I. J. Computer Network and Information Security, **2022, 1, 25-39**
Published Online February 2022 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2022.01.03

A Bayesian Attack-Network Modeling Approach to Mitigating Malware-Based Banking Cyberattacks

Aaron Zimba
Department of Computer Science and Information Technology, Mulungushi University
E-mail: gvsfif@gmail.com
ORCID: http://orcid.org/0000-0002-2587-106X

Received: 06 April 2021; Revised: 13 June 2021; Accepted: 13 August 2021; Published: 08 February 2022

Abstract: According to Cybersecurity Ventures, the damage related to cybercrime is projected to reach $6 trillion annually by 2021. The majority of the cyberattacks are directed at financial institutions as this reduces the number of intermediaries that the attacker needs to attack to reach the target - monetary proceeds. Research has shown that malware is the preferred attack vector in cybercrimes targeted at banks and other financial institutions. In light of the above, this paper presents a Bayesian Attack Network modeling technique of cyberattacks in the financial sector that are perpetuated by crimeware. We use the GameOver Zeus malware for our use cases as it's the most common type of malware in this domain. The primary targets of this malware are any users of financial services. Today, financial services are accessed using personal laptops, institutional computers, mobile phones and tablets, etc. All these are potential victims that can be enlisted to the malware's botnet. In our approach, phishing emails as well as Common Vulnerabilities and Exposures (CVEs) which are exhibited in various systems are employed to derive conditional probabilities that serve as inputs to the modeling technique. Compared to the state-of-the-art approaches, our method generates probability density curves of various attack structures whose semantics are applied in the mitigation process. This is based on the level exploitability that is deduced from the vertex degrees of the compromised nodes that characterizes the probability density curves.

Index Terms: Cyberattack, Crimeware, Banking malware, Bayesian network, GameOver Zeus.

1. Introduction

Internet usage has today touched almost every area of our daily lives including the way we handle finances [1, 2]. Traditional ways of trading and marketing, both at personal and corporate levels, have been replaced by innovative Internet applications and online systems [3]. Banks in several countries have jumped onto the bandwagon to provide access to financial services through the Internet to customer accounts. The evident advantage of such online services is the convenience and elimination of expensive retail offices and bureaucratic paper transactions. More recently, mobile banking has emerged as a channel to provide various platforms for online banking. This in part is due to the increase in the number of websites from just one, the first ever-website in 1991 [4], to about 1.75 billion as of January 2020 [5]. In the same way, the number of Internet users has grown to about 4.4 billion [6]. This enormous number of Internet users has attracted cybercriminals who have evolved in their tactics. Just like street crime, which historically grew in relation to population growth, a similar phenomenon i.e., the evolution of cybercrime with increased Internet users and digital targets, is being witnessed today. In the same vein, cyberattacks have evolved from hobbies and self-gratification attacks [7] to financial-based crimes which pose a serious threat to today's networks [8, 9].

Cybersecurity Ventures predicts that by next year (2021), cybercrime will cost the world about $6 trillion annually in damages [10] from half the value of $3 trillion in 2015. As such, the unprecedented damage caused by cybercrime to both private and public enterprises is driving up huge spending on Information Technology security. Cumulatively, global spending on cybersecurity products and services is predicted to exceed $1 trillion from 2017 to 2021 [11]. However, cybercrime is a culmination of different types of criminal activities that either target or use a computer as an instrument to further illegal ends. There are different types of cybercrimes each targeting a specific industry. The graph in Figure 1 shows the average annual cost of cybercrime by industry [12].

As can be seen from Figure 1, cybercrime in the banking industry represents the largest segment of cyber-attacks in the technology-connected world today. Unlike cyber-attacks in other industries, cyber-attacks in the banking industry are lucrative to cybercriminals in that the attacker is nearest to the monetary proceeds in the attack chain [13]. But even in financial cybercrime, attackers employ a variety of attack vectors to actualize attacks in the banking industry. Such vectors include email attachments, weak and stolen credentials, social engineering, and malware [14]. The graph in Figure 2 shows the average annual cost of cybercrime by type of attack.

As can be seen in Figure 2, malware plays a pivotal role in cyber-attacks common in the banking industry. As such, malware is one of the main threats to the Internet security of banking systems.

Malware has today evolved with high levels of sophistication such as the use of encrypted payloads and obfuscation techniques which makes the detection thereof a challenging task. About 54% of the monetary loss caused by cyber-crimes has been attributed to the use of malware [8] and this has put more pressure on Anti-Virus companies.

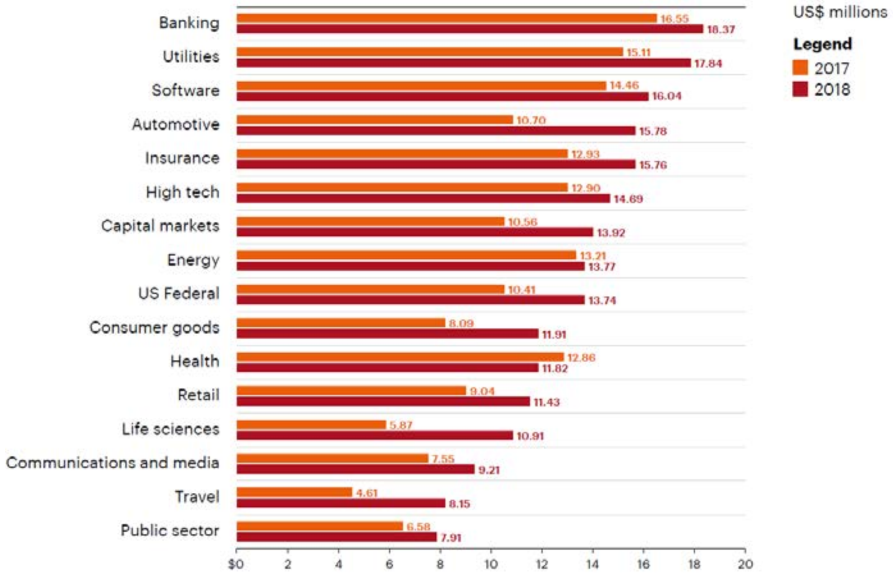

Fig.1. The 2019 average annual cost of cybercrime by industry [12]

It is not surprising that now Crimeware-as-a-service (CaaS), a new commodity based on the cloud computing paradigm, has emerged in the underground market [15]. This has in turn called for cybersecurity research in the banking industry and the associated crimeware [16, 17, 18].

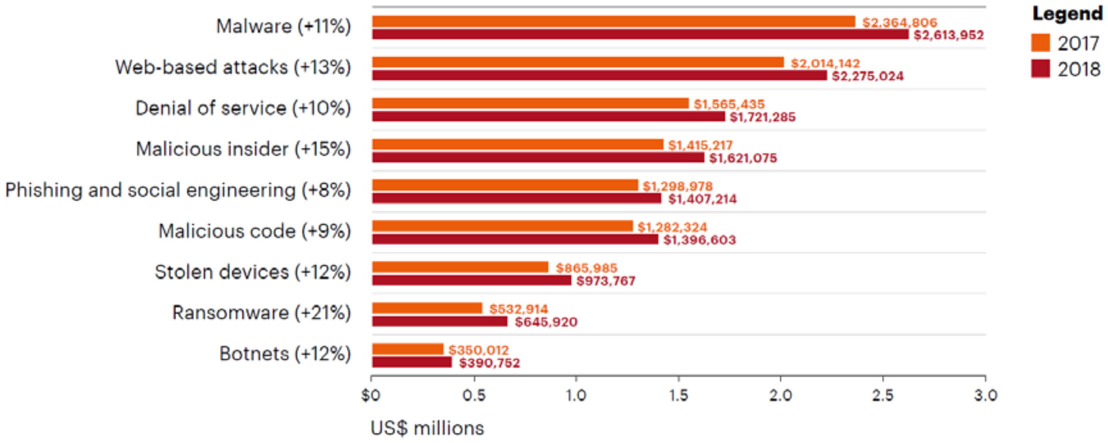

Fig.2. The average annual cost of cybercrime by type of attack [12].

It is from this perspective that this paper presents Bayesian attack-network modeling techniques on banking crimeware considering that the banking industry is the most affected by cyberattacks in monetary value and attackers employ malware as the main attack vector [12]. We evaluate our modeling approach using the GameOver Zeus (GoZ) APT botnet which is capable of exploiting cloud computing environments [19] in the presence of specific vulnerabilities. We choose the Zeus crimeware because it is the most widespread and pervasive type of financial malware [20] in the wild. The prevalence of the Zeus banking trojan is echoed by the fact that it accounts for most of the cybercrime targeting banks and small businesses [21]. As such, the major objective is to model these malware-based cyberattacks in the financial sector with the purpose of developing a methodology to be used in the mitigation process.

The rest of the paper is organized as follows; Section 2 presents related works while Section 3 is dedicated to the Zeus P2P cybercrime network. The methodology and framework are presented in Section 4 whereas the illustrative results are brought forth in Section 5. The conclusion is drawn in Section 6.

2. Related Works

There is vast literature on malware and its usage in cyber-attacks. Inasmuch as there are several reports on financial cyberattacks [22], there's little academic research on the crimeware that is used in these cyberattack campaigns. Research on banking crimeware is usually specific to a certain type of malware, trojans in most instances [23]. Authors in [24] present an overview of techniques, issues, and cite examples in the malware detection landscape. They consider a wide spectrum of banking malware and the avenues in which they pose security threats to end-users. They further detail how the malware operates and propagates, and how such threats may be addressed using the latest malware detection techniques. They argue that there is no single technique best suited to detect all types of malware and state the deficiencies of signature-based detection in that such techniques cannot detect novel malware. They conclude by arguing that an understanding of the sophistication and adaptability of banking malware and the associated infection mechanisms can contain such malware threats if such is coupled with circumspect behavior on the part of the end-user.

Authors in [20] explore the incentives and strategies employed by attackers by inspecting the instructions sent from the Command & Control (C2) to the machines infected with the Zeus malware. Based on the more than 10,000 configuration files and over a million URLs, they develop an interesting metric that ranks the relative attractiveness of domains as a target. They, however, deduce that the target size does not intimate the susceptibility to the attack, neither the attack intensity thereof but rather a threshold for getting attacked. Further analysis of the inject codes from the URLs using the Cosine similarity revealed that the vast majority of the injected codes are reused and shared. This phenomenon of code reuse, sharing, selling, or stealing among attackers suggests low entry barriers and low development costs. The overall conclusion is that the underground market of crimeware does not determine the attack volume nor the selection of attack targets.

In [3], authors present static malware analysis (reverse engineering) of the Zeus banking crimeware toolkit which emerged as a powerful cyberattack tool for controlling compromised botnets. The approach of this digital autopsy sought to uncover the various obfuscation levels and shed more light on the associated malware code. Encryption keys were extracted from reverse engineering and subsequently used to decrypt the corresponding captured network traffic and extract vital attack essentials. The authors argued that the extraction of such detailed information accorded the opportunity to deliberately hijack and insert falsified information into the botnet for the purposes of countering the malware attack.

Authors in [21], presented preliminary results on the classification of the Zeus banking malware family using different machine learning algorithms. The authors identified 65 features from the malware that are unique and robust enough to identify the different malware families associated with Zeus. They used artifacts like the file system, registry, and network features to identify distinct malware families. They report a high classification accuracy of 95%. The authors sought to address the classification problem because, since the leaking of the malware source code in 2011, various versions of the malware had emerged. Furthermore, the newer versions had new functionalities, e.g., the original Zeus malware would copy itself to the APPDATA directory and store various important information after encryption whereas the newer version stored the encrypted configuration file in the registry under a random key name.

Authors in [25] present the techniques on how Zeus malware has penetrated mobile phone devices. The authors bring to light the non-conventional botnets orchestrated by Zeus, mobile botnets. Dubbed as Zitmo (Zeus-in-the-mobile), the primary targets are Android mobile phone users and they are used as part of the mobile botnet and C2 infrastructure. The malware leverages social engineering as the attack vector and steals mobile transaction authorization numbers (mTAN) in SMS messages between banks and customers. They report that Zitmo supports multiple platforms and the attacks thereof have been witnessed in several European countries [26]. Furthermore, the authors argue that even though the GameOver Zeus takedown operation succeeded in cutting communication within the Zeus botnet infrastructure, it did not entirely bring down Zeus because GameOver Zeus switched to DGAs (Domain Generation Algorithms) from the P2P infrastructure which equally makes it difficult to mitigate.

3. The Zeus P2P Network

Zeus is a banking trojan that first appeared in 2007 [25]. As a banking trojan, Zeus primarily extracts banking credentials from infected systems even though it can be used to carry out other malicious and cybercriminal tasks. The initial infection vectors were phishing and spam emails [27] though the malware exploits vulnerabilities discovered on exposed hosts [28]. The malware would infect potential victims and enlist them to a botnet. The attacker, behind a C2, would then use these infected machines in the botnet to carry out various attack commands which include stealing banking credentials and spreading other malware such as ransomware. Disrupting communication between the infected hosts and the C2 is one common approach used in mitigating such attacks [29].

Cybercriminals introduced GameOver Zeus whose attack paradigm was different from the traditional Zeus. GoZ uses a DGA technique and generates a large number of domains that are essentially used as "rendezvous points" between the C2 and the botnet. Using this technique, the malware can generate as many as 10,000 domains in a day [30].

GameOver Zeus is a crimeware primarily used to harvest banking credentials. However, it's also known to deliver ransomware payloads such as CryptoLocker. On December 9th, 2009 [31], the Zeus botnet was found on Amazon's EC2 IaaS cloud computing offering where it primarily used vulnerable VMs as command and control infrastructure [32, 33, 34]. Although initially used as a platform for attacking financial institutions, it has also been used as a framework for APTs [35, 36, 37, 38] and also directly in espionage activities targeting specific countries [39]. We use the GameOver Zeus botnet, which is an advanced DGA-based P2P extension of the initial Zeus family [40], as a use case for our model because it is a botnet deployable on the prevailing computing paradigm (cloud computing) utilized by various APT campaigns [41] and cyberattacks targeting individuals. We choose to use GoZ because it is still an active threat that continues to grow with the infection of new hosts, hence the dynamic network. As of 2017, it accounted for 28% of all banking related malware [42].

The diagram in Figure 3 depicts the structure of GameOver Zeus based on the FoxIT report [43]. The infrastructure is made up of three layers; the C2 layer, the Proxy layer, and the P2P layer. The C2 layer houses the command-and-control center of the attack campaign. This is where malware agent updates, configurations, directives, and updates are located. This could be another compromised network under the control of the threat actors which is usually accessed via anonymity networks such as Tor. The proxy layer is made up of compromised cloud components, as was the case of Amazon EC2 [31].

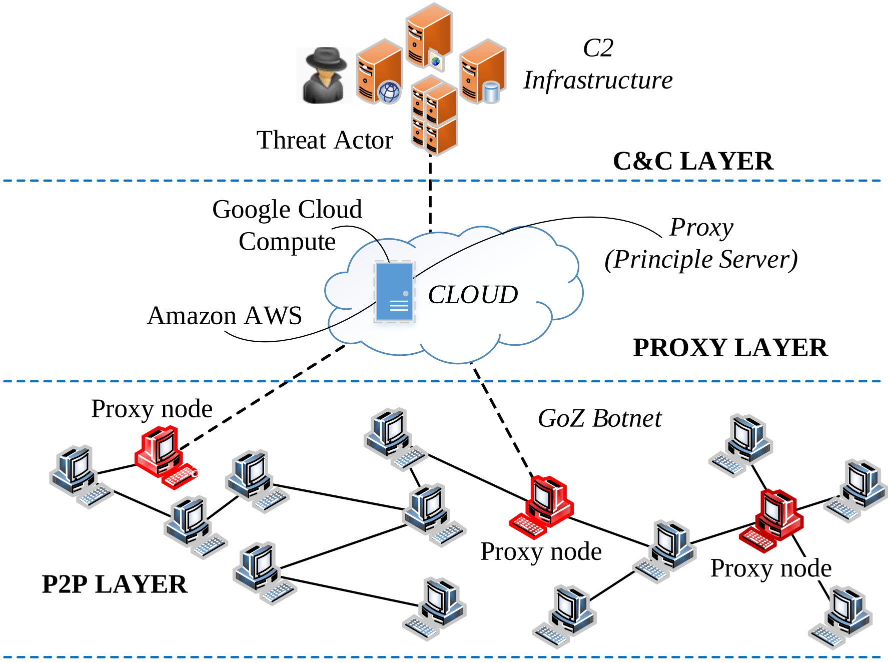

Fig.3. The infrastructure of GameOver Zeus botnet.

The compromised hosts could be located anywhere and they receive resources from the C2 layer. Such a layered approach ensured that even when the infected hosts were detected and subsequently removed, the attacker could look for other vulnerable hosts and push the malware resources from the C2 layer. As such, the key nodes at this layer are the compromised hosts known as principal proxy servers. The P2P layer is made up of two types of hosts compromised via different infection vectors: a group of internet-facing hosts also known as proxy nodes (denoted in red) and other infected nodes not directly connected to the Internet. The proxy nodes are the key nodes at this layer because they communicate directly to P2P infected hosts and the principal proxy servers.

It is visible from the network structure in Figure 3 that the propagation of the attack largely depends on the reachability from one infected node to the other. As such, the infected hosts should either share a trust relationship or exhibit vulnerabilities whose exploitation will facility the traversability of an attack from one through to the destination host. Our modeling approach in this paper explores both possibilities.

4. Methodology and Framework

In order to achieve the research objectives, we adopt a modelling methodology based on Bayesian attack networks and devise a framework of attack graphs and Common Vulnerability Scoring System (CVSS) base scores [44] where we employ CVSS-based conditional probabilities to deduce attack paths. These serve as inputs when generating probability curves which are ultimately used in the mitigation process. This technique generates probability density curves that entail the attack complexity and time expense. Bayesian networks models are adopted in our methodology because they enable us to chain the vulnerabilities exploited in targeted hosts as the attack ensues to the final host.

The Bayesian Attack Network

A Bayesian Attack Network (BAN) is a 3-tuple directed acyclic graph such that: $BAN \ = \ (\alpha, \beta, \gamma)$ where;

- $\alpha = G(V, E)$ is a graph of discrete random variables where V denotes a set of nodes V = $\{n_1, n_2, n_3, \dots n_i\}$. and E denotes a set of edges E = $\{e_1, e_2, e_3, \dots e_i\}$.
- β is a set of quantitative network parameters
- γ is a set of attack steps entailing the attack traversability from an exploited node to another via an attack edge.

We model the vulnerability exhibited by a host in the GoZ botnet as a node while the probability of exploiting such a node is modeled as an edge. As such, we come up with a weighted directed attack graph whose nodes represent vulnerabilities while the edge weights represent the attack probability. To find what vulnerabilities are exhibited by a given node, we use CVEs [45] while to find attack probability, we use the CVEs to convert the CVSS base scores to the corresponding probability.

CVSS Base Scores and Attack Probability

CVE is a database that classifies different vulnerabilities [46] while CVSS is an open set of standards that evaluates the threat level of a vulnerability by assigning a severity along a scale of {0 - 10} [47]. On the other hand, the CVSS Base Score quantifies the intrinsic properties of a vulnerability immune to perturbation over time [48]. Expert knowledge in this domain is used when assigning base scores. Considering the fact that our modeling approach utilizes Bayesian probabilities in the BAN, we convert the base score into the intrinsic probability of a vulnerable node in the following manner:

$$Pr_{intr}(n_i) = P(n_i \mid \forall\ c \in R_i) = \frac{Base\ Score_i}{10\ (CVSS\ range)} \tag{1}$$

where c and R_i denote the conditions that must exist for the vulnerability to be exploited and the relational dependencies respectively. The probability $Pr_{intr}(n_i)$ is the probability of attacking the vulnerable node n_i, hence attack probability. Since some nodes might exhibit more than one vulnerability, it is worth noting that the resultant graph in such a case contains multiple edges (representative of attack probability) to the node indicating the multiple vulnerabilities.

Bayesian and Conditional Probability Tables (CPTs)

Multiple attack paths exist in a botnet since the members of the botnet have vulnerabilities that can be exploited from different sources. If the attacker is to reach the target from a given source, she must exploit the vulnerable nodes in the botnet as she traverses the network to the destination. As such, each vertex node in the botnet carries has a conditional probability distribution dependent on the node exploited previously. The full joint probability distribution is thus expressed as:

$$Pr(n_1, \dots, n_i) = \prod_{i=1}^{n} Pr\left(n_i \mid parents(n_i)\right) \tag{2}$$

Following from Equation (2), the conditional probability of exploiting a node n_i with two incoming edges from n_{i-1} and n_{i-2} is:

$$\Pr\ (n_i) = \Pr\ (n_i | n_{i-1}, n_{i-2}) \tag{3}$$

If both nodes n_{i-1} and n_{i-2} must be exploited before reaching the node n_i, then the attack scenario is a conjunction events and the resultant conditional probability is calculated as:

$$\Pr\ (n_i) = \Pr(n_{i-1} \cap n_{i-2}) = \Pr(n_{i-1}) \cdot \Pr(n_{i-2}) \tag{4}$$

If either node n_{i-1} or n_{i-2} should be exploited before reaching the node n_i, then the attack scenario is a disjunction events and the resultant conditional probability is calculated as:

$$\Pr\ (n_i) = \Pr(n_{i-1} \cup n_{i-2}) = \Pr(n_{i-1}) + \Pr(n_{i-2}) - \Pr(n_{i-1} \cap n_{i-2}) \tag{5}$$

By leveraging Equations (1) - (5), we compute the Conditional Probability Tables (CPTs) that represent the different conditions under which a vulnerable node in the botnet can be exploited. Table 1 shows the generic CPT semantics of a node whose exploitation is dependent on other nodes both for the disjunction and conjunction attack scenarios. We henceforth base all our CPTs computations on this table.

Table 1. CPT for conjunctive and disjunctive attack events

$Parents$		$Disjunctive\ Attack\ \ Events$		$Conjunctive\ Attack\ Events$	
n_i	n_j	$Pr(c = 0 \mid F)$	$Pr(c = 1 \mid T)$	$Pr(c = 0 \mid F)$	$Pr(c = 1 \mid T)$
0	0	1	0	1	0
1	0	$1 - Pr(n_i)$	$Pr(n_i)$	1	0
0	1	$1 - Pr(n_j)$	$Pr(n_j)$	1	0
1	1	$1 - Pr(n_i \cup n_j)$	$Pr(n_i \cup n_j)$	$1 - Pr(n_i \cap n_j)$	$Pr(n_i \cap n_j)$

One thing worth mentioning is that the hosts/nodes that are exploited in the GoZ attacks do not necessarily domicile in the same network but will be part of the GoZ botnet structure residing on the C2 layer, Proxy layer, or P2P layer as depicted in Figure 3. Additionally, this situation is very strong in financial systems due to the interconnectedness of financial systems infrastructure and other ICT technologies. A practical case in point is the network of the financial system [49] depicted in Figure 4 which illustrates the interoperability of financial systems and other information systems.

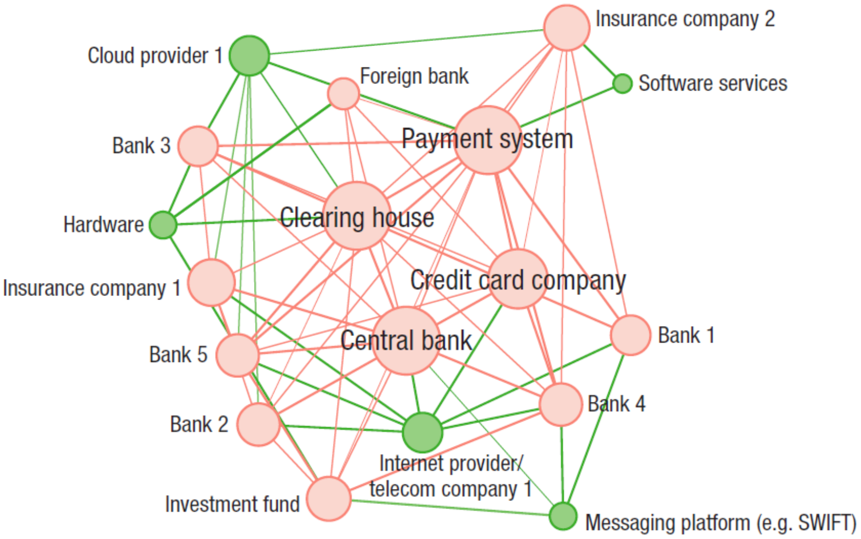

Fig.4. Financial systems and ICT networks [49]

A new version of GoZ is reported to target such systems and has so far affected over 150 different banks and 20 payment systems spanning 15 countries [50].

The implication of such a network structure as in Figure 4 is that the hosts that can exhibit vulnerabilities that can be exploited can either be servers or user workstations. This increases the infiltration points in the attack surface. To this effect, we identify the following as possible attack infiltration sources; public cloud services users, general Internet users, and trusted third-parties.

Modeling Framework

We develop a framework that integrates the tools presented in the previous section. Inasmuch as phishing and spam emails are usually utilized as attack vectors, we rather choose vulnerability exploitation as the attack vector. This is so because CVEs can be directly utilized to generate base scores (and compute the corresponding probability) unlike in spam emails or phishing where we wouldn’t be certain which vulnerability was exploited through the malicious link. Our framework is shown in Figure 5.

Our framework consists of five steps. In the first step, an attack vector is identified from the whole attack surface. In our case, the attack vector is vulnerability exploitation. In step 2, the CVE that identifies with the vulnerability from step one is extracted and the corresponding base score is deduced. Still, in step 2, the base score is used to compute the attack probability which is intrinsic to the exploited node. Step 3 uses the computed probability and combines them to evaluate the different conditional probabilities after which conditional probabilities are further computed. Step 4 generates the attack paths associated with different conditional probabilities by mapping them to the corresponding edges in the attack graph. Finally, the properties of these attack paths are used in step 5 to generate probability density curves from which we infer the attack properties of different attack scenarios. Such inference is useful when devising defensive and mitigative strategies.

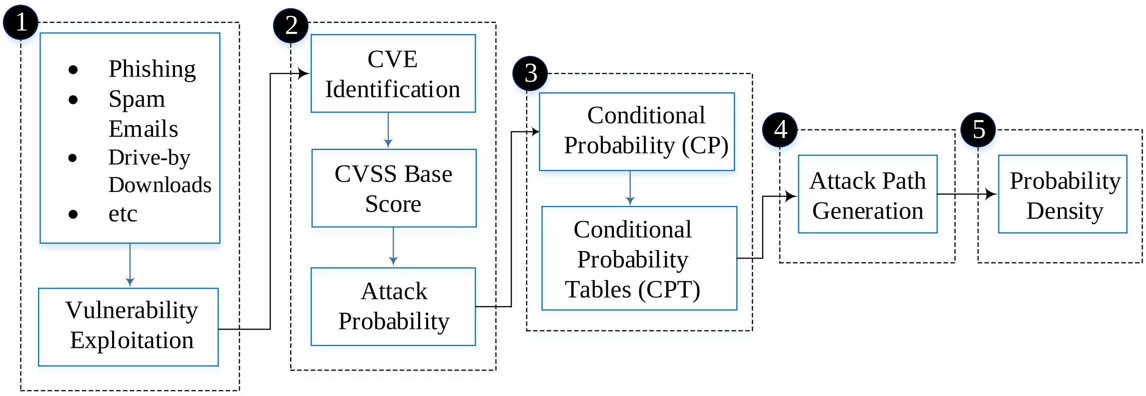

Fig.5. The Modelling Framework

5. Illustrative Results and Analysis

Based on the GoZ infrastructure in Figure 3 and the semantics of the network infrastructure in Figure 4, we deduce the corresponding attack graph as illustrated in Figure 6. We identify three attack sources denoted as n_{0i} where $i = 1,2,3$. These correspond to the earlier discussed infiltration sources respectively denoted as: n_{01} (CSP Public) – a group/domain of users with legitimate access to the cloud who would otherwise need privilege escalation to access other cloud components; n_{02} (General Internet Users) – attack sources in this domain denote entities without tenant accounts with the cloud, they denote the general user from the Internet; n_{03} (Trusted Third Party) – a network sharing trust relationship with the CSP like software companies, messaging platforms such as SWIFT, federated cloud, or cloud brokers.

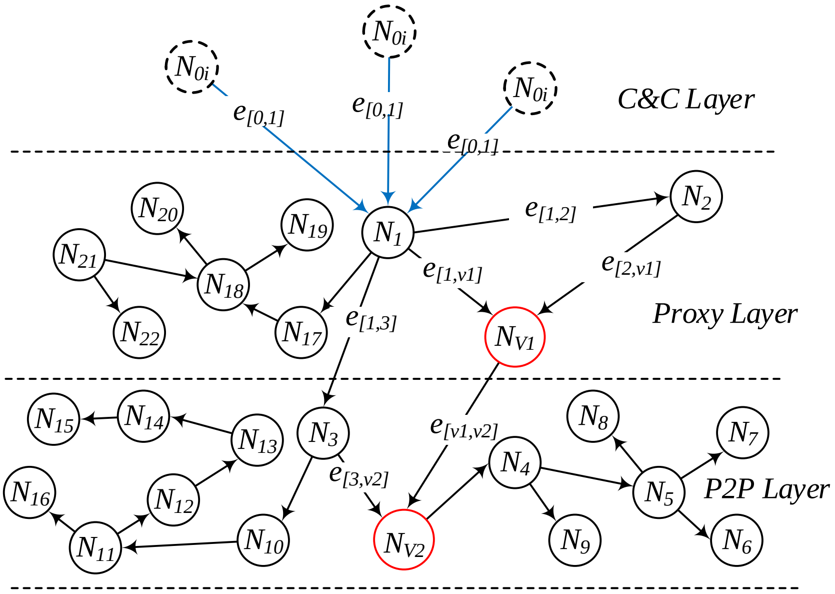

Fig.6. Attack graph based on GoZ botnet.

As such, we have three attack instances each starting at n_{01}, n_{02} and n_{03} with a different initial probability of infection $Pr(n_{0i})$. The nodes n_{v1} and n_{v2} denote the target hosts from where data (banking credentials) is to be exfiltrated. n_{v1} is a target node within the cloud network or hosts on the Proxy Layer of the GoZ botnet whilst n_{v2} is a target host residing outside the cloud network which could any host in the P2P layer. For the purposes of our modeling, if n_1 exhibits a vulnerability CVE-2010-0188 exploitable by GoZ [51] which enables the distribution of the Zeus Bot Client (Zbot) [52] to target hosts, it can be used as a pivot node to reached other nodes. In our exposition, n_{v1} exhibits two vulnerabilities, one exploitable via n_1 and the other via n_2. In the former, n_{v1} is the target where the data resides while in the latter it is used as a stepping stone (pivot) to reach node n_{v2}. Data on node n_{v2} can also be accessed by exploiting node n_3 which is only exploitable via n_1. Table 2 shows the various CVSS vulnerabilities exhibited by the nodes of the attack graph in Figure 6. Our use case contains four attack scenarios from the three infiltration sources. These attack scenarios correspond to the attack paths;

$$P_1 : e_{[0,1]} \rightarrow e_{[1,3]} \rightarrow e_{[3,v2]},$$

$$P_2 : e_{[0,1]} \rightarrow e_{[1,v1]},$$

$$P_3 : e_{[0,1]} \rightarrow e_{[1,2]} \rightarrow e_{[2,v1]}$$

$$P_4 : e_{[0,1]} \rightarrow e_{[1,2]} \rightarrow e_{[2,v1]} \rightarrow e_{[v1,v2]} \ .$$

The two attack paths P_2 and P_3 denote scenarios where the target data resides in the Proxy Layer whereas P_1 and P_4 denote attack scenarios where the target data resides in the remote GoZ P2P network. In the first attack path, the attacker attains access to the Proxy Layer via a legitimate user account from n_{01}. This could be a compromised user account or a bogus user account created by the attacker [53], hence we assume a high probability of infiltrating the principal server $Pr(n_{01}) = 1$ on the proxy layer from the C2 layer. By exploiting CVE-2016-0036 on the node n_{v2}, the attacker reaches a compromised node (proxy node) in the P2P network which is an Internet-facing device.

Table 2. Intrinsic Node Probability Assignment

Associated Attack	Node (n_i)	CVSS CVE ID	$Pr(n_i)$
CSP P Infiltration	n_{01}	Subjective	-
Zbot RCE	n_1	CVE-2010-0188	0.92
SSH Roaming	n_2	CVE-2007-0777	0.42
RDP Access	n_3	CVE-2016-0036	0.88
VM Escape	n_{v1}	CVE-2017-0109	0.74
Hyperjacking		CVE-2017-3623	0.99
RCE Attack	n_{v2}	CVE-2017-0075	0.74
RCE Backdoor		CVE-2016-0036	0.88

As such, the attacker establishes local network residency with the target node which is a condition required to exploit the target node n_{v2}. One of the methods used to establish this residency is to ascertain the subnet to which the attack belongs. If the attacker resides in a different subnet, she should endeavor to use a new VM that would fall in the same subnet. This is easily achieved relaunching the VM. This condition is stipulated in the Access Vector parameter of the base score of the vulnerability CVE-2017-0075 exhibited by node n_{v1}. If the stipulation in the Access Vector is for network access is "global" or "Internet", there would be no need for the attack to establish local residency. This would entail that the node is more vulnerable and easily exploitable. Since the target in the second attack scenario resides on the Proxy Layer, the attacker directly exploits this node without an intermediary after installation of the proxy C2 on n_1. In the third attack scenario, the attacker has to go through a pivot node n_2 to get to the target n_{v1}. In the fourth attack scenario, the target node is n_{v2} and since the pivot node n_{v1} in the Proxy Layer is outside the local network, the attacker exploits a vulnerability CVE-2016-0036 which does not require the establishment of local network residency. We deduce the connectivity matrix **(CM)**, which essentially is the adjacency matrix, to show the linkability amongst the nodes. This enables us to view the traversability of the Bayesian attack network. Without this traversability, the attacker cannot laterally move from one to the other to reach the target. In such a scenario, if there are no banking credentials and the attacker is unable to traverse to another node, the attack instance is a dead end

$$CM = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 \end{bmatrix}$$

The zeros in the major diagonal of the matrix entail that there are no loops or cycles in the attack graph. An entry of 1 in the matrix imply the exploitability of the row and column nodes represented by the element. A zero implies that the node represented by the row and column are not exploitable one from the other. Such a case, the attacker will have to find an alternative attack path. To find the extent of exploitability amongst these nodes, we map the conditional probabilities derived from CVSS base score to the connectivity matrix. The result is a weighted matrix (W) that shows more details as to the exploitability of the nodes in the attack graph.

Since the exploitability of a given node is dependent on the likelihood of exploiting its parent, we use the conditional probability to find these probabilities which we use as parameters in the edge mapping function.

The conditional probability computations of node n_1, n_2, n_3, n_{v1} and n_{v2} are shown in Table 3, Table 4, Table 5, Table 6, and Table 7 respectively.

Table 3. CPT assignments for node n_1

n_{01}	n_{02}	n_{03}	$Pr_{n_1}(c = 0 \mid F)$	$Pr_{n_1}(c = 1 \mid T)$
T	F	F	0.08	0.92
F	T	F	0.08	0.92
F	F	T	0.08	0.92

$Pr(n_1|n_{01}) = 0.92$, $Pr(n_1|n_{02}) = 0.83$,
$Pr(n_1|n_{03}) = 0.60$

Table 4. CPT assignments for node n_2

n_1	$Pr_{n_2}(c = 0 \mid F)$	$Pr_{n_2}(c = 1 \mid T)$
T	0.58	0.42
F	1	0

$Pr(n_2|n_1) = 0.39$

Table 5. CPT assignments for node n_3

n_1	$Pr_{n_3}(c = 0 \mid F)$	$Pr_{n_3}(c = 1 \mid T)$
T	0.12	0.88
F	1	0

$Pr(n_3|n_1) = 0.81$

Table 6. CPT assignments for node n_{v1}

n_1	n_2	$Pr_{n_{v1}}(c = 0 \mid F)$	$Pr_{nv_1}(c = 1 \mid T)$
T	F	0.26	0.74
F	T	0.01	0.99

$Pr(n_{v1}|n_1) = 0.68$, $Pr(n_{v1}|n_2) = 0.42$

Table 7. CPT assignments for node n_{v2}

n_3	n_{v1}	$Pr_{n_{v2}}(c = 0 \mid F)$	$Pr_{n_{v2}}(c = 1 \mid T)$
T	F	0.26	0.74
F	T	0.12	0.88

$Pr(n_{v2}|n_3) = 0.65$, $Pr(n_{v2}|n_{v1}) = 0.87$

Using the above computed conditional probabilities, we apply a direct edge-weight mapping to populate the weight matrix of the GoZ attack graph. We define an edge-weight mapping operation Ω by mapping the above conditional probabilities of a node that is dependent on its parents to an edge using the formula:

$$\Omega: E[G_i(V,E)] \rightarrow \mathbb{R}^+ \cup \{0\} \leq 1, where\ G_i(V,E) \subset \alpha.$$

Therefore, the edge weight between two nodes n_i and n_{i-1} is expressed as:

$$\Omega(e_{[i,i-1]}) = \Pr(n_i) \cdot \Pr(n_{i-1}) \tag{6}$$

To calculate the cumulative edge weight that makes up an attack path linking multiple nodes, we use the product of the probabilities (edges) since this represents a conjunction of attack events:

$$\Omega(P) = \prod_{e \subset P} \Omega(e) \tag{7}$$

We define another parameter γ of the Bayesian attack network which is the edge cardinality in an attack path. This is essentially the number of edges, hence the attack steps.

$$|\gamma(\Omega)| = \sum_{i=0}^{i-1} \Omega(e_{[i,i-1]}) \tag{8}$$

For compactness purposes, instead of computing four different weight matrices each with a distinct attack source, we aggregate the conditional probabilities of the earlier discussed attack sources. The resultant weight matrix is:

$$W = \begin{bmatrix} 0 & 0.78 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.39 & 0.81 & 0.68 & 0 \\ 0 & 0 & 0 & 0 & 0.42 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.65 \\ 0 & 0 & 0 & 0 & 0 & 0.87 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Since we use CVSS base scores for probability computation, which does not change over time, the attack activity can be modeled as a Poisson function obeying the Erlang distribution [54]:

$$f(t{:}\,\lambda, k) = \frac{(\lambda)^k}{(k-1)!} \cdot t^{k-1} \cdot e^{-\lambda t} \;\; for\, t, \lambda \geq 0 \tag{9}$$

where λ is the success rate of the attack event and k is the attack complexity dependent on the number of attack steps. The more the number of attack steps, the more complex the attack scenario.

Using Equation (7) and Equation (8), we compute the overall attack probabilities of each of the attack paths from the matrix and the associated attack steps:

$$\begin{aligned} P_1{:}\,\Omega\left(e_{[i,i-1]}\right) &= 0.411, & |\,\gamma(\Omega)| &= 3 \\ P_2{:}\,\Omega\left(e_{[i,i-1]}\right) &= 0.530, & |\,\gamma(\Omega)| &= 2 \\ P_3{:}\,\Omega\left(e_{[i,i-1]}\right) &= 0.128, & |\,\gamma(\Omega)| &= 3 \\ P_4{:}\,\Omega\left(e_{[i,i-1]}\right) &= 0.111, & |\,\gamma(\Omega)| &= 4 \end{aligned}$$

Using Equation (9), we apply the above parametric values to generate probability density curves. The resultant graphs are shown in Figure 7.

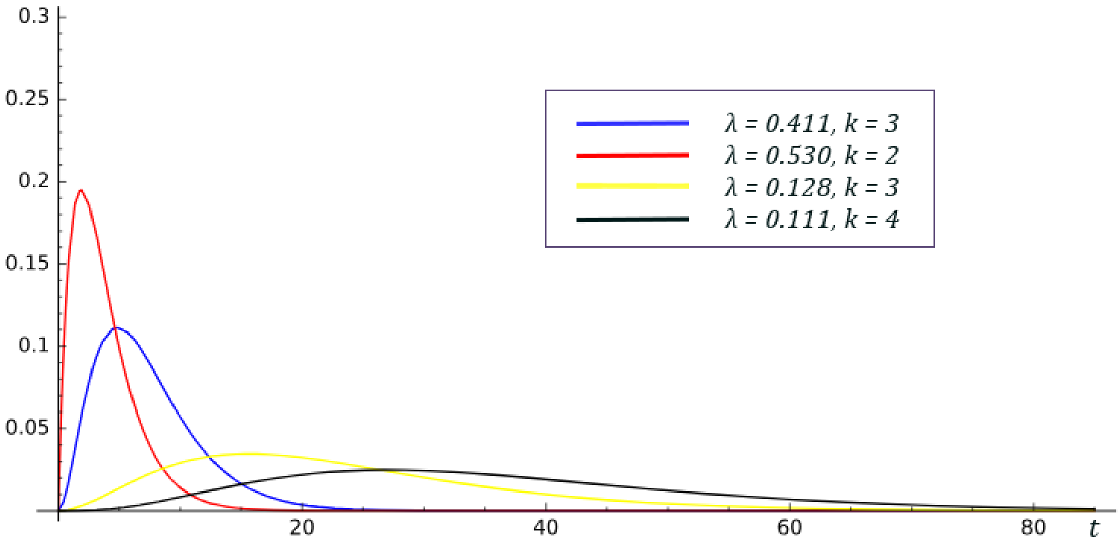

Fig.7. Probability density curves for the four GoZ attack instances.

All the four density curves are positively skewed denoting intensified attack activities in the early stages of the attack for the first two attack paths. The first attack scenario corresponds to the blue curve (1st path) with 3 attack steps where the mean and median of the distribution lie after the mode. This is echoed by the few numbers of attack actions in this path thus depicting its susceptibility. The second attack scenario is comprised of two attack instances generating the red curve (2nd path) which has a higher attack rate and fewer attack actions than the first attack path. Though the graphs of the 1st and 3rd paths have the same k value (implying a tie), it’s visible from the resultant graphs that the 1st attack path has the shortest route considering the position of the mode. As such, we can infer the characteristics of the attack paths from the generated probability density functions and use them for prioritizing which attack paths need attention in the mitigation process. In this case, key nodes and edges in the shortest paths should be given priority when designating failure nodes for mitigation purposes. This is so because they reflect a higher degree of exploitability. Such an approach is effective because in most real-world production networks, system administrators are usually tempted to prioritize security certain types of devices based on the nature of the device and not necessarily the degree of exploitability. This might give a false sense of security in that attackers might still reach the patched nodes since the attack paths through the key nodes and edges are not eliminated.

According to our framework presented in Figure 5, the attack vector used in this modeling technique is vulnerability exploitation. However, we are alive to the fact that GoZ uses phishing and spam emails as another attack vector. In such a case, it is improbable to predict the growth of the GoZ botnet because newly infected nodes will be added to the Proxy and P2P layer randomly depending on when the victim clicks on the clickbait and the availability of antivirus software on the machine. However, the probability of realizing such attacks can be obtained from expert knowledge such as phishing probability engine [55] and phishing mean click-rate [56]. Considering such a scenario, we can spontaneously grow the botnet (the attack network) using Scale-free networks of complex network theory [57]. Using such an approach, we can derive the corresponding large-scale Bayesian network demonstrated in Figure 8 using Pajek software. The nodes are denoted by the blue circles whereas the edges are denoted by the grey links.

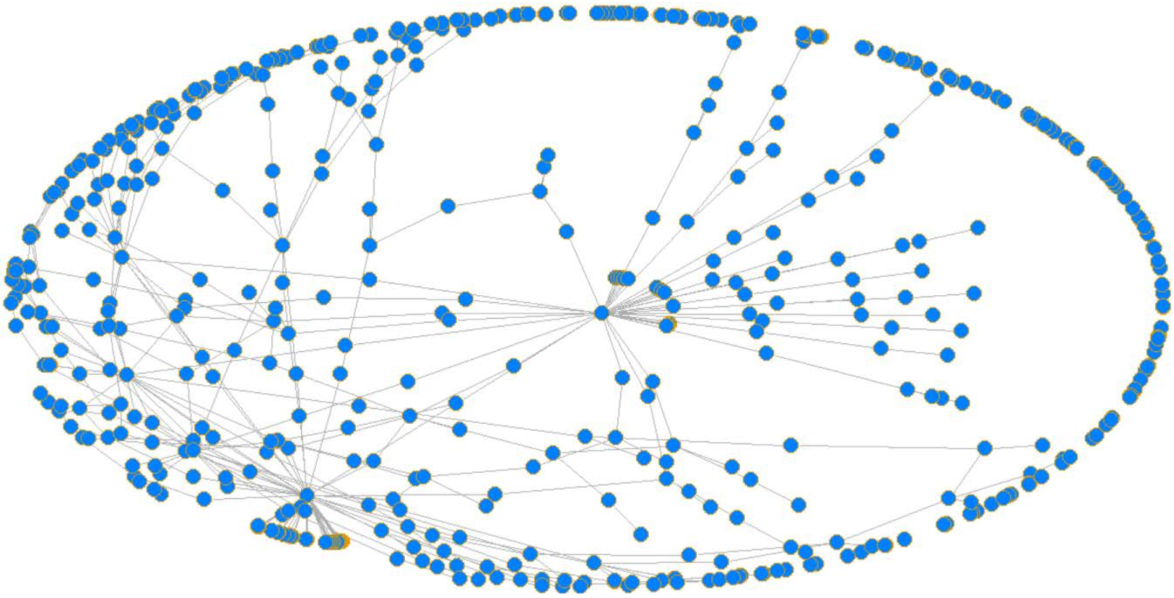

Fig.8. Graph demonstrating a large-scale Bayesian network of the botnet

The nodes of the graph, in this case, represent a host who is targeted by the phishing email [58] whilst the weight of the links represents the probability of effectuating the attack through this attack vector. The isolated nodes of the graph that are not connected represent the hosts that have fallen prey to the phishing email but are not reachable from the C2. The Bayesian network [59] exhibits dynamic characteristics because vulnerable nodes are added to the botnet, hence the GoZ botnet grows with time as a scale-free network [60, 61].

Mitigating such an attack network calls for securing nodes with the highest vertex degree and clustering coefficient. This endeavor will leave for future works.

6. Discussions

The probability density curves reflect the typical behavior of GoZ attacks where the attacker wants to capture and exfiltrate banking credentials and other information in the shortest possible time. This is echoed by the left-skews of the curves. This is in contrast to APTs that seek to establish a long undetected presence before any attack ensues. As such, the approach to countering GoZ attacks should be different from other cyber-attacks; preventative measures should be prioritized since time is a critical factor, especially when zero-day vulnerabilities are being exploited. It is not uncommon for attackers to use zero-day vulnerabilities for such malicious purposes owing to the difficulty of establishing the link between which zero-day vulnerabilities are trending on the darknet and which one is actually being used in an attack instance.

It is visible from the probability density graphs that some attack scenarios have more attack steps than others depending on the vulnerabilities being exploited and the location of the target. Some targets reside on the Proxy Layer whilst others reside on the P2P layer. Though it is logical to assume that attack paths to targets on the Proxy layer will exhibit shorter routes, this is not always the case because the attack paths are determined by the number of available and exploitable vulnerabilities.

In the event where the number of attack events (attack edges) are equal for two given attack paths, the tie is broken by the parameter λ. It is visible from the GoZ network infrastructure that the Proxy layer plays a pivotal layer in the materialization of the attack. As such, mitigation measures should seek to secure vulnerable devices on the Proxy layer. Elimination of the Proxy layer will make it extremely difficult for the infected hosts in the P2P layer to communicate with the C2 infrastructure. The reality however is that some nodes on the proxy layer could reside in a different country under a different jurisdiction. This is where collaboration of law enforcement agencies is very cardinal.

Another thing worth noting is that people access their personal banking services using mobile phones and at times personalized institutional computers such as laptops using the corporate network. Therefore, if such devices are infected with GoZ and enlisted to the botnet, the complexity of the attack graph deepens. This poses a big challenge to CISOs because such attack sources might not be captured in the institution’s security strategy. On the other hand, this calls for a proactive from CISOs to integrate detection and prevention measures that seek to identify not only C2 but other devices in the botnet that might act as pivots. One such approach is identifying the domain from the malware’s DGA or the methodology employed to generate such domains. Consequently, blocking and blacklisting such domains would serve as a temporal measure pending formulation of the overall security strategy.

We argue that one of the major challenges in combating GoZ is identifying to which layer an infected host belongs. This is so because the layers in the GoZ botnet are not owned by the attackers, therefore, it is difficult to pinpoint which infected host is at what layer or predict the layer membership without first seeing how the victim is being used. Since banking malware is known to clone and spoof banking websites, another mitigation strategy is seeking to detect the metadata and hashed values of such sites once rendered to the corporate network as an HTTP/HTTPS response.

7. Conclusion

This paper presented a Bayesian Attack-Network modeling technique of cyberattacks in the financial sector that are perpetuated by crimeware. We used the GameOver Zeus malware for our use cases as it’s the most common type of malware in this domain. We developed a modeling framework that specifies the attack vector adopted through to the generation of the probability density curves. Common Vulnerabilities and Exposures (CVEs) were employed to derive conditional probabilities that served as inputs to the modeling technique and in the generation of attack paths and probability density curves. The vulnerabilities were modeled as nodes of the weighted Bayesian attack graph whereas the weights represent the probability of exploiting the vulnerability. The edges between the nodes represented the traversability of the attacker from one exploited node to the other. The essence of this work lies in breaking this linkability amongst the nodes. Vulnerabilities that could be sequentially exploited were chained together to formulate attack paths which were consequently used to generate probability curves. Each attack path generated its own probability density curve which were large determined by the λ and k parameters, the former being the attack complexity and the latter the atomic attack steps. The skewness of the probability density curves shows intensified attack activities in the early stages of the attack. This is a reflection of the typical behavior of GoZ attacks where the attacker seeks to capture and exfiltrate banking credentials and other information in the shortest possible time.

Our analysis showed that the complex nature of the GameOver Zeus botnet which enlists victims to the C2 layer, Proxy layer, or P2P layer makes it difficult to mitigate such attacks because infected hosts can reside at any layer. However, in real-world networks, victims (members of the botnet) belong to networks of different organizations. As such, we have noted that one of the major challenges in combating GoZ and cyberattacks of its nature is identifying to which layer an infected host belongs. This is so because the layers in the GoZ botnet are not owned by the attackers, therefore, it is difficult to pinpoint which infected host is at what layer or predict the layer membership without first seeing how the victim is being used. As such, the current approach used to counter GoZ attacks where the main aim is to disrupt communication between the infected hosts and the C2 is insufficient as it tends to leave out the Proxy layer which is the pivot between the C2 layer and the P2P layer. We contend that an approach where key nodes and edges are identified is adopted and mitigation priority be given to nodes on the P2P layer. This implies that collaborative efforts amongst law enforcement agencies should not only target the C&C layer but the Proxy and P2P layers as well. This is desirable because even nodes in the other layers can be enlisted to the C&C layer depending on the exploitability. A case on point is where Twitter was used for C&C activities [62].

Additionally, we explored another scenario where the attack vector adopted was phishing emails. The nodes of the graph, in this case, represent a host who is targeted by the phishing email whilst the weight of the links represents the probability of effectuating the attack through this attack vector. It was noted in this case that the resulting Bayesian Attack Network, in this case, exhibits dynamic characteristics representative of scale-free networks owing to the fact that infection through phishing email is dynamic and unpredictable. Mitigating cyberattacks based on this attack paradigm calls for securing nodes with the highest vertex degree and clustering coefficient. This endeavor has been left for future works. As such, our models works given that the vulnerabilities are present and the exploitability and traversability can be ascertained using the adjacency and weight matrix. The limitation however is that the Access Vector parameter in the CVSS base score will determine whether the traversability can be directly achieved or not. In the event that the attacker might need more traversing owing to the nature of the Access Vector, we note that the end result could be undetermined as that would depend a number of characteristics of the attacker such as tenacity and technical skill set.

References

[1] V. Khattri and D. K. Singh, "Implementation of an Additional Factor for Secure Authentication in Online Transactions," *J. Organ. Comput. Electron. Commer.*, 2019.

[2] M. Tripathi and A. Mukhopadhyay, "Financial Loss due to a Data Privacy Breach: An Empirical Analysis," *J. Organ. Comput. Electron. Commer.*, 2020.

[3] H. Binsalleeh *et al.*, "On the analysis of the Zeus botnet crimeware toolkit," in *PST 2010: 2010 8th International Conference on Privacy, Security and Trust*, 2010.

[4] H. Kennedy, "A Brief History of Web Design," in *Net Work*, London: Palgrave Macmillan, 2014.

[5] "Internet Stats & Facts (2020)," *Web Hosting Facts*, 2020. [Online]. Available: https://hostingfacts.com/internet-facts-stats/. [Accessed: 10-Apr-2020].

[6] L. C. Vitorino, A. Lisboa, and R. J. Antunes, "Digital Era: How Marketing Communication Develops Business Innovation–Case Studies," in *Journal of Business Ethics*, 2020.

[7] C. R. Srinivasan, "Hobby hackers to billion-dollar industry: the evolution of ransomware," *Comput. Fraud Secur.*, 2017.

[8] M. Riccardi, R. Di Pietro, M. Palanques, and J. A. Vila, "Titans' revenge: Detecting Zeus via its own flaws," *Comput. Networks*, 2013.

[9] A. Zimba and D. Kunda, "Modeling of ICS/SCADA Crypto-Viral Attacks in Cloud-Enabled Environments," 2020.

[10] T. Ahmad, "Corona Virus (COVID-19) Pandemic and Work from Home: Challenges of Cybercrimes and Cybersecurity," *SSRN Electron. J.*, 2020.

[11] S. Morgan, "Global Cybersecurity Spending Predicted To Exceed $1 Trillion From 2017-2021," *Cyber-Crime Magazine*, Californoa, p. 1, Jun-2019.

[12] K. Bissell, R. Lasalle, and P. Dal Cin, "2019 Cost of Cybercrime Study | 9th Annual | Accenture," *Ninth Annual Cost of Cybercrime Study*, 2019. .

[13] R. Rishabh RB, "Flaws in E-Banking: A Prey to Cyber Hunters," *Natl. J. Cyber Secur. Law*, vol. 1, no. 2, pp. 8–15, 2019.

[14] M. J. Haber and M. J. Haber, "Privileged Attack Vectors," in *Privileged Attack Vectors*, 2020.

[15] A. K. Sood and R. J. Enbody, "Crimeware-as-a-service-A survey of commoditized crimeware in the underground market," *Int. J. Crit. Infrastruct. Prot.*, 2013.

[16] O. of F. Research, "Cybersecurity and Financial Stability: Risks and Resilience," *Off. Financ. Res.*, 2017.

[17] M. K. H. Hamid Uddin, Hakim Ali, "Cybersecurity hazards and financial system vulnerability: a synthesis of literature," *Risk Manag.*, vol. 22, no. 3, 2020.

[18] P. Dzhaparov, "Cyber risks – the big challenge facing banks," *Econ. Comput. Sci.*, vol. 1, pp. 6–18, 2020.

[19] A. K. Marnerides, P. Spachos, P. Chatzimisios, and A. U. Mauthe, "Malware detection in the cloud under Ensemble Empirical Mode Decomposition," in *2015 International Conference on Computing, Networking and Communications, ICNC 2015*, 2015.

[20] C. G. and M. van E. Samaneh Tajalizadehkhoob, Hadi Asghari, "Why them? Extracting intelligence about target selection from Zeus financial malware," in *13th Annual Workshop on the Economics of Information Security*, 2014, pp. 1–26.

[21] A. Mohaisen and O. Alrawi, "Unveiling zeus automated classification of malware samples," in *WWW 2013 Companion - Proceedings of the 22nd International Conference on World Wide Web*, 2013.

[22] A. Bouveret, "Estimation of losses due to cyber risk for financial institutions," *J. Oper. Risk*, 2019.

[23] D. Kiwia, A. Dehghantanha, K. K. R. Choo, and J. Slaughter, "A cyber kill chain based taxonomy of banking Trojans for evolutionary computational intelligence," *J. Comput. Sci.*, 2018.

[24] N. Etaher and G. R. S. Weir, "Understanding the Threat of Banking Malware," *Proc. Cyberforensics*, 2014.

[25] N. Etaher, G. R. S. Weir, and M. Alazab, "From ZeuS to zitmo: Trends in banking malware," in *Proceedings - 14th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, Trust Com 2015*, 2015.

[26] M. Eslahi, R. Salleh, and N. B. Anuar, "MoBots: A new generation of botnets on mobile devices and networks," in *ISCAIE 2012 - 2012 IEEE Symposium on Computer Applications and Industrial Electronics*, 2012.

[27] and S. Z. Michele Carminati, Luca Santini, Mario Polino, "Evasion Attacks against Banking Fraud Detection Systems," in *23rd International Symposium on Research in Attacks, Intrusions and Defenses*, 2020, pp. 285–300.

[28] M. Shankarapani and S. Mukkamala, "Anatomy of banking trojans-zeus crimeware (how similar are its variants)," in *6th International Conference on Information Warfare and Security, ICIW 2011*, 2011.

[29] "The FBI vs. GameOver Zeus: Why The DGA-Based Botnet Wins," *Lital Asher-Dotan*, 2015. [Online]. Available: https://www.cybereason.com/blog/the-fbi-vs-gameover-zeus-why-the-dga-based-botnet-wins. [Accessed: 10-Sep-2020].

[30] M. Schwartz, "Gameover Zeus Trojan Continues Resurgence," *Malware Variants Steam Ahead After "Operation Tovar" Takedown*. [Online]. Available: https://www.bankinfosecurity.com/gameover-a-7237. [Accessed: 13-Sep-2020].

[31] Dan Goodin, "Zeus bot found using Amazon's EC2 as C&C server," *The Register*, 2009. [Online]. Available: https://www.theregister.co.uk/2009/12/09/amazon_ec2_bot_control_channel/. [Accessed: 10-Nov-2019].

[32] K. Dahbur, B. Mohammad, and A. B. Tarakji, "A survey of risks, threats and vulnerabilities in cloud computing." ACM, Amman, Jordan, p. 12, 2011.

[33] A. Hutchings, R. G. Smith, and L. James, "Cloud computing for small business: Criminal and security threats and prevention measures," *Trends Issues Crime Crim. Justice*, vol. 456, no. 1, pp. 1–8, 2013.

[34] C. Babcock, "Zeus Bot Appears in EC2 Cloud, Detected, Dismissed," *InformationWeek*, 2009. [Online]. Available: https://www.informationweek.com/cloud/zeus-bot-appears-in-ec2-cloud-detected-dismissed/d/d-id/1085531. [Accessed: 09-Sep-2019].

[35] D. Sullivan, "Beyond the Hype: Advanced Persistent Threats," *Essentials Ser. Adv. Persistent Threat. Real-Time Threat Manag.*, 2010.

[36] W. P. and L. J.C, "Threat Analysis of Cyber Attacks with Attack Tree+," *J. Inf. Hiding Multimed. Signal Process.*, vol. 5, no. 4, 2013.

[37] Dr Mark Scanlon and Dr Nhien-An Le-Khac, "Proceedings of the 16th European Conference on Cyber Warfare and Security ECCWS 2017," in *Proceedings of the 16th European Conference on Cyber Warfare and Security ECCWS 2017*, 2017.

[38] J.-S. L. Ci-Bin Jiang, "Exploring Global IP-Usage Patterns in Fast-Flux Service Networks," *J. Comput.*, vol. 12, no. 4, pp. 371–380, 2017.

[39] M. Korolov, "GameOver ZeuS criminals spied on Turkey, Georgia, Ukraine and OPEC," *CSO Online*, 2015. [Online]. Available: https://www.csoonline.com/article/2961065/cyber-attacks-espionage/gameover-zeus-criminals-spied-on-turkey-georgia-ukraine-and-opec.html. [Accessed: 10-Oct-2019].

[40] P. Black and J. Opacki, "Anti-analysis trends in banking malware," *2016 11th International Conference on Malicious and Unwanted Software, MALWARE 2016*. IEEE, New York, pp. 1–7, 2017.

[41] A. Caglayan, M. Toothaker, D. Drapeau, D. Burke, and G. Eaton, "Behavioral analysis of botnets for threat intelligence," *Inf. Syst. E-bus. Manag.*, 2012.

[42] A. Gezer, G. Warner, C. Wilson, and P. Shrestha, "A flow-based approach for Trickbot banking trojan detection," *Comput. Secur.*, 2019.

[43] M. Sandee, "Gameover Zeus: Backgrounds on the bad guys and the backends," 2015.

[44] NIST, "Common Vulnerability Scoring System Calculator," *NIST* , 2020. .

[45] MITRE, "CVE - Common Vulnerabilities and Exposures," *Common Vulnerabilities Expo.*, 2016.

[46] MITRE, "CVE - Common Vulnerabilities and Exposures," *Common Vulnerabilities Expo.*, 2016.

[47] FIRST, "Common Vulnerability Scoring System v3.0: Specification Document," *Forum of Incident Response and Security Teams (FIRST)*. 2015.

[48] A. Zimba, H. Chen, and Z. Wang, "Bayesian network based weighted APT attack paths modeling in cloud computing," *Futur. Gener. Comput. Syst.*, vol. 96, 2019.

[49] T. Gaidosch, F. Adelmann, A. Morozova, and C. Wilson, "Cybersecurity Risk Supervision," *Dep. Pap. / Policy Pap.*, vol. 19, no. 15, Sep. 2019.

[50] B. Donohue, "Chthonic Zeus Variant Targeting Online Bank Users Globally," *Kaspersky Daily*. [Online]. Available: https://www.kaspersky.com/blog/new_chthonic_zeus_malware/7062/. [Accessed: 02-Oct-2020].

[51] V. Zakorzhevsky, . "Kaspersky Securelist (July 2010). Zbot and CVE2010-0188," *Kaspersky*, 2010. [Online]. Available: https://securelist.com/zbot-and-cve2010-0188/29619/.

[52] L. Watkins, C. Kawka, C. Corbett, and W. H. Robinson, "Fighting banking botnets by exploiting inherent command and control vulnerabilities," in *Proceedings of the 9th IEEE International Conference on Malicious and Unwanted Software, MALCON 2014*, 2014.

[53] S. O. R, Ragan, "Cloudbots: Harvesting crypto coins like a botnet farmer," *BlackHat USA*, 2014.

[54] F. Arnold, H. Hermanns, R. Pulungan, and M. Stoelinga, "Time-dependent analysis of attacks," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014.

[55] S. GOUTAL, "Methods and systems for phishing detection," US9398047B2, 2016.

[56] E. J. Williams, J. Hinds, and A. N. Joinson, "Exploring susceptibility to phishing in the workplace," *Int. J. Hum. Comput. Stud.*, 2018.

[57] A. D. Broido and A. Clauset, "Scale-free networks are rare," *Nat. Commun.*, 2019.

[58] Rajendra Gupta, Piyush Kumar Shukla,"Experimental Analysis of Browser based Novel Anti-Phishing System Tool at Educational Level", International Journal of Information Technology and Computer Science, Vol.8, No.2, pp.78-84, 2016.

[59] Muhammad Iqbal, Malik Muneeb Abid, Mushtaq Ahmad, Faisal Khurshid,"Study on the Effectiveness of Spam Detection Technologies", International Journal of Information Technology and Computer Science, Vol.8, No.1, pp.11-21, 2016.

[60] Engels Rajangam, Chitra Annamalai,"Graph Models for Knowledge Representation and Reasoning for Contemporary and Emerging Needs – A Survey", International Journal of Information Technology and Computer Science, Vol.8, No.2, pp.14-22, 2016.

[61] Shun-Li Lou, Xu-Hua Yang,"Random Connection Based Scale-free Networks", International Journal of Information Technology and Computer Science, vol.5, no.6, pp.10-15, 2013.

[62] Pantic N, Husain MI. "*Covert botnet command and control using twitter"*. In Proceedings of the 31st annual computer security applications conference 2015 Dec 7 (pp. 171-180).

Authors' Profiles

Aaron Zimba is lecturer at Mulungushi University. He obtained his PhD in Network and Information Security at the University of Science and Technology Beijing in the Department of Computer Science and Technology. He received his Master and Bachelor of Science degrees from the St. Petersburg Electrotechnical University in St. Petersburg in 2009 and 2007 respectively. He is also a member of the IEEE. His main research interests include Network and Information Security, Network Security Models, Cloud Computing Security and Malware Analysis.

How to cite this paper: Aaron Zimba, "A Bayesian Attack-Network Modeling Approach to Mitigating Malware-Based Banking Cyberattacks", International Journal of Computer Network and Information Security(IJCNIS), Vol.14, No.1, pp.25-39, 2022. DOI: 10.5815/ijcnis.2022.01.03