section (4) shows the case studies that demonstrate the AT applicability. Finally, in Section (5), we discuss the obtained values and conclude.

[1] The worm of Robert T. Morris was developed purely for a research purpose. However, Morris pleaded guilty by the attack, repented, and was given 400 hours' community service and a fine of $10,000 [25].

2 Cybernetic threats

Cybernetic threats cause several problems in our daily lives, such as electrical blackouts, cloned credit cards, and so on [43]. All of these threats may result in the theft of valuable, sensitive data like government and classified information [27]. Therefore, these threats may affect the way of life as we know. This section presents the main concepts about Malware, DDoS attacks, and the cybernetic threats, as well as the Internet of Things (IoT) devices vulnerabilities.

2.1 What is Malware?

Malware or malicious software is a term first presented by Radai in 1990 [67] and refers to any malicious program or code that is adverse to systems. The first PC-based Malware, known as Brain[2], was built in 1986 and distributed in floppy disks [57, 5]. The Malware is usually classified into six categories [55, 66], according to its goal and propagation method:

The Virus is a program that self-replicate within a host by attaching themselves to applications and/or documents that become a carrier of the malicious code;

Worms is a malicious code that self-replicates across the network;

Trojan horse is usually disguised as useful programs, but hold malicious code to attack the system, leak info or install threats like viruses and ransomware;

Ransomware is a method that locks the access to a device and/or encrypts files, then forces to pay a ransom to get the data back;

Backdoor explore the system vulnerabilities, subverting local security policies to allow remote access and control over a network;

Spyware is Malware that privately sees the computer user's actions lacking permission and reports it to the software's author.

Table 1 provides several representative types of Malware and many of his underlying subcategories, e.g., trojan, viruses, and their variations. Also, we supply some notable cases of each Malware type, as well as a qualitative severity ranking of its operations based on their effects [6, 17, 51, 57, 61, 72].

In this subsection, we approached the main threats through Malware and the severity of attacks. Next subsection, we will describe the DDoS attacks and their taxonomy.

[2] A Malware developed in Pakistan, by two brothers, Basit and Amjad.

Malware Type	Notable Attacks	Severity
Adware	Typhoid, Hijackthis	Low
Botnet	Mirai, Pushdo, Cutwail, Cyclone	High
Bluetooth viruses	Ronie, Commwarrior	Average
DDoS	Smurf, Loic, HOIC, Slowloris, Hping3	High
Ransomware	WannaCry, Petya, GonnaCry	High
Remote Access Trojan (RAT)	Gh0st, Dridex, NanoCore, Mydoom	High
Phishing	Warez, Heartbleed, Wifiphisher	High
Trojan horse	Zeus, IcedID, Emotet, Revealer, Spyrix	High
Worm	Badtrans, Bagle, Blaster, Daprosy Worm	High

Table 1: Types of Malware

2.2 Distributed Denial-of-Service (DDoS) Attacks

The DDoS coordinated attacks adopt a large number of compromised hosts [32, 58]. The malicious agent identifies the vulnerabilities and install malicious software in several machines to control remotely [21]. The computer infected is called zombie[3]. In this next step, the attacker coordinates all its zombies to access the same service at the same time, in order to overload the server. Next, we explain with most details the behavior of each of the primary DDoS attacks and how they are broadly classified in terms of automation degree, exploited the vulnerability, attack dynamics, and impact.

2.2.1 Classification by automation degree

The DDoS attacks are classified by the automation degree and can be divided into three main categories: manual, semi-automatic, and automatic attacks.

Manual attacks: to install malicious code, first, the attacker performs a vulnerability analysis over the devices connected to the Internet; next, the attackers execute a set of commands to start the attack [11, 58].

Semi-automatic attacks: the malicious agent follows a method called handler[4] agent, where the attacker performs a search in one or more handlers, looking for vulnerable machines to compose its zombie army. In the next step, the

[3] A zombie, robots, bots, daemon, or agent is a device connected to a network that a remote attacker has accessed and infected (set up) to command, controller, and dissemination, e.g., viruses, spam, and DDoS attacks [19].

[4] Compromised host with a Malware installed that can control many other computers [75].

attacker sets the configurations, which are the victim address and the attack initialization command. Semi-automatic attacks can be subdivided into two main groups: direct and indirect attack [58, 76]. In the direct attack, the malicious agent sends a command to the zombies to perform an attack on a victim; there is a high possibility to detect the attacker in this approach [9]. On the other hand, in indirect attacks, the malicious agent sends information to zombies that communicate with reflectors[5] informing the victim address IP; all steps are executed through command and control (scripts pre-defined). The IRC[6] channels are used as in-direct communication attacks, which makes it harder to track an attacker [34, 35, 86].

Automatic attacks: the steps are automatized. Moreover, the attacker uses a backdoor to access vulnerable machines [91]. This attack-type requires lower attacker exposition [80]; the attacker sends the command only once to control and accomplish an attack [65].

2.2.2 Classification by exploited vulnerability

The coordinated attacks by exploited vulnerabilities can be subdivided into four categories, flood attacks, amplification attacks, protocol exploitation attacks, and application attacks.

Flood attacks: is given through either sending a massive traffic amount to a particular server or service by exhausting all its resources; However, the attack type could fit in two distinct ways, i.e., packet send massive through User Datagram Protocol (UDP) and Internet Control Message Protocol (ICMP) [92, 94].

Amplification attacks: amplification attacks can be made through reflectors, and amplifications. In the first technique, the malicious agent utilizes a computer set to attack the victim through Internet Protocol (IP) address spoofing. In this case, the attacker may infect a machine, then forges the source IP address of a packet, making the reflectors reply to the victim address IP. On the other hand, amplification uses zombies machines; this intrusion is performed through a set of techniques, where an attacker infects computers through Malware (e.g., viruses). Among the attacks, we evaluated

[5] The reflectors are non-compromised systems that only send replies to a request [32].

[6] Internet Relay Chat (IRC) is an application layer protocol that facilitates communication in the form of text [53].

two protocol types used for invasion, which are Network Time Protocol (NTP) and Simple Network Management Protocol (SNMP) [38, 93].

Protocol exploration attacks: the type of attack that explores resource-specific or protocol implementation bug, e.g., SYN/ACK attacks, where the invader starts a connection, sends an SYN (synchronize) to the server, then the server responds with SYN+ACK and waits for ACK (acknowledgment) [8, 71].

2.2.3 Classification by an attack dynamic

The dynamic classification attacks can be divided into two steps, constant and variable offenses, which are described below.

Constant: These attacks have a constant rate. Also, after the onset of command and control, the agent's machines achieve attack packets at a steady rate [58, 97].

Variable: The variable attacks change the attack rate using several zombies machines. A victim's service could degrade slowly over a long period, causing performance loss and thus substantially delaying attack detection. Also, the attack rate adjusts based on the victim's behavior, or command and control defined [47, 90].

2.2.4 Classification by impact

The impact classification of attacks has two definitions attacks, disruptive, and degrading. Both can cause service performance loss. Next, we explain each attack.

Disruptive: The goal is to deny the clients access to the victim's service [58]. Furthermore, disruptive attacks need a coordinated response system; there are many communications points between the attacker and agents for coordinating the attack [21, 58].

Degrading: This attack-type exhausts the victim resource a few so that the capability will be interrupted slowly [41, 58].

2.3 IoT Devices Security

The Internet of Things (IoT) won the spotlight in recent years supported by the increase in the amount of non-traditional devices, such as industrial machinery, health equipment, and appliances connected to the Internet [50, 85]. As a result, the Internet has grown from a research tool for academic purpose only to an essential asset for everyday life, including water, electricity, and gas facilities [39, 89]. However, as a source of valuable resources, it soon became a target to criminals that seek to use that technology for illicit purpose or to deny the legitimate use of these resources to their rightful owners. The Internet context allowed resources to be attacked from anywhere, making cybersecurity a certain item. Usually, computer security is related to three main themes [3, 26, 96].

Confidentiality: This term concerns the ability of a system to be trusted, which means there is no disclosure and unauthorized access to data and information — maintaining confidentiality and security [4, 50].

Authentication: It consists of verifying if the data has not tampered, and check if a claimed author is whom he claims to be [12, 49].

Access: It guarantees that only authorized users can access data, infrastructure, resources, and information [50, 95].

Nevertheless, it is not difficult to find out vulnerable IoT devices, based only on a single Google search. The malicious users can discover the dork[7] and easily apply an SQL injection to websites, confidential files, and other methods. Currently, the vulnerabilities of IoT devices are used to orchestrate DDoS attacks through tools such as the Mirai botnet [2, 39, 52].

3 Motivation and Application of an Attack Tree

The attack trees (AT) model assists in finding the causes that take the attack success, aiming to support the designer, reports, description, and details of each attack. However, the attack methods change over time; even so, the techniques remain the same. Below we described each attack technique and their descriptions.

1. *Define the analysis scope:* The designer defines metrics, characteristics, and the environment behavior to perform the system's analysis and evaluation;
2. *System and functions understanding:* This step is characterized as the preliminary phase of an AT construction and requires an understanding of the system's characteristics and activities. This aspect demands the team cooperation comprising the safeness, reliability, and availability analysis to create the system's logical and functional structure;

[7] The Google dorks, sometimes just referred to as a dork, a quest that combines characters and advanced search operators to information find that is not readily available on a website [44].

3. *Attack success definition:* Attack Success (AS) events are characterized as the occurrence of an action, whose circumstance may lead to unsafe operating context, failures, or malfunction. Also, if more than one AS occurs it is necessary to investigate a different attack tree, with the others AS or it is necessary to accomplish a join of these two trees;
4. *Construction of the Attack Tree:* This phase consists of AT elaboration, where we observed and identifying; the computational environment and the likely failures; even so, most failures have modes and effects different. Where, through AT, we can represent these environments and threats easily, assigning logic gates to describe the logic behavior, in subsection (3.1), we explain the step-by-step to construct an Attack Tree;
5. *Qualitative analysis:* The Qualitative Analysis gives a logical expression of the attack success. This analysis enumerates all attack modes whose combinations cause the attack success to occur and locate the potential weak structure points. In subsection (3.2), we explain in detail;
6. *Quantitative analysis:* The Quantitative Analysis measures the occurrence likelihood of attack success and any intermediate levels. Also, this analysis may rank each event according to its leverage on the overall system measure. In section (3.3), we define in detail;
7. *Analysis result:* The results are shown in graphical forms; the available most AT tools provide facilities to accomplish this task. This section (4) is shown a practical example of how to model virtual threats scenarios Attack Tree using.

The failure modes and their effects on the AS are the challenging part of the task and depend mainly on the analyst skill or the group of experts involved in this process. The event sequences are central to this book-chapter topic [79, 78]. Indeed, the cybernetic threats analysis community seems to have brought forth much event-based formalism, i.e., cause-effect diagrams.

Even used to describe system-internal events, these can be adapted to current external attacker scenarios as well [23]. Steffan [77] compared the AT with attack nets, a threat study based on Petri Nets. In [54] described the union of security knowledge between attack net model and AT.

3.1 Construction of the Attack Tree

The nodes AT represent attacks, and the root node is the invader purpose. Each discovered node is recursively conducted as another AS and so on. However, the construction repeats some nodes aiming to identify nodes at a more exceptional resolution, until the detail desired level be reached.

Logic gates represent the communication between nodes through AT levels. Each logic gate represents the occurrence of a high-level node that receives two or more basic event as an input. These basic nodes are the final nodes of an AT and reflect the state of an elemental cause, whose deviation from the correct operation can affect the occurrence of the AS. Next, the attack tree concepts are presented.

The graphical attack tree construction is based on the representation of nodes and their relationship by using suitable symbols. Usually, logic gates represent the conditions to an event occur, which are presented as rectangles (root nodes), while the spheres represent leaf nodes. Figure 1 presents the most used symbols [42, 70].

3.1.1 The OR Logic Gate

The logic operator OR is denoted as the sum operator and shown through literature as the + or $\vee$ symbols. The OR gate output event is satisfactory when one or more action is true.

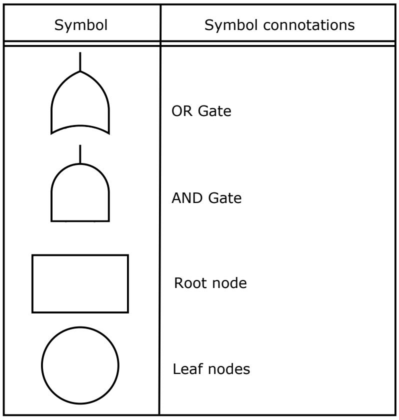


Fig. 1: The symbols used in AT construction.

Example 3.1. Consider a system S whose logical structure can be represented by the AT of Figure 2, where the $\text{Root}_{\text{node}} = \overline{S}$ is the system attack event and is the output of an OR gate whose inputs are $\overline{A}$ (attack of component A), $\overline{B}$ (attack of component B), $\overline{C}$ (invasion of component C) and $\overline{D}$ (attack of component D). The top event $\mathrm{Root}_{\mathrm{node}} = \overline{S}$ occurs if one or more of the input events occur, the algebraic form to represent this AT is $\mathrm{Root}_{\mathrm{node}}$, $\overline{S} = \overline{A} \vee \overline{B} \vee \overline{C} \vee \overline{D}$.

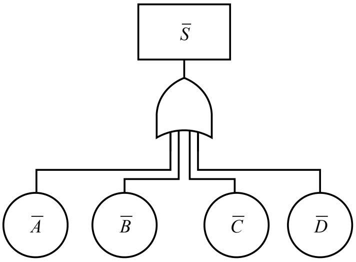


Fig. 2: Graphical representation of an AT using OR logic gates.

3.1.2 The AND Logic Gate

The logical operator AND denotes a multiplication logic ($\cdot$ or $\wedge$). The output events of an AND gate its valid *if and the only if* all the input events are jointly satisfied.

Example 3.2. Consider an S system whose your representation has a logic structure that can be represented by the AT at Figure 3, where the $Root_{node} = \overline{S}$ is the attack success, and the output events are by AND gates $\overline{A}$, $\overline{B}$, $\overline{C}$, and $\overline{D}$. The $Root_{node}$ only occurs when all the components are attacked, the algebraic form for representing is $Root_{node}\overline{S} = \overline{A} \wedge \overline{B} \wedge \overline{C} \wedge \overline{D}$.

3.1.3 Attack Tree with OR and AND Gates

The attack trees can represent complex systems under attack. However, it can be many connected components, and through them many redundant settings that may have both OR and AND gates in the same tree.

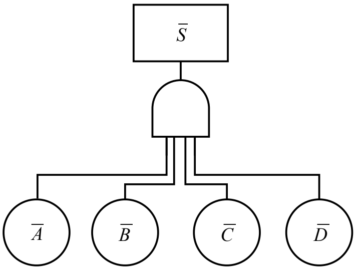


Fig. 3: Graphical representation of an AT using AND logic gates.

Example 3.3. Consider an S system where attacks can occur, the leaf nodes represent these attacks, the nodes $\overline{A}$ and $\overline{B}$ belong to an AND gate, while the $\overline{C}$ belongs to an OR gate, to arrive root node, the condition G_1 or $\overline{C}$ must be satisfied. Figure 4 show this representation.

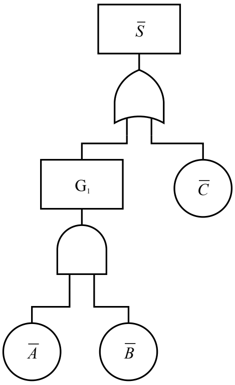


Fig. 4: Graphical representation an Attack Tree using OR and AND logic gates.

3.2 Qualitative analysis

The qualitative analysis is a logic evaluation of event combinations that can result in the occurrence of one top event. This analysis involves the logic expression derivation [7, 24, 29, 30]. These two methods for deriving the logical expression, top-down, and bottom-up:

Top-down: It is an evaluation method it starts from the $\text{Root}_{\text{node}}$ and expands the logical expression in all basic events until it reached all leaf nodes. An evaluation method starts from the $\text{Root}_{\text{node}}$ and expands the logical expression in all basic events until it reached all leaf nodes. Also, this method features function from top events expressed by with minimal cut set (MCS) [48, 81].

Bottom-up: In the bottom-up method, the logical expression is directly obtained through a recursive search along the AT, with no need for a preliminary search for the minimal cut sets [10, 73].

3.2.1 Cut and Minimal Cut Sets

The cuts are a failure set that, if occur, implies the occurrence of the top event, and the minimal reduction set cannot be reduced without losing status as a cut set [15, 48, 82]. The cuts set and minimal cuts set determination interactively in a top-down manner, starting from the $\text{Root}_{\text{node}}$ and applying the Boolean algebra rules [88] until all the terminal nodes are reached [28].

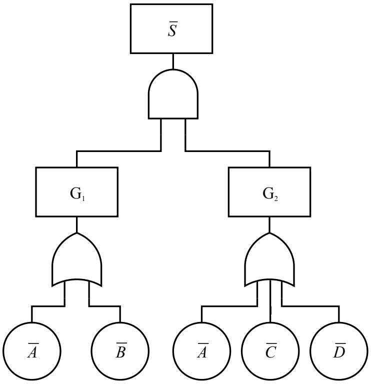


Fig. 5: Attack Tree with two-level

Example 3.4. At the OR gate of Figure 4, in the occurrence, any single attack $\bar{A}, \bar{B}, \bar{C}$ or $\bar{D}$ leads to the $\text{Root}_{\text{node}}$, so each attack is a system MCS.

$$\text{Root}_{\text{node}} = \overline{A} \vee \overline{B} \vee \overline{C} \vee \overline{D}. \tag{1}$$

Example 3.5. Already the Figure 5, with AND gate, if only a attack occur $\bar{A}, \bar{B}, \bar{C}$ or $\bar{D}$ leading to the $\text{Root}_{\text{node}}$, thus, so each attack is a system MCS.

$$\text{Root}_{\text{node}} = \overline{A} \wedge \overline{B} \wedge \overline{C} \wedge \overline{D}. \tag{2}$$

The cuts set of Figure 5 is shown in Eq. (3), and the minimal cuts set is shown in Eq. (4). With one cuts set 1st order and 2nd-second order. Where the smaller the order, the more critical is the cut.

$$\text{Root}_{\text{node}} = (\bar{A}, \bar{A}); (\bar{A}, \bar{C}); (\bar{A}, \bar{D}); (\bar{B}, \bar{C}); (\bar{B}, \bar{D}). \tag{3}$$

$$\text{Root}_{\text{node}} = (\bar{A}); (\bar{B}, \bar{C}); (\bar{B}, \bar{D}). \tag{4}$$

The cuts and minimal cuts assist in the identification of the attacks with occurrence likelihood more; it is challenging to find reductions in a tree with many leaves, but there tools, SecurITree, Isograph, and SeaMonster that can support the cuts recognition [37, 56, 84]. At the next step, we evaluated the quantitative analysis to get the metrics values for the security model.

3.3 Quantitative analysis

The quantitative analysis is represented by aspect or relevant properties in security models, i.e., attributes sometimes called metrics, including the likelihood of an attack, impact, attacker benefit, feasibility, and pain factor. In this section, we will address each aspect mentioned.

The attack success likelihood an AND gate will be obtained by Eq. (5), and attack an OR gate will be calculated by Eq. (6). Also, equations are defined in the interval [0, 1]. n represents the tree leaf nodes number, and attack likelihood in an instant i; (A_i) describes the event success [22].

$$P_{\text{AND}} = \prod_{i=1}^{n} A_i \tag{5}$$

$$P_{\text{OR}} = 1 - \prod_{i=1}^{n} (1 - A_i) \tag{6}$$

Example 3.6. Consider an S system where can occur several attacks in leaf nodes; we assign the values for $\overline{A}$, $\overline{B}$, $\overline{C}$, $\overline{D}$, $\overline{E}$. Below we explained as to get the result shown Figure 6. Through gate OR we get the occurrence $G_1 = 0.3 \times 0.5$, and $G_2 = (1 - (1 - 0.7) \times (1 - 0.9) \times (1 - 0.6))$. They were obtained through gate AND, and the sub-leaf $\overline{S} = (1 - (1 - 0.65) \times (1 - 0.378))$ was get with gate OR. With that we were able to find the attack success probability $\overline{S} = 0.7823$.

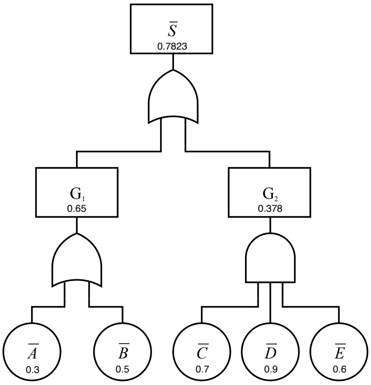


Fig. 6: Example an Attack Tree with likelihood metrics.

Table 2 shows the descriptions to represent all possible impacts in any system [22]. We use this table for attacks impact verification. Next, we will show an example of [40].

Numerical Range	Impact	Impact definition
$1 \leq I < 4$	**None**	**Minor impact on the system.**
$4 \leq I < 7$	**Low**	**A moderate impact on the system.**
$7 \leq I < 10$	**Average**	**Significant damage results to the system.**
10	**High**	**The system completely compromised, inoperable, or destroyed.**

Table 2: System impact

The attack impact through AND gate can be found by Eq. (7), where the range numerical exposed in Table (2) shows the variability of each attack, where I_i represents the attack impact, n represents the number of the tree leaf nodes.

$$\frac{10^n - \prod_{i=1}^{n}(10 - I_i)}{10^{(n-1)}} \tag{7}$$

The attack impact through the OR gate can be found by Eq. (8), where the range numerical exposed in Table (2) shows the variability each attack, where

Max_I represent the maximum attack impact, n represents the number of the tree leaf nodes.

$$\mathrm{Max}_{i=1}^{n} I_i \tag{8}$$

Example 3.7. Consider an S system as in Figure 7 where can occur several attacks in leaf nodes and can impact on system availability. Through Eq. (7) the gate AND, and Eq. (8). We assign value this impacts, extracted of Table (2), we find the results for $\overline{A}$, $\overline{B}$, $\overline{C}$, $\overline{D}$, $\overline{E}$. Given this context we can find the impacts; Through gate OR we get the occurrence $G_1 = 7$, i.e., impact maximum, and through gate AND we get $G_2 = 10 - ((10-7) \times (10-5) \times (10-9)) \div 10$, and $S = 10 - ((10-7) \times (10-8.5)) \div 10$, i.e., the impact $Root_{\text{node}}$ will be $\overline{S} = 9.55$.

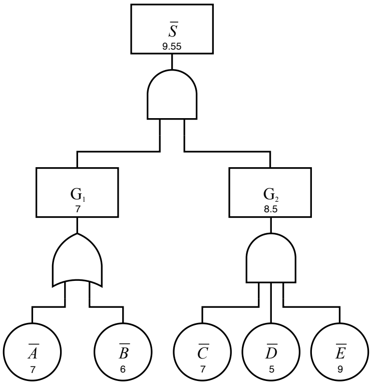


Fig. 7: Example an Attack Tree with impact metrics.

Table (3) shows the technical complexity of an attack implementation in TCI_i that varies between $[10 \leq 100]$. Where T_i is a technical ability that varies in number between [0, 1], we use the table values for the attacks variability verifying and for construct the profiles that will use for modeling the attack scenarios [16, 37, 60].

$$T_i = \frac{1}{TCI_i} \tag{9}$$

TCI_i	Agent knowledge	Knowledge definition
$10 - 30$	**Average user**	**Can easily surf the Internet, less system knowledge.**
$40 - 60$	**Power-user**	**Low programming skills.**
$70 - 90$	**Advanced user**	**IT security specialist.**
≤ 100	**Experienced user**	**Master in computational security, knowledgeable of tools and hacking practices.**

Table 3: Technical ability

3.3.1 Profiles definition

This subsection will address agent and attacker profiles. There are four profiles, two to the attacker evaluation, and, other two to victims' evaluation. We used these profiles to draw the victim and attacker behavior according to the attack tree. Also, we use to find the interest metrics, attacker benefit, feasibility, and pain factor. The profiles that we will use our technical ability, occurrence attack likelihood, noticeability, operational losses, and reputation loss [36, 37, 70, 83]. The obtained values through Tables (2) and (3), help us in profiles construction, next we explain each profile presented.

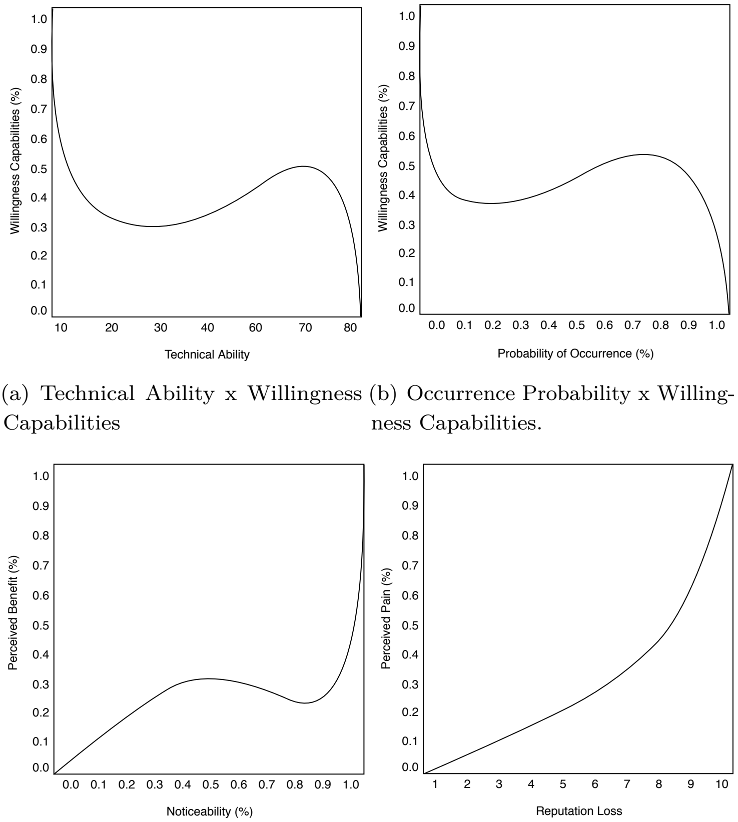
Then, with a list of IoT devices vulnerable, the infects attacker the IoT devices with Malware, next choose a victim to attack. We use IoT devices' existing vulnerabilities to represent the real-world attacks. These values were that we extracted publications, reports, and Common Vulnerabilities and Exposures (CVE). Also, each vulnerability and metric as described in Table 4.

The technical ability profile described in Figure 8(a) was developed to evaluate the attacker's technical ability. Through Eq. (9) We find the values correspondent to describe profile. The y-axis (Willingness Capabilities) brings values between [0, 1], and x-axis show Table (3) obtained values [18, 69].

The occurrence probability profile in Figure 8(b) was build using values obtained from Eq. (5) and (6) to design the profile behavioral pattern. Where, the y-axis (Willingness Capabilities) and x-axis have values between [0, 1] [98, 14].

The noticeability profile described in Figure 8(c) shows how much victim is visible to virtual threats. The y-axis (Perceived Pain) and x-axis values vary between [0, 1], i.e., the occur likelihood is interconnected with victim visibility [46].

The reputation loss profile character in Figure 8(d) was obtained through Table (2) and Eq. (7), and (8) to the attack impact assess, where we find the perceived pain correlating the x-axis with the y-axis, the profile gives us the victim reputation level [13, 31, 64].



(a) Technical Ability x Willingness Capabilities

(b) Occurrence Probability x Willingness Capabilities.

(c) Noticeability x Perceived Pain.

(d) Reputation Loss x Perceived Pain.

Fig. 8: Profiles.

Example 3.8. Consider an S system where the attacker has a technical ability 70, 60, 70, 50, and 90. We a draw line in Figure 10 to demonstrate that the attacker will have an 8% opportunity according to the profile curve. We use Eq. (9) for likelihood calculate of the attacker technical ability corresponding the Figure 9 attack tree.

This subsection presented the victim and attacker profiles and how to evaluate the impact of an attack by using an AT. The next section presents two study cases applied to IoT devices vulnerabilities.

Table 4: Input parameters referents of attack

Threats	$Prob_i$	TCI_i	Rep_i
Phishing	**0.022**	**50**	**Average**
Bribe	**0.032**	**80**	**Average**
Social Engineering	**0.059**	**70**	**High**
Poor Configuration	**0.088**	**40**	**High**
Sendmail Exploit	**0.088**	**40**	**High**
Steal Password	**0.065**	**60**	**High**
Sniff Network	**0.088**	**70**	**High**
Hijacking	**0.065**	**60**	**High**
Misuse	**0.065**	**50**	**None**
Keylogger	**0.078**	**60**	**High**
Violation of organization political	**0.055**	**50**	**Average**
Zeus	**0.075**	**60**	**Average**
Revealer	**0.044**	**70**	**Low**
Mirai	**0.088**	**50**	**High**
Pushdo	**0.054**	**50**	**Low**

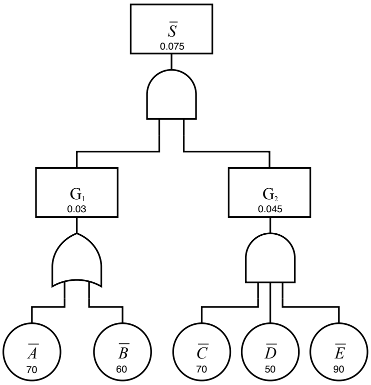


Fig. 9: A technical ability example in AT.

4 Case Studies

In this case study, we evaluated an invasion scenario with DDoS and Malware in IoT devices. This section broaches an attack tree model used to represent the scheme of the invasion proposed in Figure 12. Next, we explain each followed step.

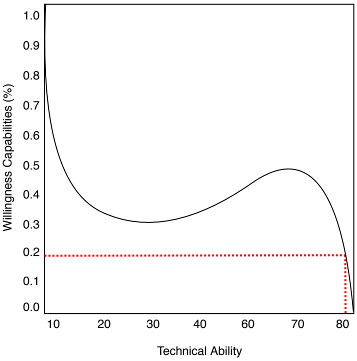


Fig. 10: A technical ability profile example.

The proposed setting, as described in Figure 12. The steps to find the IoT devices, and create a network of bots follow this logic. The intruder utilizes handlers to perform research in anonymous mode.

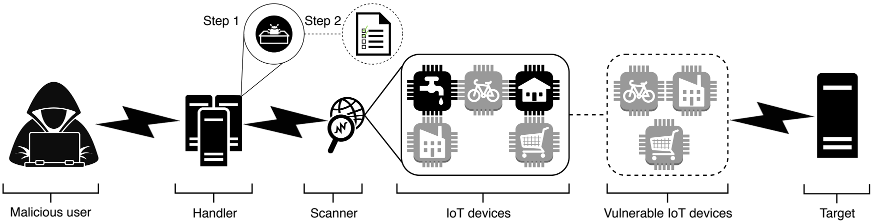
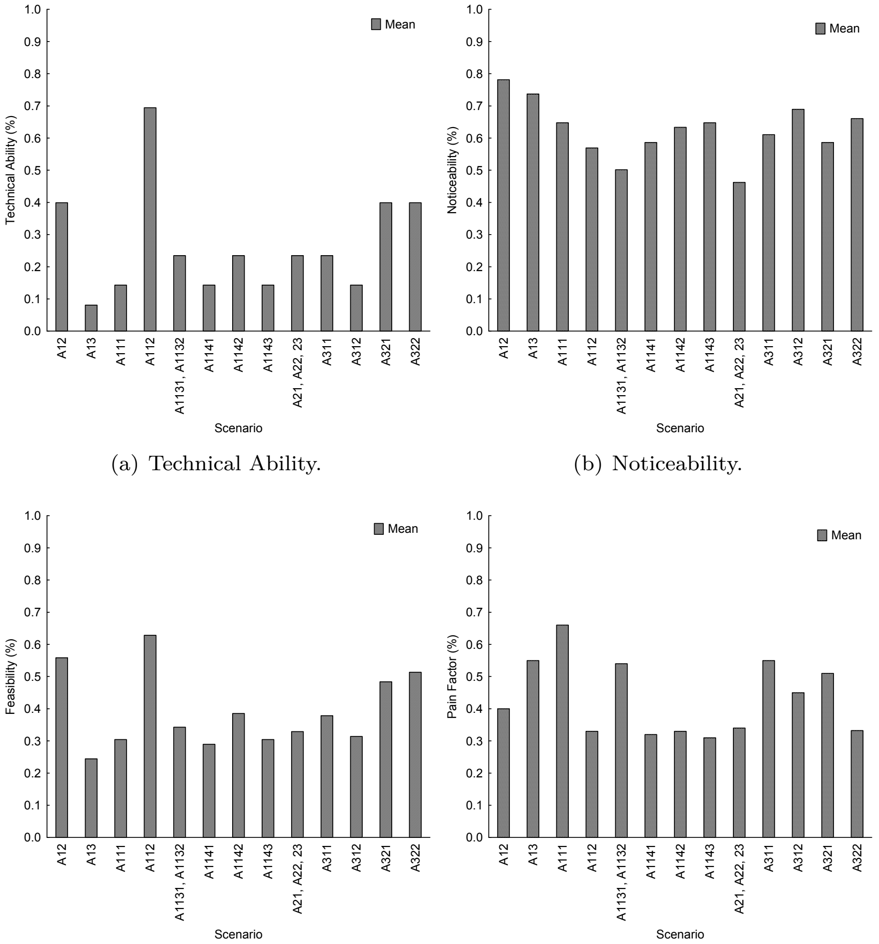


Fig. 12: Scenario of possible attacks.

The obtained results show us the importance of the evaluated metrics; we model the attack tree according to Figure 13 adopt tool SecurITree of Amenaza enterprise [36]. Next, we explain each result; the attack methods were transcribed according to Listing 1. We obtained the attack likelihood 0.51 attack chance. Also, the result in Figure 11(a) shows the technical ability of intruder, observing the result, we can infer that there 0.7 occur chances an attack-type A112, the Figure 11(b) noticeability bring relative values the victim exposition to a particular threat, the result shows us 0.78 of opportunity applying A12.



(a) Technical Ability.

(b) Noticeability.

(c) Feasibility.

(d) Pain Factor.

Fig. 11: Proposed evaluation result.

Figure 11(c) feasibility is relative to the attack ease, where can be more accessible to the attacker, the attack more feasible is the technique A112 with 0.65 of occurring chance and A12 in second place with 0.55 likelihood occur. Figure 11(d) portrays pain factor the attack methods that, if it happens, can problems beget the victim, i.e., service downtime, more costs, and so on, the values obtained show us 0.65 chance using the technique A111.

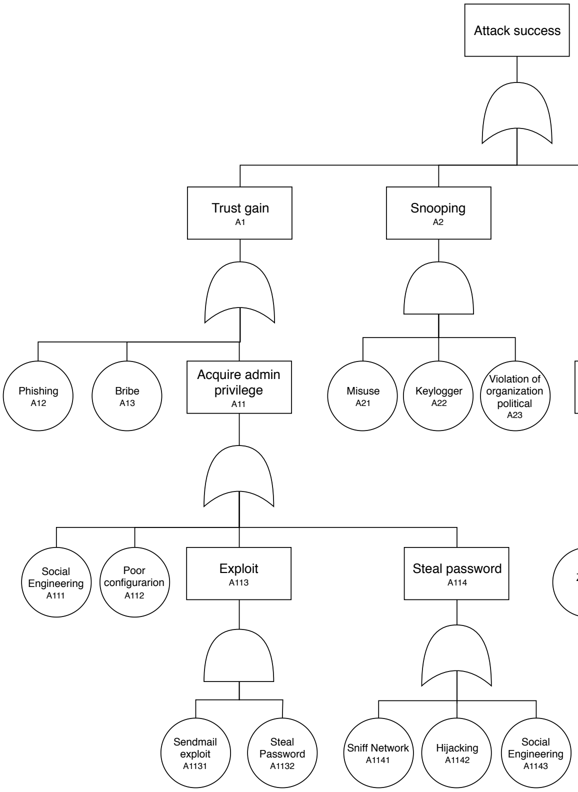


Fig. 13: Attack Tree

This section presented a DDoS and Malware attack scenario using IoT devices and show the impact of these attacks on a computational infrastructure.

5 Final Remarks

In this chapter, an analysis was presented to evaluate the DDoS and Malware attacks impact using IoT devices. Thereby, we developed a case study to demonstrate the applicability of attack trees on security scenarios. In this context, we obtained; the feasibility, attack likelihood, pain factor, technical ability, and noticeability. These metrics are essential to measuring the total threat capacity. Moreover, this chapter brings a broad understanding as far as it is concerned about the virtual threats and impacts.

Listing 1: Stem and leaf of AT

```
A1: Trust gain (OR)
    A12: Phishing
    A13:Bribe
    A11: Acquire admin privilege (OR)
        A111:Social Engineering
        A112: Poor configuration
        A113: Exploit (AND)
            A1131: Sendmail exploit
            A1132: Steal Password
        A114: Steal password (OR)
            A1141: Sniff Network
            A1141: Hijacking
            A1141: Social Engineering
A2: Snooping (AND)
    A21: Misuse
    A22: Keylogger
    A23: Violation Of organization political
A3: Software attack (OR)
    A31: Trojan Horse (OR)
        A311: Zeus
        A312: Revealer
    A32: Botnet (OR)
        A321: Mirai
        A322: Pushdo
```

Acknowledgment: We want to thank the MoDCS Research Group for their support and would like to thank Terry Ingoldsby for making accessible SecurITree tool.

References

[1] L. Abrams. Dramatic increase of ddos attack sizes attributed to iot devices. https://www.bleepingcomputer.com/news/security/dramatic-increase-of-ddos-attack-sizes-attributed-to-iot-devices/, 2018.

[2] M. Antonakakis, T. April, M. Bailey, M. Bernhard, E. Bursztein, J. Cochran,

Z. Durumeric, J. A. Halderman, L. Invernizzi, M. Kallitsis, et al. Understanding the mirai botnet. In *26th {USENIX} Security Symposium ({USENIX} Security 17)*, pages 1093–1110, 2017.

[3] A. Avizienis, J. Laprie, and B. Randell. Fundamental Concepts of Dependability. *Technical Report Series-University of Newcastle upon Tyne Computing Science*, 2001.

[4] A. Avižienis, J. Laprie, B. Randell, and U. of Newcastle upon Tyne. Computing Science. *Fundamental Concepts of Dependability*. Technical report series. University of Newcastle upon Tyne, Computing Science, 2001.

[5] G. Avoine, P. Junod, and P. Oechslin. *Computer system security: basic concepts and solved exercises*. EPFL Press, 2007.

[6] J. Aycock. *Computer viruses and malware*, volume 22. Springer Science & Business Media, 2006.

[7] A. A. Baig, R. Ruzli, and A. B. Buang. Reliability analysis using fault tree analysis: a review. *International Journal of Chemical Engineering and Applications*, 4(3):169, 2013.

[8] S. M. Bellovin. Security problems in the tcp/ip protocol suite. *ACM SIGCOMM Computer Communication Review*, 19(2):32–48, 1989.

[9] M. H. Bhuyan, H. J. Kashyap, D. K. Bhattacharyya, and J. K. Kalita. Detecting distributed denial of service attacks: methods, tools and future directions. *The Computer Journal*, 57(4):537–556, 2013.

[10] A. Bobbio, L. Portinale, M. Minichino, and E. Ciancamerla. Improving the analysis of dependable systems by mapping fault trees into bayesian networks. *Reliability Engineering & System Safety*, 71(3):249–260, 2001.

[11] R. Braga, E. de Souza Mota, and A. Passito. Lightweight ddos flooding attack detection using nox/openflow. In *LCN*, volume 10, pages 408–415, 2010.

[12] K. Brown. Three-legacy mode payment card with parametric authentication and data input elements, Sept. 9 2004. US Patent App. 10/800,821.

[13] E. Byres and J. Lowe. The myths and facts behind cyber security risks for industrial control systems. In *Proceedings of the VDE Kongress*, volume 116, pages 213–218, 2004.

[14] Q. Chen and J. D. McCalley. Identifying high risk nk contingencies for online security assessment. *IEEE Transactions on Power Systems*, 20(2):823–834, 2005.

[15] X. Chen, H. Ren, and C. Bil. Fault tree analysis for composite structural damages. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, 228(9):1466–1474, 2014.

[16] B. H. Cheng, P. Sawyer, N. Bencomo, and J. Whittle. A goal-based modeling approach to develop requirements of an adaptive system with environmental

uncertainty. In *International Conference on Model Driven Engineering Languages and Systems*, pages 468–483. Springer, 2009.

[17] M. Christodorescu, S. Jha, S. A. Seshia, D. Song, and R. E. Bryant. Semantics-aware malware detection. In *2005 IEEE Symposium on Security and Privacy (S&P'05)*, pages 32–46. IEEE, 2005.

[18] A. Conklin. Cyber defense competitions and information security education: An active learning solution for a capstone course. In *Proceedings of the 39th Annual Hawaii International Conference on System Sciences (HICSS'06)*, volume 9, pages 220b–220b. IEEE, 2006.

[19] E. Cooke, F. Jahanian, and D. McPherson. The zombie roundup: Understanding, detecting, and disrupting botnets. *SRUTI*, 5:6–6, 2005.

[20] A. corporation. Risk techniques, fault tree. https://www.arescorporation.com.

[21] C. Douligeris and A. Mitrokotsa. Ddos attacks and defense mechanisms: classification and state-of-the-art. *Computer Networks*, 44(5):643–666, 2004.

[22] K. S. Edge, G. C. Dalton, R. A. Raines, and R. F. Mills. Using attack and protection trees to analyze threats and defenses to homeland security. In *Military Communications Conference, 2006. MILCOM 2006. IEEE*, pages 1–7. IEEE, 2006.

[23] S. Einarsson and M. Rausand. An approach to vulnerability analysis of complex industrial systems. *Risk analysis*, 18(5):535–546, 1998.

[24] C. A. Ericson. Fault tree analysis. In *System Safety Conference, Orlando, Florida*, pages 1–9, 1999.

[25] K. Eugene. A brief history of ddos attacks. https://eugene.kaspersky.com/2016/12/06/a-brief-history-of-ddos-attacks/.

[26] M. Fahmideh and D. Zowghi. An exploration of IoT platform development. *Information Systems*, 87:101409, 2020.

[27] B. Felter. 5 of the most famous recent ddos attacks. https://www.vxchnge.com/blog/recent-ddos-attacks-on-companies, 2019.

[28] I. N. Fovino, M. Masera, and A. De Cian. Integrating cyber attacks within fault trees. *Reliability Engineering & System Safety*, 94(9):1394–1402, 2009.

[29] J. Fussell. Review of fault tree analysis with emphasis on limitations. Technical report, Aerojet Nuclear Co., Idaho Falls, Idaho (USA), 1975.

[30] J. Fussell and W. Vesely. New methodology for obtaining cut sets for fault trees. *Trans. Amer. Nucl. Soc*, 15(1):262–263, 1972.

[31] A. Garg, J. Curtis, and H. Halper. Quantifying the financial impact of it security breaches. *Information Management & Computer Security*, 11(2):74–83, 2003.

[32] P. Gasti, G. Tsudik, E. Uzun, and L. Zhang. Dos and ddos in named

data networking. In *2013 22nd International Conference on Computer Communication and Networks (ICCCN)*, pages 1–7. IEEE, 2013.

[33] V. D. Gligor. A note on denial-of-service in operating systems. *IEEE Transactions on Software Engineering*, 10(3):320–324, 1984.

[34] J. Goebel and T. Holz. Rishi: Identify bot contaminated hosts by irc nickname evaluation. *HotBots*, 7:8–8, 2007.

[35] A. Hussain, J. Heidemann, J. Heidemann, and C. Papadopoulos. A framework for classifying denial of service attacks. In *Proceedings of the 2003 conference on Applications, technologies, architectures, and protocols for computer communications*, pages 99–110. ACM, 2003.

[36] T. R. Ingoldsby. Amenaza demos. *Amenaza Technologies Limited*, 2010.

[37] T. R. Ingoldsby. Attack tree-based threat risk analysis. *Amenaza Technologies Limited*, pages 3–9, 2010.

[38] X. Jing, J. Zhao, Q. Zheng, Z. Yan, and W. Pedrycz. A reversible sketch-based method for detecting and mitigating amplification attacks. *Journal of Network and Computer Applications*, 2019.

[39] C. Kolias, G. Kambourakis, A. Stavrou, and J. Voas. Ddos in the iot: Mirai and other botnets. *Computer*, 50(7):80–84, 2017.

[40] M. Kührer, T. Hupperich, C. Rossow, and T. Holz. Exit from hell? reducing the impact of amplification ddos attacks. In *23rd {USENIX} Security Symposium ({USENIX} Security 14)*, pages 111–125, 2014.

[41] K. Kumar, R. Joshi, and K. Singh. A distributed approach using entropy to detect ddos attacks in isp domain. In *2007 International Conference on Signal Processing, Communications and Networking*, pages 331–337. IEEE, 2007.

[42] R. Kumar, S. Schivo, E. Ruijters, B. M. Yildiz, D. Huistra, J. Brandt, A. Rensink, and M. Stoelinga. Effective analysis of attack trees: A model-driven approach. In *International Conference on Fundamental Approaches to Software Engineering*, pages 56–73. Springer, Cham, 2018.

[43] O. Kupreev, E. Badovskaya, and A. Gutnikov. Ddos attacks in q4 2018. https://securelist.com/ddos-attacks-in-q4-2018/89565/, 2019.

[44] L. Lancor and R. Workman. Using google hacking to enhance defense strategies. *SIGCSE Bull.*, 39(1):491–495, Mar. 2007.

[45] F. Lau, S. H. Rubin, M. H. Smith, and L. Trajkovic. Distributed denial of service attacks. In *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0*, volume 3, pages 2275–2280. IEEE, 2000.

[46] J. Lee, L. Bauer, and M. L. Mazurek. The effectiveness of security images in internet banking. *IEEE Internet Computing*, 19(1):54–62, 2014.

[47] K. Lee, J. Kim, K. H. Kwon, Y. Han, and S. Kim. Ddos attack detection method using cluster analysis. *Expert systems with applications*, 34(3):1659–1665, 2008.

[48] W.-S. Lee, D. L. Grosh, F. A. Tillman, and C. H. Lie. Fault tree analysis, methods, and applications a review. *IEEE transactions on reliability*, 34(3):194–203, 1985.

[49] D. D.-H. Lin, A. A. Shaheen, and K. K. Yellepeddy. Multiple remote data access security mechanism for multitiered internet computer networks, Apr. 18 2000. US Patent 6,052,785.

[50] H. Lin and N. Bergmann. Iot privacy and security challenges for smart home environments. *Information*, 7(3):44, 2016.

[51] R. Maciel, J. Araujo, J. Dantas, C. Melo, E. Guedes, and P. Maciel. Impact of a ddos attack on computer systems: An approach based on an attack tree model. In *2018 Annual IEEE International Systems Conference (SysCon)*, pages 1–8. IEEE, 2018.

[52] MalwareMustDie. Mmd-0055-2016 - linux/pnscan ; elf worm that still circles around. http://blog.malwaremustdie.org/2016/08/mmd-0054-2016-pnscan-elf-worm-that.html.

[53] C. Mazzariello. Irc traffic analysis for botnet detection. In *2008 The Fourth International Conference on Information Assurance and Security*, pages 318–323. Ieee, 2008.

[54] J. P. McDermott. Attack net penetration testing. In *NSPW*, pages 15–21, 2000.

[55] G. McGraw and G. Morrisett. Attacking malicious code: A report to the infosec research council. *IEEE software*, 17(5):33–41, 2000.

[56] P. H. Meland, D. G. Spampinato, E. Hagen, E. T. Baadshaug, K.-M. Krister, and K. S. Velle. Seamonster: Providing tool support for security modeling. *Norsk informasjonssikkerhetskonferanse, NISK*, 2008.

[57] N. Milošević. History of malware. *arXiv preprint arXiv:1302.5392*, 2013.

[58] J. Mirkovic and P. Reiher. A taxonomy of ddos attack and ddos defense mechanisms. *ACM SIGCOMM Computer Communication Review*, 34(2):39–53, 2004.

[59] M. Moore. Ddos attacks could cost the uk £1bn this year. https://www.techradar.com/news/ddos-attacks-could-cost-the-uk-pound1bn-this-year, 2019.

[60] D. Mougouei, W. Rahman, and M. M. Almasi. Measuring security of web services in requirement engineering phase. *International Journal of Cyber-Security and Digital Forensics (IJCSDF)*, 1(2):89–98, 2012.

[61] B. Nelson, A. Phillips, and C. Steuart. *Guide to computer forensics and investigations.* Cengage learning, 2014.

[62] Neustar. Worldwide ddos attacks e cyber insights research report. *A Neustar Security Solutions Exclusive*, pages 1–52, 2017.

[63] H. Orman. The morris worm: A fifteen-year perspective. *IEEE Security & Privacy*, 1(5):35–43, 2003.

[64] K. Popović and Ž. Hocenski. Cloud computing security issues and challenges. In *The 33rd International Convention MIPRO*, pages 344–349. IEEE, 2010.

[65] R. Power. *2002 CSI/FBI computer crime and security survey.* Computer Security Institute, 2002.

[66] A. Qamar, A. Karim, and V. Chang. Mobile malware attacks: Review, taxonomy & future directions. *Future Generation Computer Systems*, 97:887–909, 2019.

[67] Y. Radai. The israeli pc virus. *Computers & Security*, 8(2):111–113, 1989.

[68] A. D. Rayome. Major ddos attacks increased 967% this year. https://www.techrepublic.com/article/major-ddos-attacks-increased-967-this-year/, 2019.

[69] K. Scarfone, M. Souppaya, A. Cody, and A. Orebaugh. Technical guide to information security testing and assessment. *NIST Special Publication*, 800(115):2–25, 2008.

[70] B. Schneier. Attack trees. *Dr. Dobb's journal*, 24(12):21–29, 1999.

[71] C. L. Schuba, I. V. Krsul, M. G. Kuhn, E. H. Spafford, A. Sundaram, and D. Zamboni. Analysis of a denial of service attack on tcp. In *Security and Privacy, 1997. Proceedings., 1997 IEEE Symposium on*, pages 208–223. IEEE, 1997.

[72] M. Sikorski and A. Honig. *Practical malware analysis: the hands-on guide to dissecting malicious software.* no starch press, 2012.

[73] R. M. Sinnamon and J. Andrews. New approaches to evaluating fault trees. *Reliability Engineering & System Safety*, 58(2):89–96, 1997.

[74] I. Smurf. Denial-of-service attacks. *Book Smurf IP Denial-of-Service Attacks, Series Smurf IP Denial-of-Service Attacks*, 1998.

[75] S. M. Specht and R. B. Lee. Distributed denial of service: Taxonomies of attacks, tools, and countermeasures. In *ISCA International Conference on Parallel and Distributed Computing (and Communications) Systems*, pages 543–550, 2004.

[76] A. Srivastava, B. Gupta, A. Tyagi, A. Sharma, and A. Mishra. A recent survey on ddos attacks and defense mechanisms. In *International Conference on Parallel Distributed Computing Technologies and Applications*, pages 570–580. Springer, 2011.

[77] J. Steffan and M. Schumacher. Collaborative attack modeling. In *Proceedings of the 2002 ACM symposium on Applied computing*, pages 253–259. ACM, 2002.

[78] L. P. Swiler and C. Phillips. A graph-based system for network-vulnerability analysis. Technical report, Sandia National Labs., Albuquerque, NM (United States), 1998.

[79] L. P. Swiler, C. Phillips, D. Ellis, and S. Chakerian. Computer-attack graph generation tool. In *Proceedings DARPA Information Survivability Conference and Exposition II. DISCEX'01*, volume 2, pages 307–321. IEEE, 2001.

[80] P. Szor. *The art of computer virus research and defense*. Pearson Education, 2005.

[81] H. Tanaka, L. Fan, F. Lai, and K. Toguchi. Fault-tree analysis by fuzzy probability. *IEEE Transactions on reliability*, 32(5):453–457, 1983.

[82] Z. Tang and J. B. Dugan. Minimal cut set/sequence generation for dynamic fault trees. In *Annual Symposium Reliability and Maintainability, 2004-RAMS*, pages 207–213. IEEE, 2004.

[83] T. Tidwell, R. Larson, K. Fitch, and J. Hale. Modeling internet attacks. In *Proceedings of the 2001 IEEE Workshop on Information Assurance and security*, volume 59. United States Military Academy West Point, NY, 2001.

[84] M. Walker and Y. Papadopoulos. Qualitative temporal analysis: Towards a full implementation of the fault tree handbook. *Control Engineering Practice*, 17(10):1115–1125, 2009.

[85] E. K. Wang, Z. Liang, C.-M. Chen, S. Kumari, and M. K. Khan. Porx: A reputation incentive scheme for blockchain consensus of iiot. *Future Generation Computer Systems*, 2019.

[86] Z. Wang. An elastic and resiliency defense against ddos attacks on the critical dns authoritative infrastructure. *Journal of Computer and System Sciences*, 99:1–26, 2019.

[87] H. Watson. Bell telephone laboratories. *Launch Control Safety Study," Bell Telephone Laboratories, Murray Hill, NJ USA*, 1961.

[88] J. E. Whitesitt. *Boolean algebra and its applications*. Courier Corporation, 2012.

[89] J. Wurm, K. Hoang, O. Arias, A.-R. Sadeghi, and Y. Jin. Security analysis on consumer and industrial iot devices. In *2016 21st Asia and South Pacific Design Automation Conference (ASP-DAC)*, pages 519–524. IEEE, 2016.

[90] Y. Xie and S.-Z. Yu. Monitoring the application-layer ddos attacks for popular websites. *IEEE/ACM Transactions on Networking (TON)*, 17(1):15–25, 2009.

[91] A. Yaar, A. Perrig, and D. Song. Siff: A stateless internet flow filter to mitigate ddos flooding attacks. In *IEEE Symposium on Security and Privacy, 2004. Proceedings. 2004*, pages 130–143. IEEE, 2004.

[92] Z. Yan, J. Liu, L. T. Yang, and W. Pedrycz. Data fusion in heterogeneous networks. *Information Fusion*, 53:1 – 3, 2020.

[93] Y. Yılmaz and S. Uludag. Timely detection and mitigation of iot-based cyberattacks in the smart grid. *Journal of the Franklin Institute*, 2019.

[94] D. York. Chapter 4 - control channel attacks: Fuzzing, dos, spit, and toll fraud. In D. York, editor, *Seven Deadliest Unified Communications Attacks*, pages 71 – 92. Syngress, Boston, 2010.

[95] S. Yu, C. Wang, K. Ren, and W. Lou. Achieving secure, scalable, and fine-grained data access control in cloud computing. In *2010 Proceedings IEEE INFOCOM*, pages 1–9. Ieee, 2010.

[96] T. Yu, V. Sekar, S. Seshan, Y. Agarwal, and C. Xu. Handling a trillion (unfixable) flaws on a billion devices: Rethinking network security for the internet-of-things. In *Proceedings of the 14th ACM Workshop on Hot Topics in Networks*, page 5. ACM, 2015.

[97] J. Yuan and K. Mills. Monitoring the macroscopic effect of ddos flooding attacks. *IEEE Transactions on Dependable and secure computing*, 2(4):324–335, 2005.

[98] X. Zhang and S. Wang. Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security. *Pattern Recognition Letters*, 25(3):331–339, 2004.