A Survey on Advanced Persistent Threats: Techniques, Solutions, Challenges, and Research Opportunities

Adel Alshamrani, *Student Member, IEEE*, Sowmya Myneni, *Student Member, IEEE*, Ankur Chowdhary, *Student Member, IEEE*, and Dijiang Huang, *Senior Member, IEEE*

***Abstract*—Threats that have been primarily targeting nation states and their associated entities have expanded the target zone to include the private and corporate sectors. This class of threats, well known as advanced persistent threats (APTs), are those that every nation and well-established organization fears and wants to protect itself against. While nation-sponsored APT attacks will always be marked by their sophistication, APT attacks that have become prominent in corporate sectors do not make it any less challenging for the organizations. The rate at which the attack tools and techniques are evolving is making any existing security measures inadequate. As defenders strive to secure every endpoint and every link within their networks, attackers are finding new ways to penetrate into their target systems. With each day bringing new forms of malware, having new signatures and behavior that is close to normal, a single threat detection system would not suffice. While it requires time and patience to perform APT, solutions that adapt to the changing behavior of APT attacker(s) are required. Several works have been published on detecting an APT attack at one or two of its stages, but very limited research exists in detecting APT as a whole from reconnaissance to cleanup, as such a solution demands complex correlation and fine-grained behavior analysis of users and systems within and across networks. Through this survey paper, we intend to bring all those methods and techniques that could be used to detect different stages of APT attacks, learning methods that need to be applied and where to make your threat detection framework smart and undecipherable for those adapting APT attackers. We also present different case studies of APT attacks, different monitoring methods, and mitigation methods to be employed for fine-grained control of security of a networked system. We conclude this paper with different challenges in defending against APT and opportunities for further research, ending with a note on what we learned during our writing of this paper.**

Manuscript received March 28, 2018; revised October 25, 2018; accepted December 8, 2018. Date of publication January 9, 2019; date of current version May 31, 2019. This work was supported in part by the Naval Research Laboratory under Grant N00173-15-G017, in part by the National Science Foundation, U.S., under Grant DGE-1723440, Grant OAC-1642031, and Grant SaTC-1528099, and in part by the National Science Foundation, China, under Grant 61628201 and Grant 61571375. The work of D. Huang was supported in part by NSF, in part by ONR, in part by ARO, in part by NATO, and in part by the Consortium of Embedded System. *(Adel Alshamrani and Sowmya Myneni contributed equally to this work.) (Corresponding author: Adel Alshamrani.)*

A. Alshamrani is with the Department of Cybersecurity, College of Computer Science and Engineering, University of Jeddah, Jeddah 23218, Saudi Arabia, and also with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85281 USA (e-mail: asalshamrani@uj.edu.sa).

S. Myneni, A. Chowdhary, and D. Huang are with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85281 USA (e-mail: sowmya.myneni@asu.edu; achaud16@asu.edu; dijiang.huang@asu.edu).

Digital Object Identifier 10.1109/COMST.2019.2891891

***Index Terms*—Advanced persistent threat, APT, targeted attacks, intrusion detection.**

I. INTRODUCTION

THANKS to the strong emphasis on information security on the part of security researchers across the world, security that once was exclusive to military and well-established organizations has now started to become part of every organization. However, this does not suffice as each day we are introduced to a new type of malware, and a new form of attack. There were days when an attacker or a group of attackers goal was to bring down an organization for financial gain or even to prove themselves by damaging the reputation of the company. In all those attacks, the attackers were not trying to hide their actions. There are still these types of attacks, however, there is a different breed of attacks that has become increasingly prominent over the last couple of decades, and this different class of attacks is what this paper is all about. This class of attacks is characterized by slow and low movement of a group of attackers to accomplish their goal, which is usually stealing the target's data without getting caught. The term given to this class of attacks is Advanced Persistent Threats (APT). APT attackers might use familiar methods to break into their target entity's network, but the tools they utilize to penetrate are not familiar. As the term specifies, the tools used are advanced, and they need to be so for an attacker to be persistent in the network for longer periods. They keep themselves low, slowly expanding their foothold from one system to another within the organizations network, gaining useful information as they move and export it to their command and control center in a strategic fashion. APTs are usually performed by well-funded attackers provided with the resources they need to perform the attack for as long as the funding organization needs. The attack only ends when it is detected or when the funding organization gets all the data it needs. Either way, considerable damage would have been done to the organization that was the victim of an APT attack, sometimes irreparable damage, which is most common in the latter case where the attack was not detected until all the organization's data have fallen into the wrong hands. Victim organizations of APT attacks often end up being questioned on their failure to detect the attack even after having security measures such as strong intrusion detection and prevention systems. The answer to this question is what we provide in this paper.

The goal of an APT attack is not only to gather a target entity's data, but also to stay undetected until the attack has been lifted. For this, the well-funded attackers work on creating sophisticated tools such as new types of malware that are not usually detected by signature-based anti-virus software or intrusion detection and prevention systems. They gather every detail about the organization, such as the tools and techniques the organization uses, the applications it hosts, the Anti-Virus software, Intrusion Detection System (IDS), and Intrusion Prevention System (IPS) it uses. Further, they spend time in identifying the vulnerabilities in all these tools and creating malwares that would exploit those vulnerabilities. They then send out these created malwares, often via phishing/spear-phishing attempts, to gain entry into the organization's network.

In an article published in 2013 [1], Mandiant, an American cybersecurity firm reported several key findings of APT attacks performed by one of the largest APT organizations on a broad range of victims for long periods, starting from about 2006, by maintaining an extensive infrastructure of computers across the world. In its M-Trends 2017 report, FireEye points out the increase in the level of sophistication of financial attackers that is no longer any lower than advanced state-sponsored attacks. In support of this, FireEye presented evidence that shows how the attackers evaded detection by IDS/IPS with the use of backdoors that were loaded even before the operating system was loaded. With every passing year, the number of APT attacks being reported has been increasing. All this advancement in the attack methods and tools repeatedly point out the need for deployment of strong defense methodologies by every organization that wants to protect itself and its data. The defense methods should be employed at every phase of an APT attack.

The goal of this survey is to explicitly study various techniques and solutions that were tailored to APT attacks. As such, great emphasis has been placed on a thorough description of the APT stages, and possible attack methods and how attack trees can be used in defending against APT attacks. In addition, through this survey, we intend to point out challenges and research opportunities in defending against APT.

The remainder of this paper is organized as follows: Section II focuses on the definition of APT. Section III discusses various APT attack case studies. Section IV describes the individual methods and related papers for APT defense methods. Section V discusses the evaluation methodologies of APTs solutions. Sections VI and VII elaborate on current challenges in defending against APT attacks, and possible research opportunities, respectively. Section VIII provides a discussion and comparison of our work with existing surveys. Finally, Section IX concludes the paper.

II. Advance Persistent Threats

A. What Is APT?

Advanced Persistent Threat, as the name itself implies, is not like a regular attack or attack done by a regular hacker. APTs are achieved often by a group of advanced attackers that are well-funded by an organization or government to gain crucial information about their target organization or government. APT is a military term adapted into the information security context that refers to attacks carried out by nation-states. APT is defined by the combination of three words, [2], which are:

Advanced: APT attackers are usually well-funded with access to advanced tools and methods required to perform an APT attack. These advanced methods include the use of multiple attack vectors to launch as well as to keep the attack going.

Persistent: APT attackers are highly determined and persistent and they do not give up. Once they get into the system, they try to stay in the system for as long as they can. They plan for the use of several evasive techniques to elude detection by their target's intrusion detection systems. They follow "low and slow" approach to increase the rate of their success.

Threat: The threat in APT attacks is usually sensitive data loss or impediment of critical components or mission. These are rising threats to many nation entities and organizations that have advanced protection systems guarding their missions and/or data.

According to National Institute of Standards and Technology (NIST) [3], an APT attacker: (i) pursues its objectives repeatedly over an extended period of time; (ii) adapts to defenders efforts to resist it; and (iii) is determined to maintain the level of interaction needed to execute its objectives. These objectives are exfiltration of information or undermining or impeding critical aspects of a mission or program through multiple attack vectors.

To achieve the assigned goal, the attackers have to go through multiple stages of attacks in different forms while staying undetected. These multiple stages involve establishing footholds, internal network scanning, and moving laterally from one system to another in the network to reach the target system and perform their detrimental activity. Following the detrimental activity, the attackers might choose to stay to continue their malicious activities on other systems in the network or leave the system after cleaning up; depending on the funding source's requirements. These multiple stages often involve getting into one of the systems within the network and then performing privilege escalations as necessary to reach the target system, followed by accessing sensitive systems and sending the status/information over an Internet connection to the attackers' command and control center.

Chen *et al.* [4], summarized the major differences between APT attacks and traditional attacks in different aspects as shown in Table I.

B. What Is NOT APT?

Advanced Persistent Threats are often misunderstood and the term is increasingly being used in industry as an excuse for organizations' failure to protect themselves from what other wise is a targeted attack. On the other hand, lately, as explained in Section III, attacks have been recorded with goals that are not really specified by NIST under APT, but the methods used and the deterministic characteristics of those

TABLE I
COMPARISON OF TRADITIONAL AND APT ATTACKS [4]

	Traditional Attacks	**APT Attacks**
Attacker	Mostly single person	Highly organized, sophisticated, determined, and well-resourced group
Target	Unspecified, mostly individual systems	Specific organizations, governmental institutions, commercial enterprises
Purpose	Financial benefits, demonstrating abilities	Competitive advantages, strategic benefits
Approach	Single-run, "smash and grab", short period	Repeated attempts, stays low and slow, adapts to resist defenses, long term

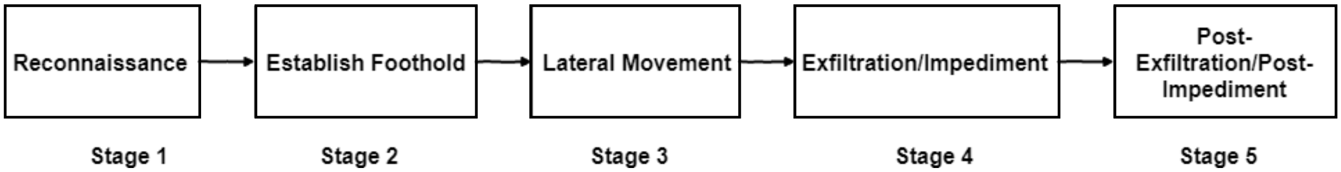

Fig. 1. APT attack model.

attacks made the security industry point out the need to revise the definition of Advanced Persistent Threats to include other domains with new attack goals. We constructed a list of criteria that would establish if an attack is APT or a normal targeted breach in the given environment. If the answer to any of these below relative criteria is *true* for the attack case in question, then that attack is not an APT attack, but a targeted attack.

This attack could have been prevented in more than one way - Given the attack process and the target environment, if the attack is not a surprise and is highly possible (relative to the target environment), then it should have been prevented with minimal countermeasures and security controls in place.

This attack did not require a great deal of adaptation by attackers - If attackers' attempts, to achieve their goal, do not need high adaption or intense evasive techniques to the defender's attempts, then the defense system of the target environment needs to be questioned.

This attack did not exhibit any novelty in its variants - A novelty in the attack methods or techniques like never before is what often makes an APT attack successful. If there is nothing new in the attack methods or techniques, then it is supposed to be detected with existing tools and techniques.

C. APT Attack Model: How APT Attacks Are Made?

APT attacks, as mentioned earlier, are well planned and highly organized towards increasing the probability of the attack's success. To be successful, they perform attacks in multiple stages. To answer the *how* part of an APT attack, we take the help of APT Attack Tree depicted in Figure 2. Schneier [5] describes attack trees and their effectiveness in evaluating the security of a system. With the goal of the attack as the root node, a properly constructed attack tree, though not limited to this function, can give information on the assumptions of the security of the system and the attacks that are likely to happen. These trees can help defenders to place necessary security measures to detect/prevent different components of the attack by automating the correlation of events reported within the system and the probability of those being part of an attack in progress. Giura and Wang [6] have presented an attack tree for APT attacks, and unlike theirs, our attack tree is generic and can be applied to different goals of the threat actors. By having the sub-trees represent different planes, the authors pointed out the correlation needed between those planes. However, the attack methodologies have changed, and a separation of multiple vectors such as physical and digital will only complicate the detection system model, as the dependencies among them is lost. Our simple attack tree structure considers the dependencies across multiple vectors of attack, and we believe our APT attack tree will help defenders in identifying how far an attacker is from achieving the attack goal, and accordingly take a reactive or proactive approach for mitigating the attack. Figures 3, 4, and 5, are special cases of our generic attack tree where we have identified the different attack vectors and their combinations required by APT attackers in order to achieve their goals. The 3 goals depicted were defined by NIST in [3] and are *Steal Organization Data*, *Undermine Organization's Critical Aspects*, and *Position for Future*. The rectangular nodes represent a collection of one or more actions, with the topmost rectangular node in each stage being the goal of that stage, while elliptical nodes are the actions that the attackers can perform to achieve their goal in each of those stages. With a threat score assigned to each of the actions (leaf-nodes) taken by the attackers as found through alerts, followed by correlating those alerts, defenders will be able to estimate the risk and response that is needed to mitigate the threat. The topmost node (root-node) in each of the three attack trees represents the assigned goal of the attacker for the chosen target.

Figure 1 depicts APT attack model given the goal of stealing organization data. It is not necessary that these stages are found in every APT attack model. In [1], Mandiant has discussed it's APT attack life cycle model consisting of 7 stages - Initial Compromise (1), Establish Foothold (2), Escalate Privileges (3), Internal Reconnaissance (4), Move Laterally (5), Maintain Presence (6) and Complete Mission (7) with stages 3 through 6 happening in any order. Ussath *et al.* [7] have discussed a 3 stage APT attack life cycle model focusing only on the representative characteristics of an APT attack. The 3 stages discussed by the authors are Initial Compromise (1), Lateral Movement (2) and, Command & Control Activity (3). Other modified versions of the APT attack life cycle model have been proposed in literature. While all these attack models are similar in terms of the operations involved in APT attacks, they are either too generalized or too specific. Addressing this, we have categorized APT attacks into 5 stages that could represent every APT attack irrespective of the goal while showing how the goal can change the stages involved as below.

Stage 1: Reconnaissance - Reconnaissance marks the beginning of any successful attack. The more attackers understand about the target, the higher their rate of success.

Stage 2: Establish Foothold - This stage represents the attackers' successful entry into their target's computer and/or computer network. In order to achieve their goal they need to establish a foothold in the target's network.

Stage 3: Lateral Movement/Stay Undetected - If the attackers' goal is to undermine critical components or to steal organizational data, they would need to laterally move within the target's network in search of those components or data.

Stage 4: Exfiltration/Impediment - When the attackers' goal is to get organizational data, actions comprising retrieving and sending this data to the attackers' command and control center fall under this stage. In addition, when the attackers' goal is to undermine critical components, actions comprising disabling or destroying the critical components of that target organization will fall under this stage.

Stage 5: Post-Exfiltration/Post-Impediment - This stage involves post-exfiltration/post-impediment activities such as continuing to exfiltrate or disable more critical components or delete evidence for a clean exit from the organization's network.

For any of the 3 APT goals, the first 2 stages are necessary for the attackers to go through, in order to increase their probability of success. These stages as explained later in this section are Reconnaissance, and Establishing Foothold. The other 3 stages are applicable based on the attackers' goal. If the goal of the attackers is to steal the organization's data or undermine critical aspects of the organization, the attackers would have to move laterally within the organization's network in search of data resources or critical components respectively and to gather information that will help them in progressing their attack. Differences in these 2 goals can be seen during the stage 4 and 5. While attackers with goal to steal the organization data involve in data exfiltration activities, attackers with goal to imped critical components involve in bringing down critical aspects of the organization. On the other hand, attackers with goal to position for future take a different path in stage 3 where they keep themselves updated with the changes happening within the organization's network, studying and understanding the working of the system and the users, thus gaining as much information as they can while staying unnoticed. APT attackers with goal to position for future do not involve in stage 4 and 5 unless their goal is changed to either to steal organization data or to undermine critical aspects. In the later parts of this section we explain in detail each of these stages and the multiple vectors that attackers can use in each of those stages.

Stage 1: One of the first steps attackers take is to learn about their target. The more they understand the target, the more successful they may be with their attack. As part of this phase, attackers extensively research about their target, gathering necessary information and intelligence of the organizations' assets towards increasing their rate of success. This information includes but not limited to the details of the employees such as their social life, habits, and websites they often visit. Further, details of the underlying IT infrastructure, such as the types of switches, routers, anti-virus tools, firewalls, Web servers used, ports open, etc. help the attackers not only to establish a foothold, but also to penetrate deeper into the target's network. Gathering information, as shown in our APT attack tree, usually involves social engineering techniques, reconnaissance performed on site, port scanning, and service scanning, which refers to psychological manipulation of people into accomplishing goals that may or may not be in the targets best interest [4]. In addition, APT campaigns query publicly available repositories, using "who is" [8], and Border Gateway Protocol (BGP) looking for domain and routing information, finding websites on the targeted network that have high-risk vulnerabilities, such as cross-site scripting (XSS) and SQL injections (SQLI), and fingerprinting organizational networks to check for opened ports, address ranges, network addresses, active machines, firewalls, IDS/IPS, running software, access points, virtual hosts, outdated systems, virtualized platforms, storage infrastructure, and so on, to decipher the networks layout [9]. In APT attacks, reconnaissance usually is passive, as attackers do not exploit a victim, but instead are collecting data in preparation for the attack. Once APT actors have collected enough information, they construct an attacking plan and prepare the necessary tools.

Stage 2: Collected information from the previous stage, as shown in our APT attack tree, can be used to exploit vulnerabilities found in the target organization's Web applications or to exploit vulnerabilities in end user systems via malware execution. Below we explain the different methods and techniques that APT attackers use in this stage.

A) Exploitation of Known Application Vulnerabilities: Exploitation of known vulnerabilities is another source that APT attackers utilize to perform APT attacks. Known vulnerabilities are usually exposed and can be obtained from well-known vulnerability databases such as Common Vulnerabilities and Exposures List (CVE), and NIST National Vulnerability Database (NVD) [10] which publicly disclosed vulnerabilities where each vulnerability is identified using an unique CVE-ID. In addition, in some cases attackers can share and collect useful information about found vulnerabilities in dark-Web and deep-Web forums [11]. According to the reported study in [7], majority of APT attacks were based on known exploits. Therefore, it is essentially important to apply security patches shortly after vulnerabilities have been released.

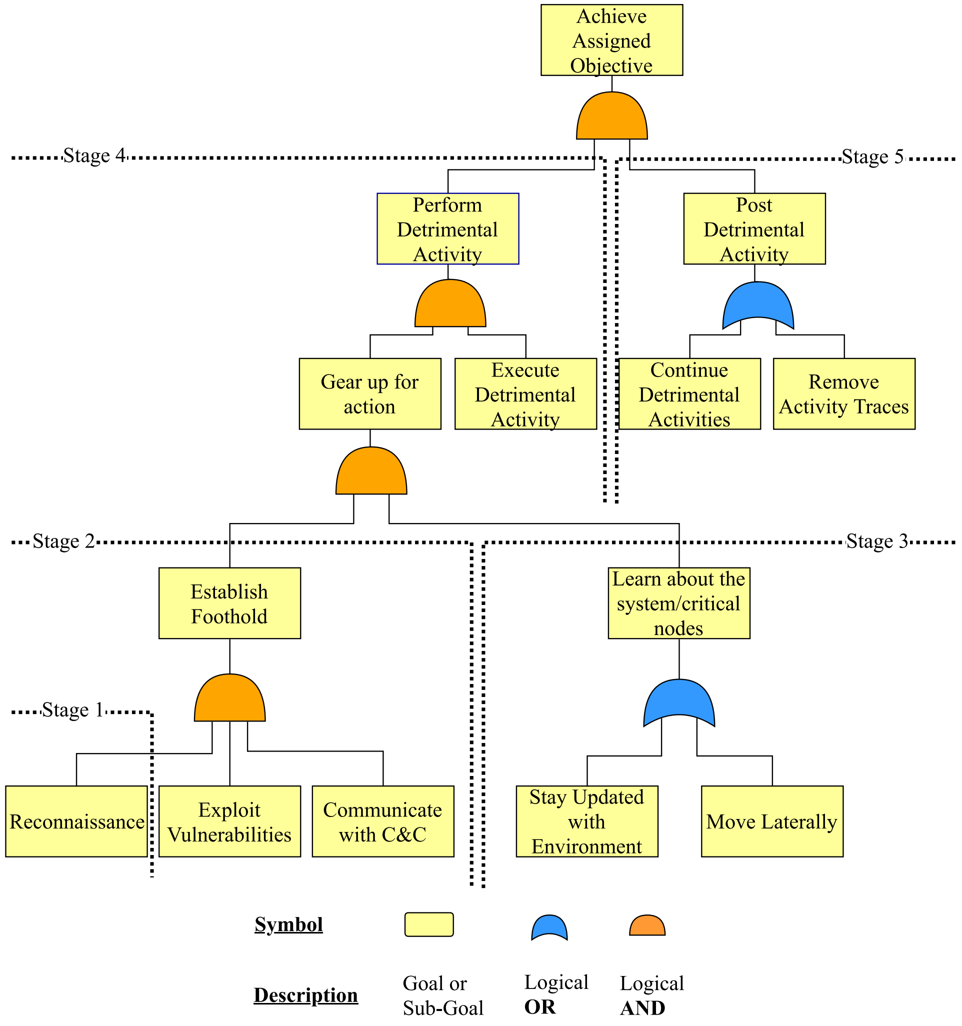

Fig. 2. APT Attack Tree.

B) Malware: According to Symantec's 2017 Internet Security Threat Report there were 357 million malware variants in the year 2016. The increasing rate of receiving malware over emails has significantly increased from 1 in 220 emails in 2015 to 1 in 131 emails in 2016. Symantec attributes this increase in malware to the botnets that deliver the spam campaigns. As shown in our APT Attack Trees Figures 3, 4, and 5, malware can be sent via spear-phishing, USB devices, and/or Web downloads.

C) Spear-Phishing: In the same threat report, Symantec also reported that targeted spear-phishing campaigns specifically in the form of business email compromise scams, are being favored by attackers instead of the old mass-mailing phishing campaigns. This starts with the attackers performing social engineering or other such techniques to gain information about the organization and then sending out emails with malware in it. These fraudulent emails are cleverly crafted, well enough to entice the targeted recipients to open the attachments. Employees unaware of the malware might risk the organization's network by opening the attachment or link that leads to installation and execution of malware. This malware when executed might exploit either known or unknown vulnerabilities to establish a foothold in the organizational network. Figure 6 shows an illustration of spear phishing example.

In [1], an APT actor sent a spear-phishing email to a Mandiant's employee in which the email seemed to be sent by the Mandiant's CEO. The APT actor created a specific email's account using real-name (Mandiant's CEO name). The email contained a malicious ZIP file for the goal of installing an executable backdoor "WEBC2-TABLE".

Smadi *et al.* [12] proposed a machine learning model that used J48 decision tree classifier to detect phishing emails. Their model was trained on 23 features generated from an emails header and body. These features contain message ID domain, sender domain, message type, and number of links and characteristics of URLs in links. The J48 classifier was evaluated for a combined dataset of 4559 phishing emails and 4559 legitimate emails using 10-fold cross validation. They achieve 98.11% accuracy and 0.53% false positive rate.

D) Zero-day vulnerability: A zero-day vulnerability is a software bug that either the software manufacturer is unaware of, or is aware of but was not able to fix before the attackers could utilize it. APT attackers gather information about

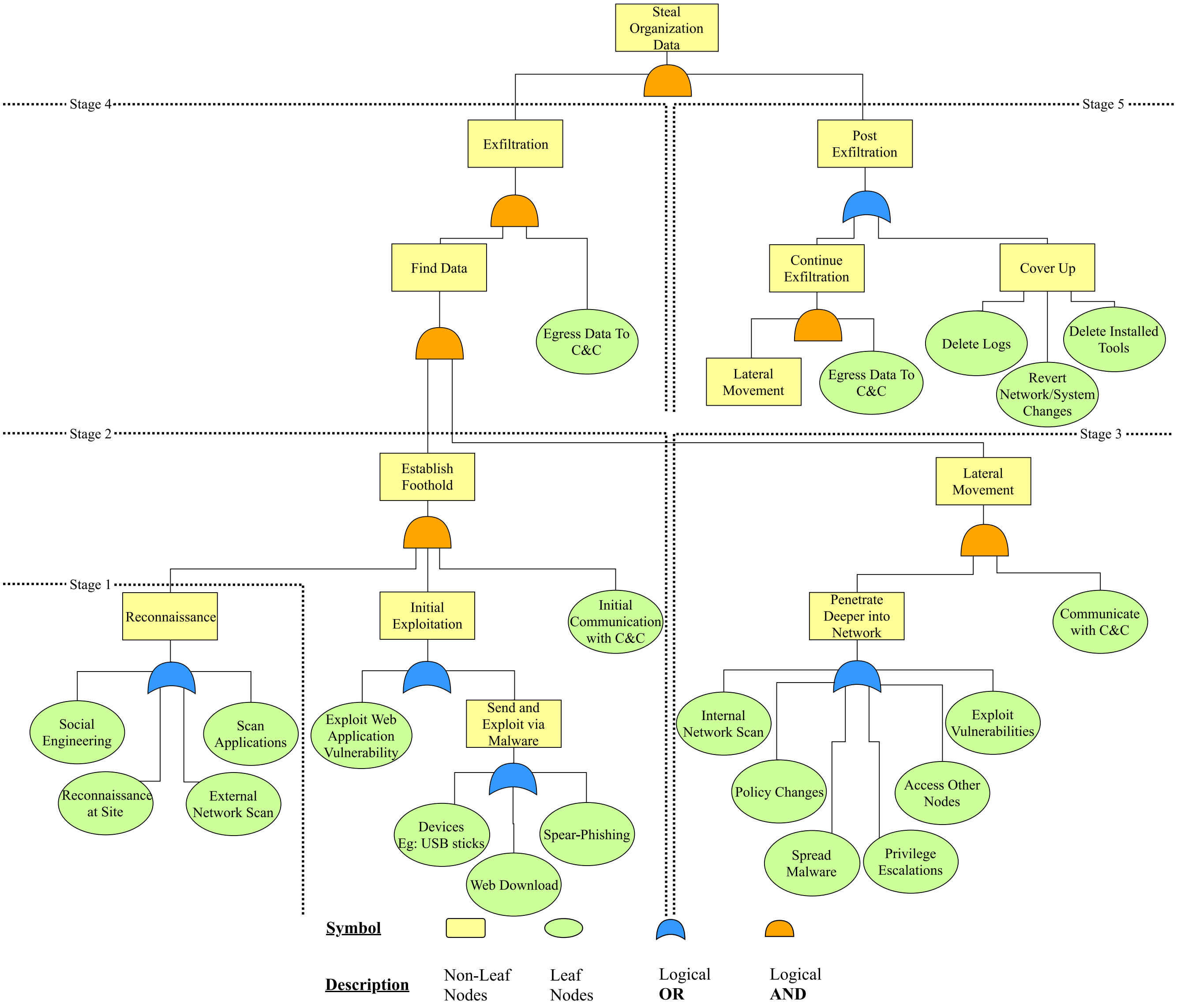

Fig. 3. APT Attack Tree - Case When Attackers Goal is to Steal Organization Data.

the organization's system components, such as the operating system versions, patches ran, and software components installed on those systems, including anti-virus and anti-malware components. They then go about identifying any vulnerabilities in those versions that could be utilized to gain entry into the target's network. However, according to the reported study in [7], only a few APT attacks were performed and achieved through zero-day vulnerability, and the majority of APT attacks were based on known exploits.

Lee and Lewis [13] have focused on APT achieved through malware sent via emails. The authors have examined several emails with binaries and have come up with a solution involving constructing an undirected graph where nodes of the graph represent the email addresses and edges correspond to the exchange of email messages that connect the nodes with an aim that this graph would give them further helpful information in analyzing the targeted malware. However, the problems with this solution is that there could be several attack nodes without any links to other attacks which could mean that there was not enough visibility of the recipients of the attacks or those could be unique attacks that warrant further investigation before concluding with the existence of an APT attack.

E) Web Download: As mentioned earlier, spear-phishing emails could have malicious files attached to them that need to be opened, or they could contain links to malicious websites that when employees visit, they unknowingly download malware. Alternatively, attackers can inject malicious code into one of the websites frequently visited by the targeted employees. This latter attack technique is called the watering-hole attack.

F) Watering-Hole Attack: Unlike phishing attacks that involve luring employees to malicious websites, watering-hole attacks involve infecting one of the websites that the target organization's employees frequently visit. As depicted in Figure 7, the attackers use the targeted employees' information gathered in Stage 1, and find vulnerabilities in the websites visited by them towards injecting malicious code into one or more of the vulnerable websites. Once the targeted employee

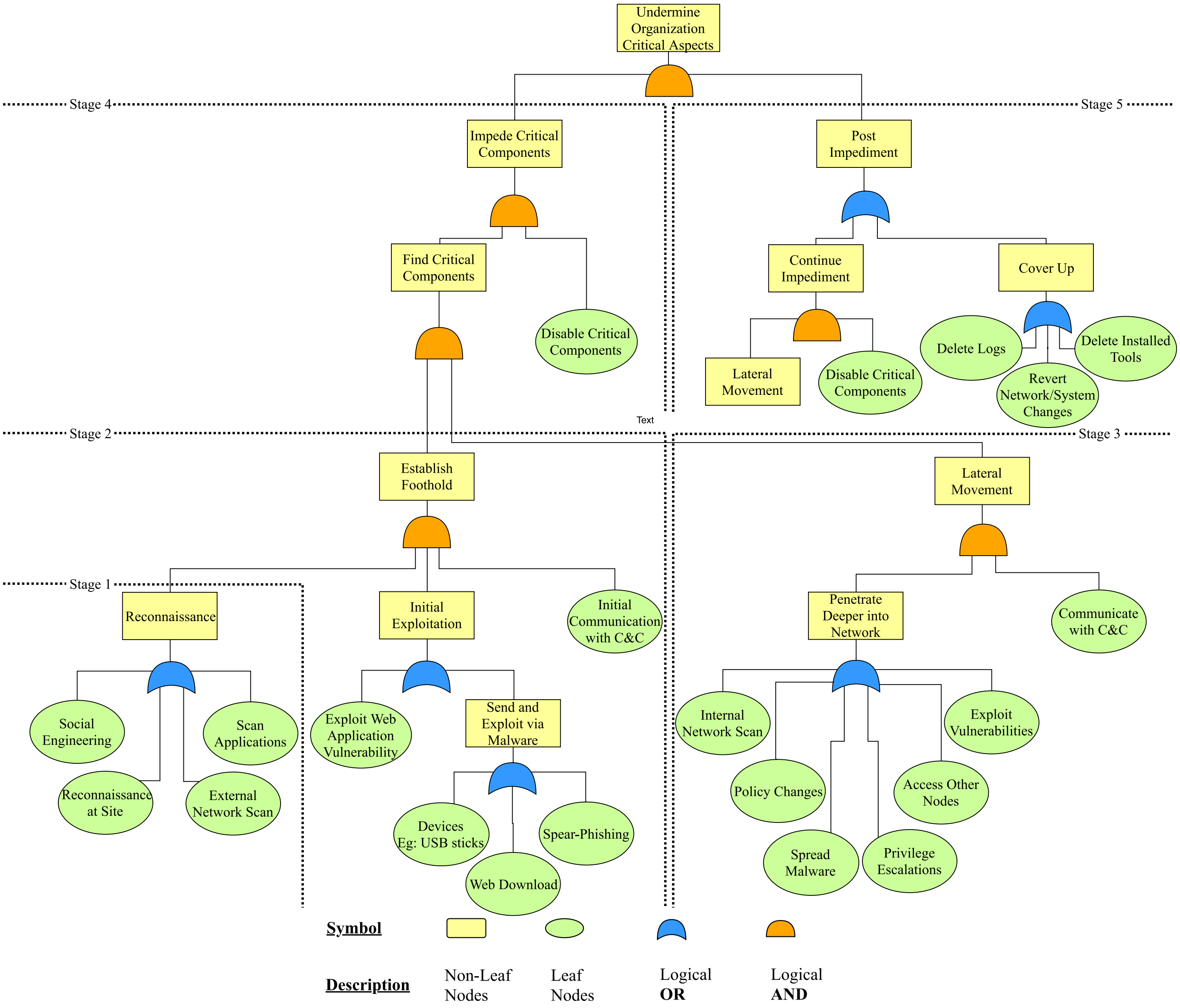

Fig. 4. APT Attack Tree - Case When Attackers Goal is to Undermine Organization's Critical Components.

or group of employees, visit(s) the infected website, the malicious code is downloaded onto the system giving the attackers' access to the system.

After sending out emails with malicious attachments or links to websites with malicious software, attackers patiently wait for the malware to run within the organization's network that would open the gates to the organization's system. The challenge for an APT attacker here is to have the malware run without being detected by the anti-virus tools, and intrusion detection and prevention systems. Once the attackers get control of the system through the malware execution that exploits vulnerabilities in the system, they keep low to go undetected to the next phase. At this point, APT attackers aim to build a Command and Control (C&C) communication channel after infiltrating the targeted network to deploy subsequent attacks. Most malware makes use of Domain Name Systems (DNS) to locate their domain name servers and compromised devices, so APT attackers can establish a long-term connection to victims devices for stealing sensitive data.

Stage 3: Now, once the attacker has gained an access to the targeted system, he/she can spread over to other systems within the target's internal environment. The attacker uses various techniques to access other hosts from a compromised system and get access to sensitive resources. Most often, stolen legitimate credentials are used during this stage. This includes putting malware and other tools on different machines inside the compromised system components and hiding them. Some times, this phase involves privilege escalation, and at other times it involves getting passwords of the users through key loggers. Other times, it could be through pass-the-hash techniques and/or vulnerabilities exploitation. The chosen method depends on the environment of the target system. The goal of the attackers in this phase is to expand their foothold to other systems in search of the data that they want to ex-filtrate. Therefore, once the attacker has reached this advanced stage, it is very difficult to completely push out such attacker out of the environment [7]. Table II shows some techniques and methods used to accomplish lateral movement. Hash and password

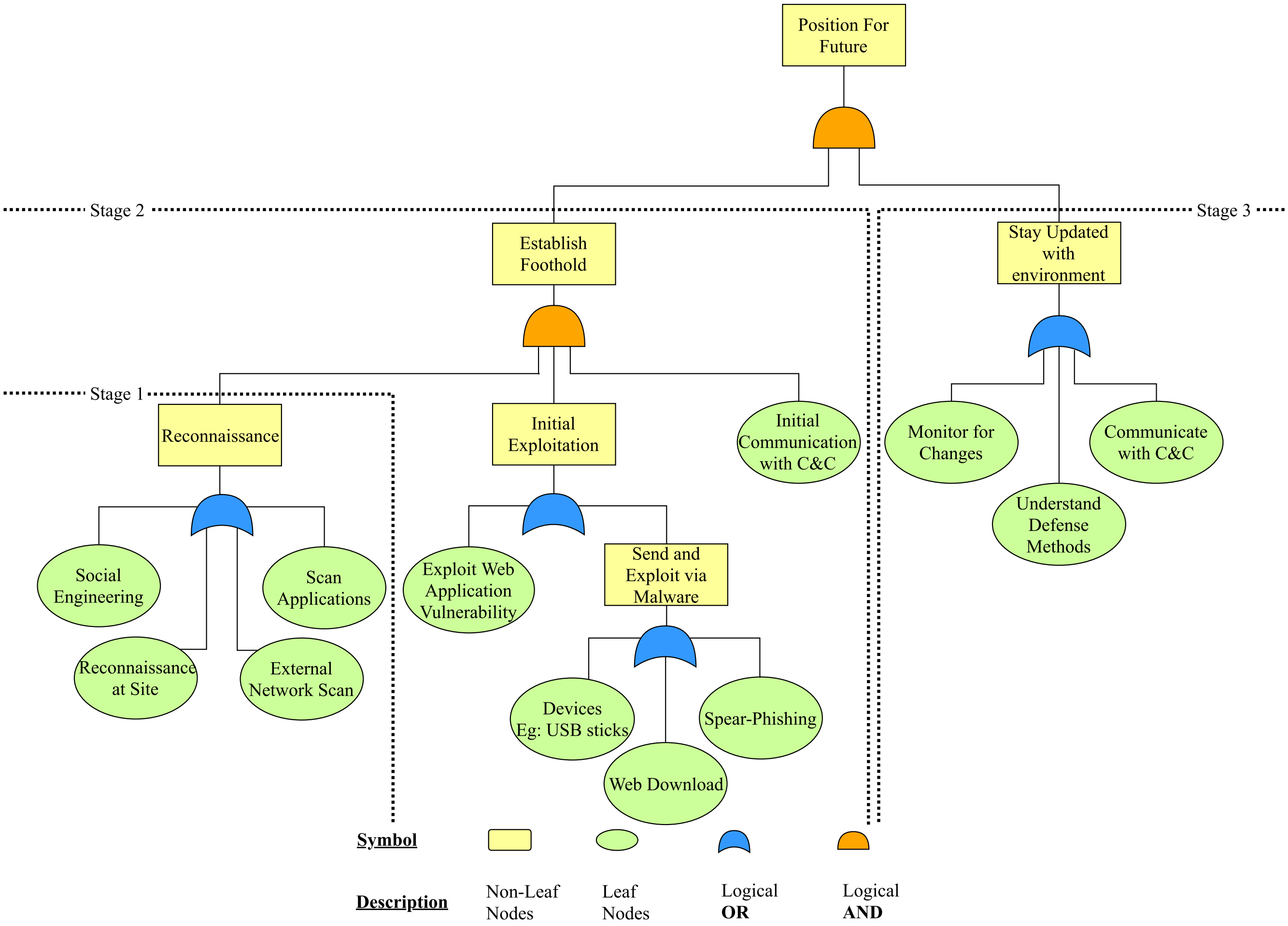

Fig. 5. APT Attack Tree - Case When Attackers Goal is to Position For Future.

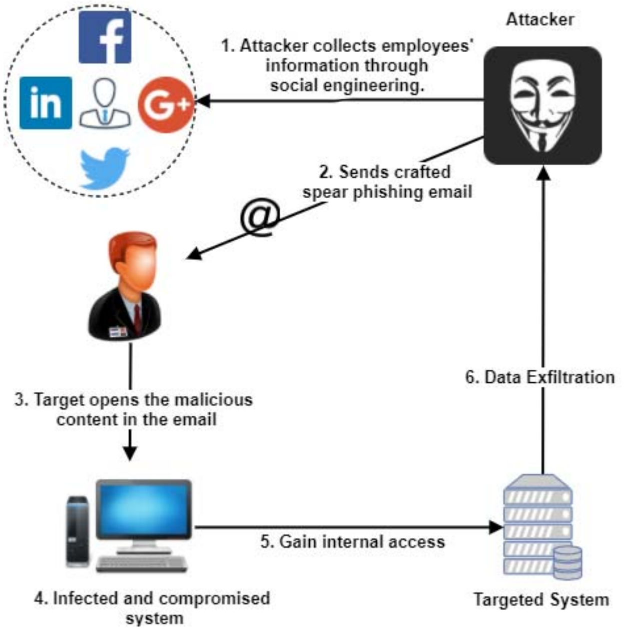
Attacker
1. Attacker collects employees' information through social engineering.
2. Sends crafted spear phishing email
@
3. Target opens the malicious content in the email
4. Infected and compromised system
5. Gain internal access
6. Data Exfiltration
Targeted System

Fig. 6. Spear phishing example.

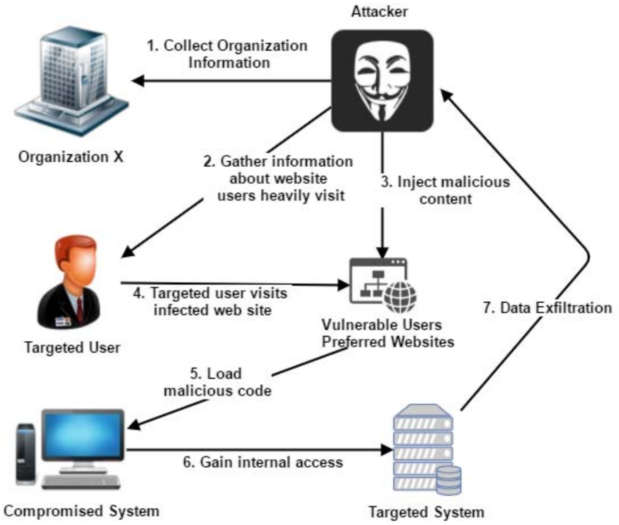
Attacker
1. Collect Organization Information
Organization X
2. Gather information about website users heavily visit
3. Inject malicious content
Targeted User
4. Targeted user visits infected web site
Vulnerable Users Preferred Websites
7. Data Exfiltration
5. Load malicious code
Compromised System
6. Gain internal access
Targeted System

Fig. 7. Watering-hole example.

dumping (credential dumping) is the process of obtaining account login and password information from the operating system and software. Credentials can be used to perform lateral movement and access restricted information. APT attackers can utilize the usage of valid credentials to move stealthily within an environment. The main method of gathering hashes and passwords is to utilize appropriate applications that are able to dump this information from a system. Currently,

TABLE II
TECHNIQUES AND METHODS OF THE APT CAMPAIGNS ([7])

APT Group	Standard OS Tools	Hash and Password Dumping	Exploit Vulnerabilities
MsnMM (Naikon Group)[15]	✓		
Carbanak [16]	✓	✓	
Duqu 2.0 [17]	✓	✓	✓
Naikon APT [18]	✓		
EquationDrug (Equation Group) [19]			✓
Operation Cleaver [20]	✓	✓	✓
Shell Crew [21]	✓	✓	
Icefog [22]		✓	
Regin [23]	✓		
Anunak [24]	✓	✓	✓
Deep Panda [25]	✓	✓	

TABLE III
MOST NOTICEABLE DATA EXFILTRATION INCIDENTS IN 2017

Date	Organization	Number of affected people	What got leaked
19th June	Republican National Committee	200 million	Names, Phone NOs, Home addresses, Voting details, DOB
13th July	Verizon	14 million	Names, Phone NOs, PINS
15th Mar	Dun & Bradstreet (DB)	33.7 million	Email addresses, Contract information
12th Mar	Kansas Department of Commerce	5.5 million	Social Security Numbers
21st Mar	America's Job Link Alliance (AJLA)	4.8 million	Names, DOB, Social Security Numbers
7th July	World Wrestling Entertainment (WWE)	3 million	Names, Earnings, Ethnicity, Address, Age Range
17th July	DOW JONES	2.2 million	Names, Customer IDs, Email Addresses
29th July	Equifax	143 million	Social Security Numbers, Names, Addresses, Drivers licenses
1st Aug	Esports entertainment (ESEA)	1.5 million	Locations, Login details, Email addresses, DOB, Phone NOs

mimikatz is the most widely used hash and password dumping tool because it is able to dump clear text passwords and it also offers further features. Windows Credential Editor (WCE) is another tool that is used by APT attackers to gather valid credentials. Although there are different techniques for dumping Windows credentials, the most common method is to extract and analyze parts of the Windows Local Security Authority (LSA) process [7]. These tools are in use by both professional pen-testers and adversaries. In APT1 [1], during the lateral movement stage, three methods were used: Installation of new backdoors on multiple systems, usage of legitimate VPN credentials, and signing into Web portals.

Stage 4: In case of stealing organization data, the attackers export the data they collected to their command & control server. Since most of the intrusion detection and prevention systems do ingress filtering and not outgress filtering, their data exfiltration could go undetected. Depending upon the organization's defense methodologies, the attackers' might intelligently split the data exfiltration into batches and to servers with different IP addresses. Ullah *et al.* [25] summarize the latest data exfiltration incidents in 2017 as shown in Table III.

Stage 5: The goal of an APT attack is not just performing detrimental activity, but to keep doing so until the attack has been lifted by the attack sponsor. The sponsor could choose to lift the attack once the data retrieved is found to be what is wanted or could keep the attack still active to keep getting data as long as the attackers can. In either case, the attackers would have to cover their tracks so that they leave no clue of themselves or the sponsoring entity. If there is no need for further exfiltration or impediment, any tools installed during the attack, or logs that could give strong evidence are removed as part of this stage.

D. Command and Control (C&C) Communication

As stated earlier, APT attackers need to have an open communication channel between their servers and victims' machines. This is known as (C&C) or (C2) which is an essential component during the lifetime of APT attacks. The C&C communication applies mainstream network services such Hyper Text Transport Protocol (HTTP), HTTP Secure (HTTPS), Internet Relay Chat (IRC), Peer-to-Peer (P2P), custom protocols, and others. HTTP-based connections are preferable over others due to the fact that, first, HTTP-based C&C traffic are labeled as legal in most enterprise, second, other C&C protocols such as P2P and IRC traffic has distinct network features such as ports, and package content, which are easily identifiable and can be blocked [7], [26].

III. APT ATTACKS CASE STUDY

APT attacks that have become prominent over the past decade actually have been reported even before the term APT was coined in the late 2000s. However, in those early times, nation entities were the targets of such advanced and persistent threats, which later started to include different non-nation and non-governmental organizations.

A. Titan Rain

In 2003, a series of coordinated cyber attacks, later codenamed *Titan Rain*, have emerged that infiltrated several computers and networks associated with U.S. Defense Contractors with a goal to steal sensitive data. These were found to continue until the end of 2015, stealing unclassified information from their targets, though no reports of stolen classified information were made. The level of deception involved and the use of multiple attack vectors marked these attacks as the first of their kind.

B. Hydraq

One of the first APT attacks on commercial companies that has drawn great attention was Hydraq, name used in referring to the Trojan that establishes the backdoor, well known under the original name given to this attack, ‘Operation Aurora’. This coordinated attack involved the use of several malware components that are encrypted in multiple layers to stay undetected for as long as they can. The attack found to be launched in 2009 has targeted different organization sectors, Google being one of them and the first one to announce it, followed by Adobe. The name ‘Aurora’ came from the references in the malware that got injected during the malware’s compilation on the attackers machine. The malware was found to use a zero-day exploit in Internet Explorer (CVE-2010-0249 and MS10-002) [27] to establish foothold on the system. When users visited the malicious site, Internet Explorer was exploited to download several malware components. One of the malware components established a backdoor to the machine, allowing attackers to get onto the organization’s network as and when needed. In some of the earlier cases, the malware exploited a vulnerability in Adobe reader and acrobat applications (CVE-2009-1862) to establish foothold on few companies. Unlike the earlier instances, the later instances of these malware were found to no longer use the zero-day vulnerabilities. Nevertheless, the attacks continued for several months after, in different countries across the globe under different variants of the Trojan Hydraq. The common aspect of the trojan is, the malware gathers system and network information initially, followed by collecting usernames and password into a file that is later sent to its command and control center whose IP address or domain name is hard-coded within the malware.

C. Stuxnet

In 2009, a sophisticated worm that spreads itself to other components in the entity with a goal to impede Iran’s uranium nuclear project, had been launched. At first, this malware was found to exploit a zero-day vulnerability found in LNK file of Windows explorer. Microsoft named this malware as *Stuxnet* from a combination of file names found in the malicious code (.stub and MrxNet.sys) after being reported about this zero-day vulnerability. However, it was later found out that in addition to the LNK vulnerability, a vulnerability in printer spooler of Windows computers was used to spread across machines that shared a printer. And then this malware used 2 vulnerabilities in Windows keyboard file and Task Scheduler file to gain full control of the machine by performing privilege escalation. In addition, it used a hard coded password within a Siemens Step7 software to infect database servers with Step7 and from there infect other machines connected to it. After the malware first enters a system, it sends the internal IP and the public IP of that system along with the computer name, operating system of the system, and whether Siemens Step7 software was installed on that machine, to one of its 2 command and control centers running in 2 different countries. Through these command and controllers the attackers either let the malware infect the system or updated the malware with new functionality. It was soon found out that Stuxnet was way beyond control with several computers in different countries being infected with this malware. Two of the zero-days used in Stuxnet were not new in Stuxnet, they have been exploited earlier by other small malwares though were not found at that time. After security researchers across the globe have dug into Stuxnet for several months, it was found out that this malware was way beyond what it looked like and it actually sends commands to programmable logical controllers targeted to impede the Iran’s uranium nuclear project. Several reports were published by researchers and firms across the world, with more or less conflicting information on the detailed execution of Stuxnet as in [28] and [29]. However, they all agree that Stuxnet was found to be like never before, a havoc that a digital code could create in physical world. It was not just all about 4 zero-day vulnerabilities, 2 stolen certificates, and 2 command and control centers, it was more than that, a cleverly crafted, layered piece of malware that could be tweaked by the attackers through the command and control centers using over 400 items in its configuration file. The end date of Stuxnet was found to be in 2012, 3 years after it was unleashed. Though Iran found out the existence of this 500 KB malware in its Natanz plant in 2010, amidst all the havoc of Stuxnet, some of its centrifuges were already damaged, slowing down its nuclear weapon generation process.

D. RSA SecureID Attack

In 2011, RSA, a secure division of EMC Software announced a sophisticated cyber-attack on its systems that involved the compromise of information associated with its SecureID, a 2 factor authentication token product. This is another attack that infiltrated an organization’s network through phishing emails sent to the organization’s employers. As part of this attack, the attackers sent 2 different phishing emails to different groups of employers with an excel sheet attached. The phishing emails went into the junk folder on the employers end, however, they were crafted well enough that an employee opened the attached excel sheet. This excel sheet when opened exploits the zero-day vulnerability (CVE-2011-0609) of adobe flash player to install a backdoor. When the employee opened the aforementioned attachment, the backdoor got installed onto the employee’s system. This

TABLE IV
CASE STUDY ANALYSIS

APT Attack	Date	Goal	Attack Vectors Used
Titan Rain	2003 - 2005	Steal Organization Data	Social Engineering, Backdoors
Hydraq	2009 - 2011	Steal Organization Data	Social Engineering, Phishing, Backdoors, Zero-Day exploits
Stuxnet	2009 - 2012	Impede Critical Components	Malware via USB devices, Zero-Day Exploits, Backdoors
RSA SecureID Attack	2011 - 2011	Steal Organization Data	Spear-Phishing, Zero-Day Exploits, Backdoors
Carbanak	2013 - 2015	Steal Money	Social Engineering, Spear-Phishing, Backdoors, Key Loggers, Form Grabbers, Video Captures of Victim's Activities, Remote Administration Tools

installed backdoor was found to be a variant of a well known remote administration tool that now the attackers could use to remote access the employee's machine. With this remote access in place, the attackers started harvesting credentials of several employees in an effort to reach the target system where they performed privilege escalations, stole the data and files, compressed and encrypted them before sending them to their remote command and control center via ftp. RSA detected this exfiltration but not before some of the data got exfiltrated.

E. Carbanak

Carbanak, unlike the APT attacks discussed earlier, was an attack for stealing money from financial institutions. The attacks started in 2013, with the attackers getting into the internal network of the their target banking/financial institution through spear-phishing attacks, have gone undetected until early 2014. According to [30], emails sent to employees had files attached to them that when executed exploit Microsoft office vulnerabilities (CVE-2012-0158, CVE-2013-3906, and CVE-2014-1761) giving the malicious code in the attachment ability to install backdoor. This backdoor was named Carbanak, supposedly after, Carberp, the malware used as backdoor that was considered to be a variant of a known malware 'Anunak'. Once the attackers established a foothold, they started internal reconnaissance as part of their lateral movement, through key loggers, form grabbers and such which were sent to the attackers C&C server. Researchers found videos of employees activities were captured and sent to the attackers command and control server as part of their internal reconnaissance. The novelty in this attack was the different tools they used, and a custom binary protocol they established to communicate with their C&C servers from the victims machines. It was found that the attackers studied each of their victims through their internal reconnaissance and used attack methods that would specifically apply to that victim. They created fake transactions in the victim's internal database to hide their money transfer transactions. Carbanak seemed to stop in 2015, only to be found later that it continued to show up through 2017 in different variants.

Table IV presents the attack vectors used in the above APT attacks. It can be observed that as the years passed by with more and more digital elements becoming part of the physical world, the level of deception involved and the techniques used have increased in their sophistication. Symantec, in its 2018 Internet Security Threat Report, Volume 23, discussed evolving forms of cyber-crime that tend to be targeted attacks in the mask of ransomware. According to this report, targeted attack groups are using ransomware as decoy to perform detrimental activities on their targets. NotPetya, one of such attacks, was reported by ESET, an IT security company, as the work of one of the evolving APT groups whom they call Telebots. At this rate, in order to defend against APT attacks a defense-in-depth approach that learns to adapt to the attackers' methods needs to be developed, which we discuss in the next section. The case studies point not only to the need for intelligent defense mechanisms, but also to the scope of these APT attacks. Though these APTs have started in nation-state sectors, it did not take much time for the attackers to extend their scope to non-governmental and commercial sectors with goals of stealing corporate data posing the biggest threat to any company with data as their biggest asset. However, the more recent advanced persistent threats point out that organizations with assets other than data such as finance organizations where money is the major asset, are also facing these threats. The Carbanak attack discussed in our case study is one such example.

IV. CLASSIFICATION OF APT DEFENSE METHODS

As mentioned earlier, defending against an APT attack cannot be done by a single tool. A defense-in-depth approach with appropriate defense mechanisms implemented to detect/prevent each stage of an APT attack at multiple points and across multiple levels of the network needs to be employed. Correlation of the events from these different defense measures plays a key role in protecting an organization/entity against APT attacks. This approach of defense-in-depth relies on the fact that even if the attackers could evade detection by one of the several employed defense measures, there is another layer of defense that they should evade detection by. A proper defense-in-depth approach should make sure not all layers of defense measures can be evaded. In addition, these layered defense measures give defenders time and a risk estimate that would help them come up with a mitigation approach to employ.

Yang *et al.* [31] have evaluated the security of cyber networks under advanced persistent attacks. They modeled cyber networks under advanced persistent threats launched by a strategic attacker. They did so by defining the equilibrium of the cyber network as a security metric and evaluated the impact of the attack and defense strategies on this equilibrium metric.

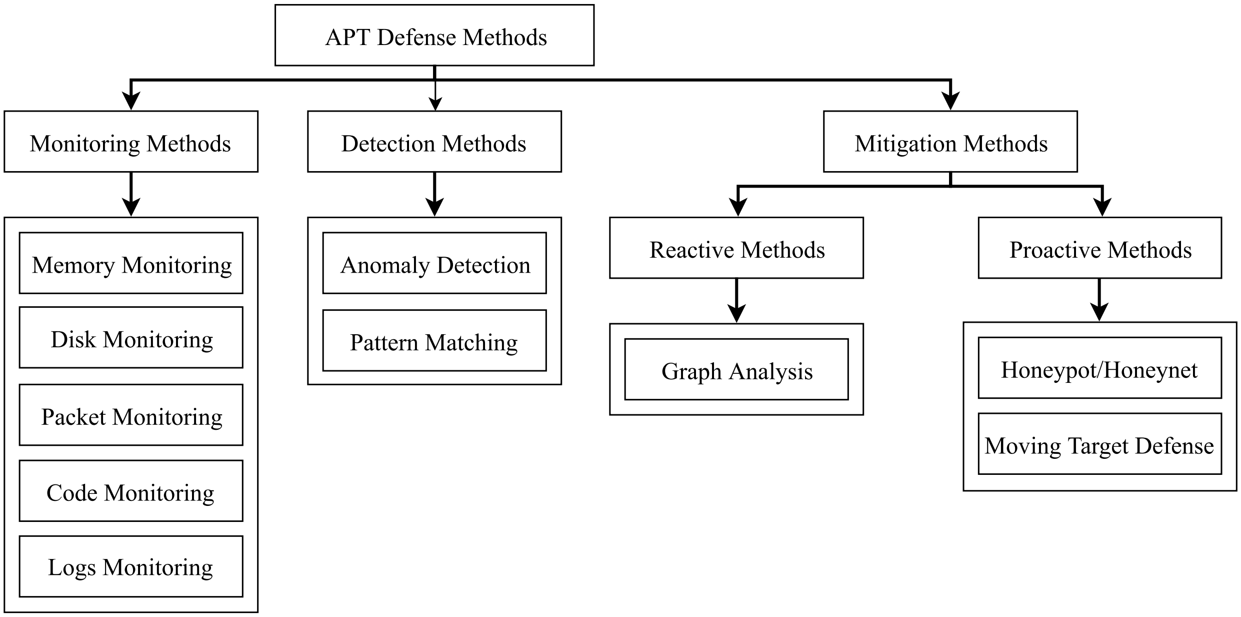

Fig. 8. APT defense method classifications.

They theoretically confirmed that the equilibrium security of a cyber network descends with the increase of the resources per unit time used for attacking a node, and the same ascends with the increase of the resources per unit time used for preventing or recovering a node. In addition, they studied and theoretically analyzed 3 other factors on this equilibrium and reported that the equilibrium security of a cyber network will decline with addition of new edges to the network, the equilibrium security attains maximum when the prevention-resources is close to that of recovery resources, and that this same metric goes up with the increase of the defense resources per unit time. With this analysis they conclude that while cyber networks with dense connections are more vulnerable to APT attacks than those with sparse connections, they recommend distributing the defense resources equally among prevention and recovery as, in the context of APT, recoveries are as important as prevention. Finally they suggest configuring more resources for cyber networks is always an effective means of protecting them against APTs. Below, we have discussed the various defense methods that can be employed to defend against APTs along with their current solutions.

We categorize APT defense methods into 3 major categories, they are *Monitoring Methods, Detection Methods, and Mitigation Methods*. Each category or class can be subcategorized into different categories as shown in Figure 8. In the following subsections, each of these classes will be precisely explained.

A. Monitoring Methods

One of the basic and first steps to defending against APT is to start monitoring the entire network system at multiple points and multiple levels, leaving no entry point un-monitored.

1) Disk Monitoring: Every end system as part of the organization's network needs to be monitored for any malicious behavior through anti-virus, firewall, or content-filtering as necessary. Applying patches as necessary to the software running on the system will help minimize the entry points for an attacker by removing known vulnerabilities that could otherwise spread malware to vulnerable systems within the network. In addition, monitoring CPU usage for each of these end systems within the network will help in identifying any suspicious behavior at the end system level.

2) Memory Monitoring: One of the ways malware can evade detection is by running within the memory of the end system rather than from a file. This so-called fileless malware uses a process that is already running within the memory to execute itself. As there is no separate process running in the background, it leaves no trace except the unexpected memory usage by a process that can be identified if monitored. Duqu 2.0, which infested Kaspersky labs in 2015 [32], ran within the memory of an already running process, and thus bypassed the verification of the caller process that usually happens on their systems.

As each day passes by, new and sophisticated malware is coming into existence. The authors of the paper [33] have portrayed the characteristics of different types of malware, and proposed a solution 'Panorama' that will detect these different types of malware. However, their proposed solution involves gathering malware and benign samples as training data and extracting taint graphs from it. They then transform this taint graph into a feature vector upon which standard classification algorithms are applied to determine a model. This model is then used to identify malicious behavior on a system. Their proposed solution is based on the common characteristics of the different types of malware such as key loggers, password thieves, stealth backdoors, etc. The characteristics commonly exhibited by these malware are anomalous information access and processing behavior. For instance, keyloggers and password thieves intercept the keystroke inputs. In order to stay undetected, stealth backdoors, as observed by the authors, either use an uncommon protocol such as ICMP, create a raw socket, or intercept the network stack to communicate with remote adversaries. ICMP-based stealth backdoors access ICMP traffic, raw-socket based stealth backdoors access all the packets with the same protocol number. Example, a TCP raw socket receives all TCP packets. The stealth backdoors intercepting the network stack behave like network sniffer which eavesdrop on the network traffic to obtain valuable information.

Virvilis and Gritzalis [34] focused on APT attacks through malwares such as Stuxnet, Duqu, Flame, and Red October. They discussed the issues that enabled the malware authors to evade detection from a wide range of security solutions and proposed counter measures for strengthening our defenses against similar threats. The paper goes over the evading techniques that APT attackers would use, such as, rootkit functionality, endpoint scanning with changed payload, encryption and obfuscation of network traffic, steganography, execution of malware in memory and fake digital certificates. The authors recommend patch management, strong network access controls and monitoring, strict Internet policies, protocol aware security solutions, monitoring DNS queries, monitoring for access to unusual domains, monitoring network connections, honeypots and honeynets, along with the standard host-based intrusions prevention systems as countermeasures for APT.

Korkin and Nesterow [35] presented a novel approach that detects zero-day malware in the memory dump under deliberate countermeasures. This proposed method uses modern graphics card or CUDA-enabled GPU hardware to detect malware in memory. Through this paper, the authors discussed the highest stealth malware out there, ways to detect this malware that is in the form of hidden drivers, and finally they propose an architecture of the software tool that uses CUDA-enabled GPU hardware to speed-up memory forensics.

Xu *et al.* [36] proposed a hardware-assisted malware detection solution that uses machine learning to monitor and classify memory access patterns. Unlike one model that distinguishes all types of malicious activity in 25, this solution is based off one model for each application separating its malware infected executions from legitimate executions. Their work is based on the fact that an infected application run will modify the control-flow/data structures compared to a benign run. This will be reflected in its memory access pattern. They achieve this by having In-processor monitoring of the memory accesses that looks at the virtual addresses for a more consistent signature. They used epoch markers - system calls, function calls, and the complete program run to detect the malicious behavior of an infected program. Their solution covers both user-level and kernel-level threats and demonstrated very high detection accuracy against kernel level rootkits.

Vaas and Happa [37] have proposed a solution to identify disguised processes. Their solution involves training a machine learning algorithm to identify anomalous behavior of a machine's processes on a per-application basis. Their approach is structured into 3 phases: *Acquisition phase, learning phase and production phase*. Their approach identifies the anomalous behavior of a process of an application through its virtual memory consumption for two reasons. Firstly, they believe memory utilization is less volatile in comparison to network operations or CPU usage readings. Secondly, they use virtual rather than physical memory consumption as the latter does not account for the amount of memory swapped to the hard drive. In the first phase, acquisition phase, the memory fingerprint of a target machine using process and system utilities (psutil) is gathered. In the second phase, the learning phase, the machine learning algorithm computes a model for every application based on the fingerprint, a threshold and a threshold factor to detect anomalous behavior. If the distance of the current print wrt the model print is 0, then the verification can be terminated. If not, the distance is checked to see if it exceeds threshold modified by a multiplicative constant factor. If so, they increase a counter that is maintained per-application basis, and serves as a buffer that is regularly checked to alarm in case the counter exceeds a given threshold.

3) Packet Monitoring: The most crucial part of an APT attack is the communication with the Command and Control Center (C&C). Communication with C&C happens not just once, but often multiple times, usually the first time the system is compromised and repeatedly later for data transfer. Monitoring at the end system level for any network packets with new destination IP addresses, packets with huge payloads, and large numbers of packets sent to the same IP address would help in identifying any suspicious behavior from within an end system.

Marchetti *et al.* [38] proposed a framework that can detect, out of thousands of hosts, a few hosts that show suspicious activities. They do not claim to identify the hosts that are surely compromised. They defend that their solution will help analysts to focus on a limited number of hosts rather than thousands of hosts in removing APT from the system. This solution of theirs provides a ranked list of top-k suspicious hosts generated by observing the key phases of APT across several hosts over time and comparing those analysis results of each of the hosts with their past and with other hosts of the observed network. Their solution works even for encrypted communications as the payload is not inspected. In addition the authors claim that this solution is scalable as most analyses can be executed in parallel thus giving us an efficient solution. The proposed framework in this paper involves flow collection and storage, feature extraction, feature normalization, computation of suspiciousness scores and ranking. Traffic going from internal host to outside is monitored as the framework assumes that an APT attacker would do it from internal to external rather than from external to internal to evade detection by traditional intrusion detection systems. And then features are computed for each internal host every time internal T, and then these computed features are extracted for further analysis.

McCuskr *et al.* [39] focus on the notion of tracking various network objects such as hosts, hostgroups, and networks, and determining if they are threats. The overall system layered the network flow activities into five layers from network flow collection to threat analysis. Events and data are collected from a number of different network sensors such as network flow, NIDS, honeypots, and then features can be extracted and aggregated over multiple periods to creating a sample space.

They designed three layers, to focus on the use of discriminative supervised/semi-supervised models to identify behavior primitives.

Villeneuve and Bennett [40] claim that monitoring and analyzing network traffic help in detecting APT activities. They analyze different APT campaign such as Taidoor, IXESHE, Enfal, and Sykipot which have been used to establish targeted attacks. These malware(s) establish communication with a C&C server using known protocols such as HTTP and usually configured through three ports 80, 443, 8080. Attackers usually use these ports because they know that often only these ports are open at the firewall level. However, attackers may use these ports to pass unmatched traffic type such as that sending any non-HTTP traffic on port 80 or any non-HTTPS traffic on port 443. This can trigger alert for further investigation. Monitoring timing and size of network traffic is another aspect to consider for APT detection. This is due to the fact that malware(s) typically sent *beacon*, which is basically communication packets, to C&C servers at given intervals. Thus, monitoring consistent intervals using DNS requests or URLs will help. Although designed malware use HTTP for C&C communication, they usually send requests using Application Programming Interfaces (APIs). Therefore, analyzing HTTP headers can help to distinguish API calls from typical browsing activities.

Vance [41] proposed a solution that utlizes flow based analysis to detect targeted attacks by determining normal versus abnormal behavior. Unlike typical network based detection, in flow based analysis, network traffic is aggregated so the amount of the data to be analyzed is reduced. Traffic based volume of transferred data, timing or packet size is analyzed and the result is a high detection rate, low false positives.

Hu *et al.* [42] have discussed APT attacks coupled with insider threats as a 2-layer game characterizing the joint threats from APT attackers and insiders as a defense/attack game between the defender and the APT attacker(s) and an information trading game among the insiders. The authors of this paper claim to identify the best response strategies for each player, and prove the existence of Nash Equilibrium for both games.

4) Code Monitoring: Creating software completely free of bugs is a mirage. Every software developed, every code that you release can never be guaranteed to be error-free. While making the code itself error-free is quite difficult, making sure that it is error-free when running in different environments is not possible. These bugs are the means for attackers to penetrate systems. While some of them could be known prior to the code release, there is always a possibility of unknown bugs. The possible vulnerabilities in the source code can be identified by static analysis techniques such as Taint Analysis and Data Flow analysis. In addition, monitoring the code during its execution for its performance and to make sure it runs within its scope, neither utilizing unexpected resources nor using up memory regions that otherwise are not accessible, would lead to identifying a threat much earlier before it can spread to other systems.

5) Log Monitoring: Logs are an important part of not only forensic analysis, but also when used appropriately can help in detecting or even preventing attacks in their early stages. Correlation of these logs such as memory usage logs, CPU usage logs, application execution logs, and system logs would yield a copious amount of information that would make sense and help in defending systems or network against unknown attacks rather than just have the individual logs that often end up in a pile to be searched afterward for evidence of an attack.

One such paper that correlates data collected from different type of logs is [43]. Bohara *et al*. this paper have proposed an intrusion detection approach that combines the network and host logs to find any malicious activity. From these logs, they extract 4 features, identification, network traffic based, service based and authentication based features which are further refiend to reduce redendancies through the use of Pearson Correlation Coefficient, following which those that do not contribute to clustering are removed. The resulting data is clustered to identify the maclicious activity. Their prosposed solution takes the approach of unsupervised learning to detect anomalies without any profiling the normal behavior of the system.

Shalaginov *et al.* [44] analyzed DNS logs to identify the communication packets "beacon" activities between infected internal hosts and external malicious domain names. Basically, they believe that a downloaded malware, as a foothold, will require opening an external communications channel to the Command and Control (C&C) server. This behaviour will leave a record of itself in network flow and DNS logs. Authors proposed a methodology for DNS logs analysis and events correlation by considering low latency interval time where they assumed the infected hosts will communicate the C&C server several time per day. From identifying an infected host, they link other hosts that have communicated with same suspicious domains. They pre-process the DNS logs to filter unwanted data, and only obtain IPv4 addresses from DNS logs. Then, they start to represent the meta-data in a graph fashion where the graph's vertices represent host IP address and domain names, while each edge corresponds to one query from an internal host to an external machine. The proposed methodology was evaluated using real DNS logs collected by Los Alamos National Laboratory published in 2013.

One of the challenges in log monitoring is that there is so much data to look at and analyze to detect an attack. Yen *et al.* [45] proposed an approach to address this problem by extracting information and knowledge from the dirty logs. The proposed approach involves 3 layers, first layer filters and normalizes the log data using network configuration. Second layer processes this normalized data into different features. Third layer performs clustering over these extracted features to determine any suspicious activity. Beehive, the name of this proposed solution, uses logs from different sources such as Web proxy logs, DHCP server logs, VPN server's remote connection logs, authentication attempt logs and anti-virus scan logs. The solution then proceeds with extracting features based on destination, host, policies and traffic, following which the features are clustered through an adapted k-means clustering algorithm to identify hosts whose behavior significantly differ from normal.

Bhatt *et al.* [46] discussed the kill chain attack model, and proposed a solution that works for a layered architecture, with outer layer having the least valuable assets and inner layer having the most valuable assets. Given this architecture, the attacker is assumed to perform, at least once, all the different stages of the attack model in order to get past a layer. Each layer can be accessed through the processes and applications running within the immediate outer layer. The solution requires that the probability of finding common vulnerabilities among different layers is very low, so that the possibility of reuse of the knowledge about vulnerabilities of a layer to attack another layer is minimized. The framework suggested by the authors detects attacks only with appropriate sensors that detect different stages of an APT attack at each layer. These sensor would be triggered by rules created with respect to the patterns of malicious behavior. Alerts and logs collected by these sensor should be stored and correlated to identify stages and phases of attacks in progress.

Niu *et al.* [47] proposed an approach to detect APT malware and C&C communication activities through DNS logs analysis. They evaluate their approach using DNS logs of mobile devices. Their approach assigns scores to C&C domains and normal domains. Therefore, to distinguish between normal and abnormal (C&C) domains, they select normal domains according to the number of DNS requests initiated by internal devices and extract fifteen features which are categorized under four general categories: *DNS request and answer-based features*, *domain-based features*, *time-based features*, and *whois-based features*.

B. APT Detection Methods

We classify the techniques for detecting APT into the following groups: *Anomaly based detection, detection by Pattern Matching.*

1) Anomaly Detection: One of the key characteristics of an advanced persistent threat is to adapt to the defender's efforts to resist it. And to defend against such a threat, the defense methods employed need to learn about and adapt to the offenders' attempts. These methods should constitute collecting data from several sources, learning from the collected data, and make predictions on the collected data to estimate and respond to the next possible attack.

Table V shows the attack methods and corresponding and existing defense techniques or countermeasures for each APT stage. However, these techniques are static, and thus APT attackers will find ways to evade these defense methods. For instance, they would create new malwares or change the existing malwares to have new signatures for the purpose of evading detection by the organization's anti-malware tool. Furthermore, they could have these malwares behave as normally as possible without raising any alarm at their behavior on a system. Here arises the need to employ different categories of anomalies such as point, contextual, and collective anomalies, and different anomaly detection methods that could detect these close-to-normal behaviors through these anomalies.

Extensive research has been done in anomaly detection techniques and methodologies over the past decade. Some of the early works that give a good understanding of anomaly detection in general have been discussed in this section, followed by works that have used those anomaly detection techniques to detect different stages of APT attacks.

Approaches and Methods: Hodge and Austin [48] have surveyed different outlier (anomaly) detection approaches and methodologies. They classified outlier detection approaches into three types: first, unsupervised clustering, an approach that processes the data as static distribution, pinpoints the most remote points, and flags them as potential outliers. Second, supervised classification that requires pre-labeled data, tagged as normal or abnormal. The third type is semi-supervised recognition or detection that takes pre-classified data to models only normally or very rarely models abnormal data. As new data arrives, the model is tuned to improve the outlier detection rate by defining a boundary of normality. Unlike supervised classification, this does not require any training data for abnormality and yet learns to recognize abnormality. The authors further classified the anomaly detection methodologies into *statistical anomaly detection*, *neural networks based anomaly detection* and *machine learning based anomaly detection* and explained how each of these methodologies handles outliers and made recommendations as to when they are appropriate for previously defined approaches.

Bhuyan *et al.* [49] have provided an overview of the extensive research done in network intrusion detection systems, specifically in network anomaly based detection approaches. They have given a qualitative survey of the different methods, systems, tools, and analysis pertaining to network anomaly detection. In addition, they have covered a wide variety of attacks, focusing on their sources and characteristics while comparing and giving performance metrics for various detection approaches.

Chandola *et al.* [50] gave a broad overview of anomaly detection techniques and how they are applicable to different research and application domains, along with the challenges associated with each of those techniques. The authors discussed different aspects of anomaly detection such as the nature of the input data, type of anomaly, available data labels, and output of anomaly detection. They point out that the nature of the attributes of the input data determines the applicability of anomaly detection techniques. In addition to the anomaly detection techniques classified and discussed by Hodge and Austin [48], Chandola *et al.* discussed two other anomaly detection techniques—information theoretic anomaly detection techniques and spectral anomaly detection techniques, both of which can operate in unsupervised settings, with the former making no assumptions on the underlying statistical distribution for the data, while the latter can automatically perform dimensional reduction, making it suitable for handling high-dimensional data sets.

Later, Chandola *et al.* [51] have discussed anomaly detection in a different perspective. They discussed the problems associated with detecting anomalies in discrete sequences and various techniques that address these problems. They classified the sequence anomaly detection into *sequence-based*,

TABLE V
APTs Stages and Corresponding Attack and Defense Methods

Stages	Attack Methods	Defense Measures
Reconnaissance	Social Engineering	User awareness
Accomplishing a foothold	Spear Phishing, Watering-hole	Malware Inspection, Content filtering, Blacklisting
Lateral movement	Privileges Escalation, Malware, Vulnerabilities exploitation	Access Control Listing, Firewall, Password Control
Exfiltration	Command and control	Firewall, Proxy, Encryption Use Control, blacklisting
Cover up	Traces erasing (e.g., deleting logs)	Forensics, alerts triggered

contiguous sub-sequence based anomaly detection and *pattern frequency-based anomaly detection*. The sequence-based anomaly detection approach is the basis for machine learning based anomaly detection, where sequences of training and/or test data are used to identify anomalies. The contiguous subsequence anomaly detection approach can be closely related to the detection of an end system's behavior in case of a malware download. Lastly, they explained the pattern frequency-based anomaly detection approach as one in which the frequency of the sequences is higher than normal as in case of failed login attempts.

Garcia-Teodoro *et al.* [52] have classified and reviewed several anomaly-based network intrusion detection techniques while presenting the challenges to be addressed. Their discussion included statistical-based, knowledge-based and machine-learning-based anomaly-based network intrusion detection techniques. They go over the need for anomaly-based detection techniques, and why a common signature-based approach brings two major drawbacks: pre-defined rules often being insufficient to detect unique or tailored attacks, and a lack of rules that verify application specific operation sequences.

Mehmood *et al.* [53] summarized anomaly detection based system that can be implemented using machine learning techniques. They showed the most widely used techniques such as: (1) Support Vector Machine (SVM) which classifies normalized data via appropriate kernels to divide data into two categories resulting anticipation between different datasets; (2) Fuzzy Logic (FL) which uses true or false to detect anomaly behavior; (3) Genetic Algorithm (GA) which builds mutation and crossover genomes, from existing or new genes, using heuristic search; (4) K-means which classifies data into different clusters where each cluster presents average of data based on provided means; (5) Artificial Neural Network (ANN) which accept different inputs and transform them until required output is achieved; (6) Association Rule which looks for correlations between different source of data (datasets) and how they can be applicable to cloud computing.

Zhang *et al.* [54] designed an interactive system to bridge the gap of network management and anomaly detection. They designed a Web-based visualization tool for analyzing the network and system anomalies within system logs. Their tool allows different views such as network graph, treemap, area chart, and general view. It also provides search ability based on different options such as searching by source/destination IP addresses. The dataset contains common network traffic logs such as network flow data and intrusion detection/prevention system (IDS/IPS) log files, as well as network health and status data for every single workstation and server such as CPU, memory and disk usage. This tool basically observes trending in the system activities in the form of peaks that show a source or a destination receiving or generating high volume of traffic.

Application to APT Detection: Anomaly detection will greatly benefit a defense system particularly when detecting an APT attack that is spread over several years making damage reversal quite difficult. With new malware variants being released every day and existing technologies such as rule-based analysis requiring skilled analysts to be involved in analyzing the behavior of the malware and design rule-based solutions to predict similar behaviors in the future, a gap between discovery and protection is bridging up, giving enough time for attackers to penetrate an organization's network. The earlier APT attacks are detected the better would be the state of an organization. By automating this analysis and detection part through the above anomaly detection approaches and methods that continuously monitor, learn, train, and update learning models, not only can the bridging gap disappear but even minor changes can be detected that are difficult for human analysts to observe. Learning techniques such as Perceptrons, Neural Networks, Centroids, Binary Decision Tree, Deep Learning, etc. can help process millions of data points every minute to establish normal behavior and compare data points to past behavior and identify anomalous differences in values.

However, to detect APT, a single anomaly and anomaly detection technique or method will not suffice. For instance, to detect abnormal memory usage by a process on a system, the detection system needs to know the usage history of the memory by the same process, requiring either a semi-supervised or a supervised approach to identify contextual anomalies. But to correlate and find similar or anomalous behaviors among several processes within the system and across the network of systems in the organization network, an unsupervised clustering approach towards identifying collective anomalies would be needed. However, one of the problems with anomaly detection is the amount of false positives and false negatives, specifically in case of semi-supervised and unsupervised learning methods. The reason for this is often the lack of clear distinction between normal and abnormal data. Further, the behavior of the users and the systems is not always the same, thus requiring a continuous learning and incremental model updating approach.

Nath and Mehtre *et al.* [55] evaluated four machine learning methods that have been used to statically analyze malwares and concluded that none of them suffice for defending against

TABLE VI
COMPARISON BETWEEN ANOMALY DETECTION BASED SOLUTIONS

Reference	**Learning Approach**	**Anomaly Detection Method**	**Source of Data**	**APT stages**
[57]	Semi-Supervised	Machine Learning	Network/Host logs	Establish Foothold and Lateral Movement
[58]	Supervised	Machine Learning	Network Traffic	Establish Foothold
[59]	Semi-Supervised	Statistical	Host-based logs	Accomplishing Foothold
[60]	Semi-Supervised	Machine Learning	Network Traffic	Accomplishing Foothold, Lateral Movement, Exfiltration
[61]	Supervised	Machine Learning	Malware Detection	Accomplishing Foothold
[62]	Supervised	Machine Learning	Network Traffic	Lateral Movement, Exfiltration
[63]	Supervised	Machine Learning	Network Traffic	Internal Exfiltration
[64]	Supervised	Machine Learning	Emails	Reconnaissance, Establishing Foothold
[40]	Supervised, Unsupervised	Machine Learning	Host/Network events	Not specific to a stage, established behavior profiling
[65]	Unsupervised	Machine Learning	Active Directory domain service logs	Lateral Movement, Exfiltration
[66]	Supervised	Statistical	Network traffic, Access Information	Maintaining Access, Lateral Movement, Data Exfiltration

attacks involving multi-stage or multiple files executed in parallel or sequential such as APT attacks.

In Table VI we provide high level comparison between different anomaly-based APT attack defense methods along with their learning methods (supervised, unsupervised and semi-supervised) and detection techniques (statistical, and machine learning based techniques which include rule-based & neural network approaches).

Kim *et al.* [56] have used rule-based anomaly detection to detect APT attacks. Their proposed approach involves 2 stages. In the first stage, behavior rule generation, they used machine learning and decision trees based on statistical data to generate behavior rules. In the second stage, abnormal behavior detection, they generate the feature description using MapReduce based on big data and compare it against the behavior rules obtained in the first stage to determine if the behavior of a host is abnormal.

Zhao *et al.* [57] proposed a system to detect APT malware infections. Their solution relies on the fact that APT malware uses Dynamic DNS to locate the C&C towards communicating its success in establishing foothold. Their solution involves 2 phases. Detection of malicious C&C domains followed by analysis of the associated IPs for any suspicious and malicious traffic. The authors used J48 decision tree algorithm as malicious DNS detector along with signature based detection and anomaly based detection components. The result from these 3 components is passed to their reputation engine towards obtaining high accuracy in identifying APT malware infections.

Friedberg *et al.* [58] reviewed several works on anomaly based detection techniques, and proposed a novel approach that learns the normal behavior of a system over time and report all actions that differ from the created system model. This, they claim, is in contrast to several other solutions that use a black-list kind of approach to detect an intrusion. Their proposed scheme uses log data produced by various systems and components in ICT networks from which their solution extracts a system model, that is used to detect and distinguish meaningful logs through event classes that contain implications between the events. These rules thus obtained describe the relations among different components in the network. This model is automatically and continuously generated to detect anomalies that are consequence of realistic APT attacks.

Cappers and van Wijk [59] proposed an approach to find the presence of APTs in the network by using machine learning techniques for contextually analyzing network traffic alerts by splitting them into messages and attributes. However, this approach has some limitations in terms of scalability when considering attributes with many different values, it also leaks the interaction with the number of attributes where many attributes will break the interaction whereas too few attributes will increase the risk of missing potential alerts correlations.

Yuan [60] presented a preliminary study on using deep learning-based technique for malware detection. The author believes that using conventional machine learning algorithms such as SVM, decision tree algorithm, K-NN cannot efficiently help due to the high false positive rates these algorithms generate. He states the reasons because, first, current malware and software are complex and diverse which means conventional ML models cannot capture enough features during learning phase; second, available datasets can be limited or outdated. The preliminary results in this paper show that the deep learning model overcomes conventional ML models such as random forest, isolation forest, AdaBoosting, and eXtreme Gradient Boosting, in term of accuracy. However, their model performs much slower than the conventional models.

Siddiqui *et al.* [61] also point out the high false positives obtained with the use of traditional machine learning algorithms and proposed a fractal based anomaly classification algorithm to reduce both false positives and false negatives. They use K-Nearest Neighbor (K-NN) machine learning algorithm and a data set that is a combination of two different data sources to cover both APTs traffic and non-malicious traffic. They collected APT data sets from Contagio malware database [66] and normal, non-malicious data from [67] and tested the combined dataset using a traditional supervised learning, i.e., K-NN and correlation based fractal dimension approach. They proved the better performance of their approach in terms of reduced false positives and false negative based on the fact that the correlation using fractal dimension has the capability to extract multiscale hidden information.

To detect security breaches that are designed to a specific target, Cappers and van Wijk [62] claim that deep packet inspection (DPI) and anomaly detection are indispensable. Authors proposed an approach for network traffic analysis where they consider visualization and machine learning techniques allowing system administrators to inspect and compare specific parts of the network traffic while preserving context. The proposed approach supports iterative refinement of classifier parameters based on new findings inside alerts messages (payload inspection). It uses pixel visualization to display the full structure of a network message as a horizontal line of pixels and to reduce false positives. Unfortunately, this approach focuses on monitoring traffic, thus, it can only inspect small fractions of traffic at the same time, which makes it hard to detect threats and malicious traffic over larger periods in time.

Dewan *et al.* [63] proposed an approach to distinguish between spear phishing and non spear phishing emails. They extracted features from spear phishing emails that have been sent to employees of 14 international organizations, by using social features extracted from LinkedIn. The authors performed their study on a dataset collected by Symantecs enterprise email scanning service. The authors defined nine features that extracted from LinkedIn profile of the phishing email's recipient as well as other features extracted from the emails. However, they found that the classifiers performed slightly worse with the feature set that includes social features. This is due to the limited amount from information that can be gathered from LinkedIn.

Hsieh *et al.* [64] proposed a framework for detecting APTs through monitoring active directory log data. The proposed framework focuses on taking active directory logs as time-series input and mining the sequential contexts from the collected logs. Then building probability Markov model to detect different behaviors occurring (anomaly detection). In general, the proposed framework looks for the changes in user's behavior over time through analyzing his/her accounts' log data. However, their Markov-model gives the best performance of about 66% recall rate or accuracy. This can tell that anomaly detection based on analyzing active directory log may be limited by information which active directory log can tell. Authors suggest that active directory log can be combined with other various logs or context to enhance the accuracy of anomaly detection.

Marchetti *et al.* [65] criticized that traditional defensive solutions such as signature-based detection systems and antiviruses can only detect standard malware and are ineffective against APTs. To solve APTs related threats, they propose a new framework called AUSPEX to support human analysts in detecting and prioritizing weak signals related to APT activities. The proposed framework combines different techniques based on big data analytics and security intelligence. It gathers and combines information from different sources: internal information from network probes located in an organization, and external information from the Web, social networks, and blacklists. Using network flow logs and access information, they focus on 3 major stages of APTs - foothold, lateral movement, and data exfiltration, and prioritize all internal clients that show suspicious activities.

Email spam has been known as a major method attackers use to launch APT. However, machine learning can help to learn valuable features from previous spams. Emails contain text fingerprints, URLs, phone numbers, images, attachments, etc, and these can be used to train a classifier to identify similar spams.

In APT, malware can hide in multiple-layered proxy network. For example, attackers can keep changing malicious URLs every couple of minutes, and thus blacklisting or whitelisting does not help in preventing users from visiting malicious URLS. However, machine learning can help in extracting the features of those URLs and classifying future URLs into normal or malicious.

Table VII summarizes the role of AI/ML techniques to defeat APTs at different APT stages. It shows also the challenges that are faced to each applicable AI/ML techniques. The following examples can be matched to different APT stages and how can be detected using AI/ML techniques.

A) Spear phishing: Using supervised machine learning can help to learn valuable features from previous spams. Since emails usually contains text fingerprints, URLs, phone numbers, images, attachments, etc, then it is possible to train a classifier on those contents and their features to predict similar spear phishing emails.

B) Malicious DNS domains: Such as continuously changing the IP address of the URL. This information can be detected through checking the DNS log file and find if this URL has been linked to previous IP addresses or not. Consequently, further information can be gathered to detect number of domains share the same IP address or addresses. In APT, malware can hide in multiple-layered proxy network. For example, attackers can keep changing malicious URLs every couple of minutes, so using of blacklisting and whitelisting will not suffice to prevent users from visiting malicious URLs. Neural Networks has the ability to solve this issue by back-propagation and continuous learning. Moreover, using unsupervised machine learning to learn the features of URLs and then classify new URLs to either good or bad classes. Here, we can find out that deploying both supervised and unsupervised ML techniques can increase the chance to detect APTs within the second stage *accomplishing a foothold* much better than applying only blacklisting methods.

In addition, clustering URLs or domains to identify DGAs (domain generation algorithms), which have been used by malware creators to generate domains that act as rendezvous points with the command and control servers. This can contribute to detect command and control communications.

C) User Profiling: Such as profiling set of machines that each user logs into to find anomalous access patterns. Here we can use clustering techniques to profile different users and their expansions. From here, the system administrator can identify if there were an privileges elevation or not. Therefore, deploying unsupervised ML techniques (clustering) can result

TABLE VII
APTs STAGES AND CORRESPONDING AI/ML ROLES

Stage	**AI/ML Role**	**AI/ML Techniques**	**Challenges**
Reconnaissance	Clustering	Unsupervised	High Volumes of Data to Process
Establishing foothold	Pattern Matching (classification)	Supervised	High False Positive Rate
Lateral movement	Grouping similar activities (Clustering), pattern matching (classification)	Unsupervised & Supervised	Dealing with numerous event data
Exfiltration	Pattern Matching (classification)	Supervised ML	High False Positive Rate
Cover up	Pattern recognition (neural network)	Supervised & Unsupervised ML	Huge volumes of low-quality evidence

in detecting one of the common APT stages which is *the attack expansion*.

D) Moving Data Monitoring: Such as applying deep data analysis on moved data content such as the size of moved file, for example, if a user moves more than 1GB of traffic within a limited time, but he/she usually dose not move more than 100MB per the limited time, an alert should be triggered. Using supervised ML techniques, we can detect this type of abnormal activities. Thus, the role of supervised ML techniques here is to stop one of the major stages of APT which is *the data exfiltration*.

E) Anomalous behavior: Detecting if, for example, CFO's computer makes unexpected financial transaction based on transaction's time, destination, etc. This can be achieved by deploying supervised ML model to learn the normal behavior within an organization. Thus, if a transaction is not matching a known pattern, an alert should be triggered.

2) Pattern Matching: Pattern matching is an old technique that regular intrusion detection and prevention systems employ. However, this technique has its own advantages. By observing for patterns on the behavior of a process and or application, malicious behavior can be detected. Yan and Zhang [68] proposed an approach to detect APT using structured intrusion detection. Their approach is based on high-level structured information captured in time series of network traffic. The Helix model [69] which was originally introduced as a Natural Language Processing (NLP) for behavior recognition in mobile sensing problems was utilized in their approach.

Giura and Wang [6] proposed a model of APT detection problem as well as methodology to implement it on a generic organization network. Their solution considers three types of events, candidate events, all events that are recorded by an organization logging mechanisms in any form; suspicious events, events reported by security mechanisms as suspicious, or events associated with abnormal or unexpected activity and attack events, events that traditional security systems aim to detect with regard to a specific attack activity. Events are correlated using context and correlation rules which are then filtered through detection rules to obtain a set of possible threats. They then used risk level and confidence indicators to evaluate the threats to attack goal. An APT incident is detected when the confidence indicator and the risk level of observed events go beyond specific thresholds, which are parameters specific to an organization environment.

C. Mitigation Methods

The mitigation techniques can be broadly classified into *Reactive Methods* and *Proactive Methods*.

1) Reactive Methods: Reactive methods identify possible attack scenarios based on vulnerabilities currently present in the system and perform an analysis of possible paths the attacker is likely to take to perform a multi-hop attack (one of the characteristics of APT).

A) Graph Analysis: Graph analysis is one of those fields that is noted for its ability to support analysis of complex networks and identifying sophisticated attacks. Johnson and Hogan [70] have proposed a novel approach to measure the vulnerability of a cyber network through graph analysis. Their solution specifically detects those attacks that involve lateral movement and privilege escalations using pass-the-hash (PTH) techniques to achieve the attack goal. The attack is detected by the use of a simple metric that measures with a graph how likely a node is to be reached from another arbitrary node, potentially making the network vulnerable. This metric is dynamically calculated from the authentication layers during the network security authorization phase and will enable predictable deterrence against attacks such as PTH.

Attack Graph has been used as modeling tool for study of multi-hop attacks in a network. An attack graph can be represented a $G = \{N, E\}$.

- The nodes can be expressed as $N = \{N_f \cup N_c \cup N_d \cup N_r\}$. N_f represent the fact node, e.g., access control list information $hacl(VM_1, 80, VM_2, 5000)$. This means VM_1 and VM_2 can communicate via ports 80 and 5000. N_c represents the exploit node, e.g., $execCode(VM_1, apache, user)$, which means on apache Web server an attacker can execute code with user privilege. N_d denotes the privilege level, e.g., $(root, VM_1)$ and N_r depicts the goal node, e.g., (root, DatabaseServer), i.e., gaining root privilege on the database server.
- Edges can be represented by union of edges with precondition and post-conditions of the exploit $E = \{E_{pre} \cup E_{post}\}$. Here $E_{pre} \subseteq (N_f \cup N_c) \times (N_d \cup N_r)$, which means N_c and N_f must be met in order to achieve N_d. $E_{post} \subseteq (N_d \cup N_r) \times (N_f \cup N_c)$. This means that condition N_d is achieved on satisfaction of N_f and N_c.

An attack graph can be used to study the attack path taken in APT scenarios, as an ordered sequence of events that leads to compromise of the system. Another advantage of using an attack graph is the ease of estimation of attack cost and return of investment (ROI) for a particular countermeasure on the chosen path.

The attack graph-based security analysis can help in identifying the most critical regions of the system and severity of particular attack that can contribute to the APT scenarios.

TABLE VIII
ATTACK GRAPH AND ATTACK TREE BASED METHODS

Category	Details	Complexity
Automated Attack Analysis [72]	Multi-prerequisite graph based on vulnerability and reachability information	O(E+NlgN); N is attack graph nodes and E is graph edges
Attack Cost Modeling [73]	Time Efficient Cost Effective hardening of network using Attack Graph	$O(n^{\frac{d}{2}})$; d represents the depth of the attack graph
Attack Cost Modeling [74]	Model checking based attack graph generation using Markov Decision Process (MDP)	Approximation algorithm $\rho(n) = H(max_{a \in A} \{\mu_G(a)\})$, where A is Attacks, μ is maximization function.
Scalable Attack Graph [75]	Scalable attack graph using logical dependencies.	$O(N^2) - O(N^3)$, where N is number of nodes in attack graph.
Attack Graph based Risk Analysis [76]	Scalable attack graph for risk assessment using divide and conquer approach	$O(r(n+c)^k)$, where r is small coefficient.
Attack Cost Modeling [77]	Attack Graph cost reduction and security problem solving framework Min. Cost SAT Solving.	NP-Hard problem, SAT solving methods employed.
Ranking Attack Graphs [78]	Asset Ranking algorithm for ranking attack graphs to identify attacks. Page Rank based algorithm	Similar to complexity of page rank algorithm.
Attack Cost Modeling [79]	Identifying critical portions of attack graph. Min. Cost SAT solving, Counter-example guided abstraction refinement (CEGAR)	NP-Hard problem, SAT solving methods used.

Based on the type of attack, attack goals, and input data, the attack graph methods discussed in Table VIII can be applied to the security assessment.

2) Proactive Methods: Proactive mitigation methods are based on techniques which can deceive attacker or change the attack surface to increase difficulty of attack for the attacker. We classify proactive methods into a) Honeypot & Honeynet; and b) Moving Target Defense.

A) Honeypot and Honeynet Strategies: One of the characteristics of APT attacks is the level of sophistication employed to perform the attacks. The evolving malware and attack forms are quiet difficult for defenders to keep up with. And often, a proactive approach such as a deception technology can help them battle against the unknowns and unexpected. In this defense methodology, defenders deceive the attackers by creating baits in the form of decoy documents or creating systems and or networks that are similar to the production environments but are not really part of the organization's production environment. Monitoring access to such honeypots and honeynets can help organizations detect the presence of APT attackers moving across the network of systems in search of organization's data after a foothold establishment.

Bowen *et al.* [79] addressed the insider threat problem with defense by deception approach. The paper discusses how internal misuse has been one of the most damaging malicious activities within an organization. The authors' proposed method attempts to trap the attackers who intend to exfiltrate data and use sensitive information. The solution is intended to confuse and confound the attackers with decoy data that makes it difficult for them to differentiate between original and decoy data and thus requiring more effort from the attackers in order to get into a system. These decoy documents are automatically created and are placed on decoy systems so as to entice the attackers with bogus credentials those when used would trigger an alert and thus giving away a malicious insider. Their proposition involves embedding a watermark in binary format into the decoy documents that could be detected when it is loaded into memory or egressed over a network. Additionally a beacon embedded in the decoy documents that signals a remote website upon opened for reading. If these two fail to detect a malicious insider, the contents of the decoy document is monitored as well. Bogus logins at multiple organizations as well as bogus and realistic bank information is monitored by external means. The authors classify the attacker's sophistication level to low, medium, high and highly privileged and then address the number of ways an attacker at the above mentioned levels can be deceived with exception to the highly privileged attackers that they specify is beyond the scope of this paper. They then explain the ways a decoy document can be designed, for instance, with embedded honeytokens, computer login accounts, network-level egress monitor that detects when the decoy document is transmitted, host-based monitor that detects when a document is touched, embedded beacon alerts that alert a remote server.

Urias *et al.* [80] proposed a deception framework that leverages virtualization and software defined networking to create unpredictable and adaptable deception environment. In this paper they evaluated the current state of art of deception networks pointing to the lack of contemporary technology, that does not utilize SDN or cloud technology to deploy high-fidelity environments, lack of centralized management, and lack of operational realism giving away the emulation to adversaries. They then discuss their proposed framework supporting its ability to give better insights into an adversary's actions by correlating the network and endpoint behavior data and allow them to dynamically modify the environment as needed.

Anagnostakis *et al.* [81] proposed a novel hybrid architecture that is a combination of the best of honeypots and anomaly detection. Their system has several monitors that monitor the traffic to a protected network or service, and the traffic that is considered anomalous is processed by a shadow honeypot to determine the accuracy of the anomaly prediction. This shadow honeypot is an instance of the protected application that has the same state as the normal instance of the application, but is instrumental to detect potential attacks. Attacks against the shadow honeypots are caught and any incurred state changes are discarded. Legitimate traffic that was misclassified by the anomaly detector will be validated by the shadow honeypot and will be transparently handled correctly by the system. They claim that their system has many advantages over using just an anomaly detector or honeypot as: 1) Lowers the false positives as shadow honeypot needs to confirm the anomaly; 2) Since the protected application is a mirror image of the actual application, system can defend attacks tailored against a specific site; 3) Protects application against client-side attacks; and 4) Easy integration of additional detection mechanisms. HoneyStat [13] runs sacrificial services inside a virtual machine, and monitors memory, disk and network events to detect abnormal behavior. With relatively few positives it could detect zero-day worms.

B) Moving Target Defense: Crouse *et al.* [82] discussed the importance of reconnaissance defenses involving deception and movement. They point out that Moving Target Defenses operate by constantly changing the attack surface, and thus attackers can no longer make static and long-term assumptions about the state of the network, affecting the reconnaissance phase of an attack. An example of MTD is network shuffling, which remaps the addresses in an attempt to render scanning useless. They further discuss the deception defenses involving honeypots that could be utilized to effectively mislead attackers performing reconnaissance on potential targets in a network. The authors thus proposed probabilistic models that give the benefits and costs associated with reconnaissance defenses, helping us understand under what circumstances they are most effective. They evaluated their models using 2 attacker scenarios, foothold establishment and minimum to win, finally concluding that a relatively small number of honeypots can offer a significant cyber defense in many situations that was better than defense by movement in the evaluated scenarios, although having both would yield the best reconnaissance defense performance.

MTD based deception methods can be classified into three categories based on the security modeling, i.e., *Shuffle*, *Diversity* and *Redundancy* [83].

- **Shuffle** allows system and network resources to be rearranged at various layers in protocol stack, e.g., VM migration, topology rearrangement, port hopping, etc.
- **Diversity** technique provides functionally equivalent variant of given software or Operating System.
- **Redundancy** technique involves provisioning of replicas of soft-wares or network resources such as decoy VMs, proxies, network paths. The goal of MTD is to limit the capacity of attacker by increasing the attack surface.

Another way of classifying MTD techniques is based on implementation in protocol stack as described below:

- **Network Level** MTD involves change in the network topology, e.g., IP hopping, traffic obfuscation.
- **Host Level** MTD requires change in host resources, OS, renaming of configurations, etc.
- **Application Level** MTD involves change in the application required, source code, memory mapping, software version.

The classifications described above have some overlap, e.g., application level MTD such as software version change is similar to diversity based MTD. Other MTD techniques include strategic placement of available security tools such as Intrusion Detection System (IDS) [95]. The APT scenarios rely on exploration of cloud system or network in order to create exploitation plan. The rearrangement of network or software components renders the exploratory knowledge of the attacker useless. We classify various MTD methods that can be used to prevent APT attacks in Table IX.

The MTD techniques discussed in Table IX can be effective defense against APT scenarios at various layers of protocol stack. Based on the requirement, MTD can be deployed at core or endpoint of the network. Some limitations of MTD, however, include impact on factors such as latency, bandwidth and service availability. Scalability of MTD solutions [96], and changes in network policies on reconfiguration of network resources are other aspects that need to be analyzed carefully before selecting appropriate MTD solution such as cost-impact of virtual machine and service migration as discussed in MASON framework [97].

V. Evaluation Methodologies of APTs Solutions

One of the most critical aspects to ensure the effectiveness of APT attacks detection solutions is evaluation. In this section, we present the most popular evaluation techniques and their strength and weaknesses. Since APT attacks can be quite complex and deeply buried in the usual network traffic, it is quite challenging to comprehensively test and evaluate such systems in both phases "(i) during development to enhance their effectiveness towards new attack methods/vectors through continuous algorithmic improvements, and then (ii) before deployment in order to tune configurations and adapt them to particular environments, e.g., to meet performance criteria." The evaluation of APT attack detection methodologies lacks data sets from realistic attack scenarios, and an easy performance evaluation and comparison is much harder than in other computer science domains—e.g., image categorization or semantic text analysis [100].

It has been noticed that most current APT detection solutions evaluate their proposed methodologies using machine learning models which usually involve three major components: *data collection, feature extraction, and testing*. The data collection can be either from a real network scenario or virtually manufactured (synthetic model). The real network scenario has advantages such as realistic test basis; however, it has disadvantages such as poor scalability in terms of user input, varying scenarios, privacy issues, and an attack on one's own system needed. Using a synthetic model for creating data allows full control of the amount of data gathered and how the network is set up. Synthetic models create a model with the desired properties, no regular noise, and no unknown properties. The lack of noise can be considered an advantage when

TABLE IX
MOVING TARGET DEFENSE TECHNIQUES AGAINST APT'S

Category	Details	MTD Strategy
Diversity [85]	SDN-based solutions for Moving Target Defense. Network and Host MTD	OS hiding, Network reconnaissance protection.
Shuffle [86]	Target movement based on attack probability	VM migration.
Redundancy [87]	Openflow based Random host mutation	Physical IP mapping to corresponding virtual IP. Network MTD.
Shuffle [88]	Fingerprint hopping method to prevent fingerprint attacks. Host MTD	Game theoretic model for fingerprint hopping.
Diversity [89]	Dynamic game based MTD for DDoS Attacks. Network MTD.	Dynamic game for flooding attacks.
Shuffle, Diversity, Redundancy [90]	Security Models for MTD. Network MTD	Effectiveness analysis for MTD countermeasures.
Diversity [91]	Dynamic MTD using multiple OS rotation. Host MTD.	Network Threat based OS rotation.
Shuffle [92]	Optimal MTD strategy based on Markov Game. Application MTD.	Dyanmic game, MTD Hopping.
Diversity [93]	Software Diversity and Entropy based MTD. Application MTD.	Cost, usability analysis of software diversity.
Shuffle [94]	Software Defined Stochastic Model for MTD.	High Availability and MTD Cost Modeling.
Redundancy [95]	Decoy based cyber defense using Randomization. Network MTD	IP address randomization.

TABLE X
MINING TECHNIQUES AGAINST COMMON FEATURES AND TARGETED APT STAGES

Mining Techniques	Common Features	Targeted APT Stage
Emails	terms, structure statistics, values of emal header fields like "From", "To", and "CC", email has an attachment or not, the domain name of an email address, the email sender's information, such as the writing style and the user name of the email sender [99]	Reconnaissance
Malware	strings, byte sequences, opcodes APIs/System calls, memory accesses, file system accesses, Windows registry, CPU registers, function length, PE file characteristics, and raised exceptions, network, AV/Sandbox submissions, and code stylometry [100]	Foothold (watering hole, spear phishing)
DNS logs	IP addresses, distinct domain names, number of queries at each domain name by time, authoritative answer, type of DNS packet requested, resource record time to live (i.e., high TTL values are likely indicators of malicious domains)	C&C communication
System logs	failure login attempts, source/destination IP addresses, service type, protocol type, CPU utilization, file system usage, health status, network flows, internal and external flows, running process	Lateral movement, C&C communication
Outgoing Network Traffic	source/destination port addresses, type of physical media, source/destination IP addresses, service type, protocol type, flow direction, bytes sent, average packet size, average received size, traffic flow ratio, interval of packets sent	Data exfiltration, C&C communication

the goal is to create a model that allows simple reproducibility. However, the drawback of using a synthetic-based model is that APT attacks are based on simplified attack scenarios and are deployed in controlled environments where no realistic noise is involved in the collected data, which is one of the major points APT attackers consider to stay undetected and move low and slow. Other APT studies use semi-synthetic data that is more realistic than synthetic data, and easier to produce than real data. However, it is simplified and biased if an insufficient synthetic user model is applied [100].

The second important component is the feature selection, which is a major aspect that affects the results when using machine learning to solve a problem. Usually, the collected data are raw data and cannot be directly used for evaluating machine learning models. Therefore, it is necessary to preprocess the raw data and then select needed features. It is not necessary that selected features in an APT detection solution be used in another solution. Usually, the problem formalization has an influence on this task and determines which features can be selected. For instance, when mining and investigating log data, the available features are not similar to those that can be selected from network traffic or malware behavior. A feature is information associated with a characteristic and/or behavior of the object, where the feature may be static (e.g., derived from metadata associated with the object) and/or dynamic (e.g., based on actions performed by the object after virtual processing of the object such as detonation) [101]. Table X presents a list of common features against mining techniques.

VI. CHALLENGES

The nature of the APT attacks is itself a challenge in defending against and with other parameters such as the source of that attack, whether inside or outside, the infrastructure of the defending environment, make defending against these attacks more challenging. In this section, we discuss the several challenges in defending against APT attacks.

A. *Determined and Powerful Attackers*

The biggest challenge in defending against APT attacks is the deterministic nature and the strength of the attackers. A strong defense system might be in place, but for persistent attackers it all comes down to time and building more advanced and complex tools that could bypass this defense system. And these resources are plentiful for these attackers, enabling them to develop new malwares and custom tools to help them achieve their goal.

B. Long Duration of Attacks

APT attacks are often performed over a long duration of time, and while detecting the individual events is one challenge, correlating the events over several months is another. The state of a machine that showed suspicious behavior needs to be tracked for any further incidents that could be correlated to the suspicious behavior shown earlier by the machine. And for a large network, this is quite a challenge not only due to the number of systems connected, but also due to the false positives and incorrect leads on possible APT attack in progress that the alerts triggered by those systems can cause.

C. Internal Employees

As mentioned above, APT attacks involve gathering useful information about targeted organization such as collecting employees' names, emails, addresses, etc. This is usually done using social engineering techniques which rely on the naiveté and/or gullibility of an organization's employees. People are known to be the weakest point in the APT kill chain. They can help the attackers to achieve their goals in two ways: 1) internal users intentionally disclose secret information to outside entity; or 2) by mistake, internal users provide useful information to APT attackers.

To stop or at least reduce the effectiveness of the first point, it is important to establish clear security policies that outline with whom employees may share information and how that information should be transmitted. Official channels for security and IT personnel to contact staff must be created, and vice versa. To stop or at least reduce the effectiveness of the second point, it is important to limit information access by, for example, shredding company records or any documentation that includes names or other employee information. Staff should be educated to not provide any information to outside people unless that is under known and approved procedures and how they should handle phone calls, emails, and other inquiries. In addition, provide staff with regular security awareness training to outline what strategies and tactics APT attackers can use. Therefore, educating staff is an important step toward increasing the awareness and reducing the chance of APT attacks.

D. Infrastructure-Based Challenges

One of the major challenges in detecting and preventing APT attacks is when the environment uses cloud computing resources. Not only are detrimental activities such as data exfiltration difficult to monitor, considering the ample number of ways the data can be broken and sent out, but also the large number of resources add to the difficulty of monitoring and correlating events across the entire network system.

VII. RESEARCH OPPORTUNITIES

Advanced Persistent Threats are not threats that go away when you have strong security in place; instead they just become more and more complex as the defense systems become stronger. Such is the persistent nature of these attacks. As new defense techniques are developed, attackers will be required to build advanced tools that will find them a way to get into the system and achieve their goals, but how far they go lies in the defense techniques. In this section, we identify several areas that are yet to be researched in towards defending against APT attacks.

TABLE XI
APT STAGE-BASED RESEARCH OPPORTUNITIES

Stage	**Open Research Opportunities**
Stage 1	Detecting and correlating reconnaissance activities
Stage 2	automatic detection of spear-phishing emails and their correlation to events in further stages, fileless malware, Detecting exploitation of known and zero-day vulnerabilities
Stage 3	Detecting movement of attackers that show no anomalous behavior, Attackers reverse engineering the behavior based detection systems, Security risk assessment
Stage 4	Detrimental activities with use of cloud computing resources, correlation of activities spread over a long time
Stage 5	Digital forensics

As mentioned earlier, a successful attack would require attackers to spend enough time in each of the attack stages. Though some solutions exist for each of these stages, there are several that have yet to be explored. In Table XI, we mapped some open research opportunities in APT attack stages.

Spear-phishing emails, often, have a huge role and impact in an APT attack. Many times, it is through these emails that the attackers gain a foothold in the system. An automatic detection and correlation of these emails, and removal of these emails before the target employee opens it could prevent APT attacks in earlier stages.

In addition, zero-day vulnerabilities and the exploits using these zero-day vulnerabilities are yet to researched on. That said, one other research opportunity in Stage 2 would be detecting presence of fileless malware often referred to as in-memory attacks. Techniques that rely on behavior analysis to detect these in-memory attacks are being developed. However, the problem with behavior analysis is, for one, it is associated with time frame, such as keeping track of, say 30 days of, process behavior and rising alert when the behavior is found to be not its own, and second, some processes are not easy to make profile of such as browsers whose memory usage can go up and down. Either way, these behavior analysis can be reverse engineered, and it will not be long before attackers can manipulate the behavior analysis. This same applies to the machine learning methods that are used for anomaly detection. With enough time spent by the attackers and throwing of alerts from different systems across the network, it is quite possible that the attackers could decipher the rules and working of the defense system and evade detection by it when they are ready to move ahead. A recent research by Carlini *et al.* [102] explains how secrets can be extracted from any deep learning model, and discusses model stealing and inversion attacks that can be used to extract parameters and statistics about training data respectively.

TABLE XII
A COMPARISON OF OUR SURVEY WITH EXISTING SURVEYS IN LITERATURE

Survey	**Comprehensive Analysis of APT stages**	**APT Attacks Case Study**	**Mapping of APT stages to attack vectors**	**Different Measures to take**	**APT monitoring approaches**	**APT detection methods**	**Recommended Approaches**	**Challenges**	**Research Opportunities**
[7]	✓	✓					✓		
[4]	✓	✓	✓	✓					
[26]	✓							✓	
[107]	✓			✓					
[108]	✓	✓		✓		✓			
Our Survey	✓	✓	✓	✓	✓	✓	✓	✓	✓

Further, investigating hacker communities can help to identify the zero-day vulnerabilities before being exploited. According to [103] some vulnerabilities have been discussed by the black-hat community before being publicly exposed by ethical organizations. Hackers interact and communicate with each other through forums, which are user-oriented platforms that have the sole purpose of enabling communication among hackers worldwide. These so called dark-Web forums are usually very similar to other normal Web-forums; they feature discussions on programming, hacking, and cybersecurity [104], [105]. These forums provide an opportunity to hackers worldwide to exchange their discoveries, custom tools and malware, etc. The existence of such hacker communities is common across various geopolitical regions, including the U.S., Russia, the Middle East, China, and other regions. This presents a growing problem of global significance. Research in this area has potential for a high social-impact [103].

Another area that has impact on the APT defense systems is cloud computing. Cloud computing offers different types of services and resources that can be used to send, store or process data. A defense system for an organization with no cloud resources, monitors for data exfiltration activities to an unknown or external IP. But in case of organizations having cloud resources, detecting the exfiltration activities can be quiet challenging and an area to be explored due to the multiple cloud resources, services that can be utilized in exfiltrating the data. The proposed defense system should have a strong correlation model that can correlate the interlinked activities involved in exfiltrating the organization's data. For instance, attackers can use the target organization's cloud storage service to exfiltrate the data rather than send it directly over the organization's network to their command and control center. The use of the storage service requires the attackers to steal credentials of a user account on the organization's cloud that has permissions to place or retrieve objects from this storage service. Once the credentials are stolen, they can upload the data to the storage resource, and using the same credentials can download the data onto their command and control center with out being detected.

Recent developments in attack methods are leaving little traces of forensic evidence. One such method is in-memory attacks. These are not file based, and thus give a tough time for forensic investigators on tracking their origination, and spread to other systems as all that is left is that an in-memory attack has been made and possibly the script that was run.

VIII. DISCUSSION

Since the report of the first APT attack, some works have studied APT in terms of malware, spear-phishing attacks, or in terms of exfiltrating data. From detecting a possible APT attack through collecting reconnaissance information from social profile activity, through establishing footholds via malware(s) and spear-phishing emails, to detecting extraction of huge volumes of data, several works have studied and proposed schemes for defending against only one of the stages of an APT attack. However, very few have studied and addressed defending APT attacks in their entirety. In this work, we have explored several research works focusing on individual stages of APT as well as APT in its entirety. We reviewed different techniques and methods used in defending against APT attacks and provided clarification on what threats are not APT with several real world APT attack scenarios. In order to ensure the novelty and new contribution of our survey, we thoroughly compare our work with existing surveys, as shown in Table XII.

Vukalović and Delija [106] have discussed APT attack stages and suggested educating users and system administrators about the attack vectors as a first step, followed by implementing stricter policies and static rules and to use software tools such as SNORT to detect anomalies. APT attack detection is beyond a single tool's capability or user awareness. Often, implementing stricter policies is not only difficult but is also adequate. All the attackers need to do is steal an account that has permissions from several entities that they can penetrate. Their work failed to realize the challenges in detecting an APT attack and the possibility of new attack vectors that evolve each day.

Ussath *et al.* [7] have analyzed several published reports of APT attacks and came out with the finding that spear-phishing is the most common approach chosen for initial compromise, and dumping credentials is the most common chosen method for lateral movement. In addition, their results reveal that the exploited vulnerabilities as part of the APT attacks studied were mostly known vulnerabilities, and exploiting zero-day vulnerabilities are rarely involved. Chen *et al.* [4] have studied APT attacks deeper than other contemporary works, from analysis of APT stages through case studies of APT attacks, countermeasures to be taken, and several detection methods to help detect APT attacks. Although their work gives an overall idea of APT attacks, it lacks study of defending against APT attack by collecting data from different sources.

Tankard [107] have studied APT attacks, explained the different stages, and discussed the detection techniques for defending against APT attacks. This was one of the earlier works of APT attacks and is a good foundation for what APT attacks are. However, this work like others discusses the attack vectors, specifically in terms of the monitoring methods that can help in collecting data and how machine learning and graph analysis can be utilized to detect APT over a huge network, overcoming several challenges involved in huge volume of data analysis.

IX. Conclusion

Advanced Persistent Threats are threats that involve determined and persistent well-funded attackers with goals to gain crucial data or impede critical components of their target organization or government. Unlike targeted attacks, these attacks involve use of sophisticated tools and/or techniques. In this survey, we presented to the reader a comprehensive introduction of what is the APT, what is NOT APT, and a background on how APTs are performed. We presented APT attack trees and how they can be used in a defense system. We then provided a taxonomy for classifying APT defense methods that involves monitoring, detection, and mitigation methods. In addition, we provided technical background on current APT detection and mitigation approaches and evaluation techniques to evaluate the effectiveness of APT attacks' defense approaches. We finally presented noticeable challenges in deploying APT attacks' defense methods before we concluded our survey with identifying several research opportunities that are worthy of investigation.

References

[1] D. McWhorter, *APT1: Exposing One of China's Cyber Espionage Units*, vol. 18, Mandiant, Alexandria, VA, USA, 2013.

[2] R. S. Ross, *Managing Information Security Risk: Organization, Mission, and Information System View*, document SP-800-39, Nat. Inst. Stand. Technol., Gaithersburg, MD, USA, 2011.

[3] R. Kissel, "Glossary of key information security terms," NIST Interagency/Internal Rep., Gaithersburg, MD, USA, Rep. 7298rev2, Jun. 2013.

[4] P. Chen, L. Desmet, and C. Huygens, "A study on advanced persistent threats," in *Proc. IFIP Int. Conf. Commun. Multimedia Security*, 2014, pp. 63–72.

[5] B. Schneier, "Attack trees," *Dr. Dobb's J.*, vol. 24, no. 12, pp. 21–29, 1999.

[6] P. Giura and W. Wang, "A context-based detection framework for advanced persistent threats," in *Proc. IEEE Int. Conf. Cyber Security (CyberSecurity)*, 2012, pp. 69–74.

[7] M. Ussath, D. Jaeger, F. Cheng, and C. Meinel, "Advanced persistent threats: Behind the scenes," in *Proc. Annu. Conf. Inf. Sci. Syst. (CISS)*, 2016, pp. 181–186.

[8] L. Daigle, "WHOIS protocol specification," Internet Eng. Task Force, Fremont, CA, USA, Rep. RFC 3912, 2004.

[9] A. K. Sood and R. J. Enbody, "Targeted cyber attacks: A superset of advanced persistent threats," *IEEE Security Privacy*, vol. 11, no. 1, pp. 54–61, Jan./Feb. 2013.

[10] P. Mell, K. Scarfone, and S. Romanosky, "Common vulnerability scoring system," *IEEE Security Privacy*, vol. 4, no. 6, pp. 85–89, Nov./Dec. 2006.

[11] M. Motoyama, D. McCoy, K. Levchenko, S. Savage, and G. M. Voelker, "An analysis of underground forums," in *Proc. ACM SIGCOMM Conf. Internet Meas.*, 2011, pp. 71–80.

[12] S. Smadi, N. Aslam, L. Zhang, R. Alasem, and M. Hossain, "Detection of phishing emails using data mining algorithms," in *Proc. IEEE 9th Int. Conf. Softw. Knowl. Inf. Manag. Appl. (SKIMA)*, 2015, pp. 1–8.

[13] M. Lee and D. Lewis, *Clustering Disparate Attacks: Mapping the Activities of the Advanced Persistent Threat*, Symantec, Mountain View, CA, USA. Accessed: Jun. 26, 2013.

[14] K. Baumgartner and M. Golovkin, *The Earliest Naikon APT Campaigns*, Kaspersky Lab, Moscow, Russia, 2015.

[15] (Feb. 2015). *Kaspersky Labs—Global Research & Analysis Team, Carbanak APT: The Great Bank Robbery*. [Online]. Available: https://media.kasperskycontenthub.com/wp-content/uploads/sites/43/2018/03/08064518/Carbanak_APT_eng.pdf

[16] *The Duqu 2.0*, Kaspersky Lab, Moscow, Russia, Jun. 2015.

[17] K. Baumgartner and M. Golovkin. *The Naikon APT*. Accessed: Jan. 5, 2018. [Online]. Available: https://securelist.com/analysis/publications/69953/the-naikon-apt/

[18] (Feb. 2015). *Kaspersky Labs—Global Research & Analysis Team, Equation Group: Questions and Answers*. [Online]. Available: https://wikileaks.org/ciav7p1/cms/files/Equation_group_questions_and_answers.pdf.

[19] *Operation Cleaver*, Cylance, Irvine, CA, USA, Dec. 2014. [Online]. Available: https://www.cylance.com/content/dam/cylance/pages/operation-cleaver/Cylance_Operation_Cleaver_Report.pdf

[20] (Jan. 2014). *RSA Incident Response, Shell Crew*. [Online]. Available: https://www.emc.com/collateral/white-papers/h12756-wp-shell-crew.pdf

[21] *The 'Icefog' APT: A Tale of Cloak and Three Daggers*, Glob. Res. Anal. Team, Kaspersky Lab, Moscow, Russia, Sep. 2013.

[22] *The Regin Platform-Nation-State Ownage of GSM Networks*, Kaspersky Lab, Moscow, Russia, Nov. 2014.

[23] *Anunak: APT Against Financial Institutions*, GROUP-IB and FOX-IT, Moscow, Russia, Dec. 2014. [Online]. Available: https://www.group-ib.com/resources/threat-research/Anunak_APT_against_financial_institutions.pdf

[24] D. Aplerovitch. (Jul. 2014). *Deep in Thought: Chinese Targeting of National Security Think Tanks*. [Online]. Available: http://blog.crowdstrike.com/deep-thought-chinesetargeting-national-security-think-tanks/

[25] F. Ullah *et al.*, "Data exfiltration: A review of external attack vectors and countermeasures," *J. Netw. Comput. Appl.*, vol. 101, pp. 18–54, Jan. 2018.

[26] X. Wang, K. Zheng, X. Niu, B. Wu, and C. Wu, "Detection of command and control in advanced persistent threat based on independent access," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2016, pp. 1–6.

[27] Z. Ferrer and M. C. Ferrer, "In-depth analysis of Hydraq, the face of cyberwar enemies unfolds," vol. 37, Melbourne, VIC, Australia, CA ISBU-ISI, White Paper, 2010.

[28] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security Privacy*, vol. 9, no. 3, pp. 49–51, May/Jun. 2011.

[29] N. Falliere, L. O. Murchu, and E. Chien, "W32.Stuxnet dossier," Mountain View, CA, USA, Symantec Corp., White Paper, 2011.

[30] A. L. Johnson, "Cybersecurity for financial institutions: The integral role of information sharing in cyber attack mitigation," *North Carolina Banking Inst. J.*, vol. 20, no. 1, p. 277, 2016.

[31] L.-X. Yang, P. Li, X. Yang, and Y. Y. Tang, "Security evaluation of the cyber networks under advanced persistent threats," *IEEE Access*, vol. 5, pp. 20111–20123, 2017.

[32] B. Bencsáth *et al.*, *DUQU 2.0: A Comparison to DUQU*, CrySyS Lab, Budapest, Hungary, Feb. 2015.

[33] H. Yin, D. Song, M. Egele, C. Kruegel, and E. Kirda, "Panorama: Capturing system-wide information flow for malware detection and analysis," in *Proc. 14th ACM Conf. Comput. Commun. Security*, 2007, pp. 116–127.

[34] N. Virvilis and D. Gritzalis, "The big four—What we did wrong in advanced persistent threat detection?" in *Proc. IEEE 8th Int. Conf. Availability Rel. Security (ARES)*, 2013, pp. 248–254.

[35] I. Korkin and I. Nesterow, "Acceleration of statistical detection of zeroday malware in the memory dump using CUDA-enabled GPU hardware," presented at the Proc. 11th Annu. Conf. Digit. Forensics Security Law (CDFSL), May 2016, pp. 47–82.

[36] Z. Xu, S. Ray, P. Subramanyan, and S. Malik, "Malware detection using machine learning based analysis of virtual memory access patterns," in *Proc. IEEE Design Autom. Test Europe Conf. Exhibit. (DATE)*, 2017, pp. 169–174.

[37] C. Vaas and J. Happa, "Detecting disguised processes using application-behavior profiling," in *Proc. IEEE Int. Symp. Technol. Homeland Security (HST)*, 2017, pp. 1–6.

[38] M. Marchetti, F. Pierazzi, M. Colajanni, and A. Guido, "Analysis of high volumes of network traffic for advanced persistent threat detection," *Comput. Netw.*, vol. 109, pp. 127–141, Nov. 2016.

[39] O. McCusker, S. Brunza, and D. Dasgupta, "Deriving behavior primitives from aggregate network features using support vector machines," in *Proc. IEEE 5th Int. Conf. Cyber Conflict (CyCon)*, 2013, pp. 1–18.

[40] N. Villeneuve and J. Bennett, *Detecting APT Activity With Network Traffic Analysis*, Trend Micro Incorp., Tokyo, Japan, 2012.

[41] A. Vance, "Flow based analysis of advanced persistent threats detecting targeted attacks in cloud computing," in *Proc. 1st Int. Sci. Pract. Conf. Prob. Info Commun. Sci. Technol.*, 2014, pp. 173–176.

[42] P. Hu, H. Li, H. Fu, D. Cansever, and P. Mohapatra, "Dynamic defense strategy against advanced persistent threat with insiders," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, 2015, pp. 747–755.

[43] A. Bohara, U. Thakore, and W. H. Sanders, "Intrusion detection in enterprise systems by combining and clustering diverse monitor data," in *Proc. ACM Symp. Bootcamp Sci. Security*, 2016, pp. 7–16.

[44] A. Shalaginov, K. Franke, and X. Huang, "Malware beaconing detection by mining large-scale DNS logs for targeted attack identification," in *Proc. 18th Int. Conf. Comput. Intell. Security Inf. Syst. (WASET)*, 2016.

[45] T.-F. Yen *et al.*, "Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks," in *Proc. 29th Annu. Comput. Security Appl. Conf.*, 2013, pp. 199–208.

[46] P. Bhatt, E. T. Yano, and P. Gustavsson, "Towards a framework to detect multi-stage advanced persistent threats attacks," in *Proc. IEEE 8th Int. Symp. Service Oriented Syst. Eng. (SOSE)*, 2014, pp. 390–395.

[47] W. Niu, X. Zhang, G. Yang, J. Zhu, and Z. Ren, "Identifying APT malware domain based on mobile DNS logging," *Math. Prob. Eng.*, vol. 2017, Apr. 2017, Art. no. 4916953.

[48] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artif. Intell. Rev.*, vol. 22, no. 2, pp. 85–126, 2004.

[49] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: Methods, systems and tools," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 1, pp. 303–336, 1st Quart., 2014.

[50] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surveys*, vol. 41, no. 3, p. 15, 2009.

[51] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection for discrete sequences: A survey," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 5, pp. 823–839, May 2012.

[52] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Comput. Security*, vol. 28, nos. 1–2, pp. 18–28, 2009.

[53] Y. Mehmood, U. Habiba, M. A. Shibli, and R. Masood, "Intrusion detection system in cloud computing: Challenges and opportunities," in *Proc. IEEE 2nd Nat. Conf. Inf. Assurance (NCIA)*, 2013, pp. 59–66.

[54] T. Zhang, Q. Liao, and L. Shi, "Bridging the gap of network management and anomaly detection through interactive visualization," in *Proc. IEEE Pac. Visual. Symp. (PacificVis)*, 2014, pp. 253–257.

[55] H. V. Nath and B. M. Mehtre, "Static malware analysis using machine learning methods," in *Proc. SNDS*, 2014, pp. 440–450.

[56] H. Kim, J. Kim, I. Kim, and T. Chung, "Behavior-based anomaly detection on big data," in *Proc. 13th Aust. Inf. Security Manag. Conf.*, Perth, WA, Australia, 2015, pp. 73–80.

[57] G. Zhao, K. Xu, L. Xu, and B. Wu, "Detecting APT malware infections based on malicious DNS and traffic analysis," *IEEE Access*, vol. 3, pp. 1132–1142, 2015.

[58] I. Friedberg, F. Skopik, G. Settanni, and R. Fiedler, "Combating advanced persistent threats: From network event correlation to incident detection," *Comput. Security*, vol. 48, pp. 35–57, Feb. 2015.

[59] B. C. Cappers and J. J. van Wijk, "Understanding the context of network traffic alerts," in *Proc. IEEE Symp. Visual. Cyber Security (VizSec)*, 2016, pp. 1–8.

[60] X. Yuan, "Ph.D. forum: Deep learning-based real-time malware detection with multi-stage analysis," in *Proc. IEEE Int. Conf. Smart Comput. (SMARTCOMP)*, 2017, pp. 1–2.

[61] S. Siddiqui, M. S. Khan, K. Ferens, and W. Kinsner, "Detecting advanced persistent threats using fractal dimension based machine learning classification," in *Proc. ACM Int. Workshop Security Privacy Anal.*, 2016, pp. 64–69.

[62] B. C. Cappers and J. J. van Wijk, "SNAPS: Semantic network traffic analysis through projection and selection," in *Proc. IEEE Symp. Visual. Cyber Security (VizSec)*, Chicago, IL, USA, 2015, pp. 1–8.

[63] P. Dewan, A. Kashyap, and P. Kumaraguru, "Analyzing social and stylometric features to identify spear phishing emails," in *Proc. APWG Symp. Electron. Crime Res. (eCrime)*, 2014, pp. 1–13.

[64] C.-H. Hsieh, C.-M. Lai, C.-H. Mao, T.-C. Kao, and K.-C. Lee, "AD2: Anomaly detection on active directory log data for insider threat monitoring," in *Proc. Int. Carnahan Conf. Security Technol. (ICCST)*, 2015, pp. 287–292.

[65] M. Marchetti, F. Pierazzi, A. Guido, and M. Colajanni, "Countering advanced persistent threats through security intelligence and big data analytics," in *Proc. 8th Int. Conf. Cyber Conflict (CyCon)*, 2016, pp. 243–261.

[66] M. Parkour. (2013). *Contagio Malware Database*. [Online]. Available: https://www.mediafire.com/folder/c2az029ch6cke/TRAFFIC_PATTERNS_COLLECTION#734479hwy1b97

[67] *Darpa Scalable Network Monitoring (SNM) Program Traffic (11/03/2009 to 11/12/2009)*, DARPA, Arlington, VA, USA, 2012.

[68] X. Yan and J. Zhang, "Early detection of cyber security threats using structured behavior modeling," *ACM Trans. Inf. Syst. Security*, vol. 5, pp. 1–19, Jan. 2013.

[69] H.-K. Peng, P. Wu, J. Zhu, and J. Y. Zhang, "Helix: Unsupervised grammar induction for structured activity recognition," in *Proc. IEEE 11th Int. Conf. Data Min. (ICDM)*, 2011, pp. 1194–1199.

[70] J. R. Johnson and E. A. Hogan, "A graph analytic metric for mitigating advanced persistent threat," in *Proc. IEEE Int. Conf. Intell. Security Informat. (ISI)*, 2013, pp. 129–133.

[71] K. Ingols, R. Lippmann, and K. Piwowarski, "Practical attack graph generation for network defense," in *Proc. 22nd Annu. Comput. Security Appl. Conf. (ACSAC)*, 2006, pp. 121–130.

[72] M. Albanese, S. Jajodia, and S. Noel, "Time-efficient and cost-effective network hardening using attack graphs," in *Proc. 42nd Annu. IEEE/IFIP Int. Conf. Depend. Syst. Netw. (DSN)*, Boston, MA, USA, 2012, pp. 1–12.

[73] S. Jha, O. Sheyner, and J. Wing, "Two formal analyses of attack graphs," in *Proc. 15th IEEE Comput. Security Found. Workshop*, 2002, pp. 49–63.

[74] X. Ou, W. F. Boyer, and M. A. McQueen, "A scalable approach to attack graph generation," in *Proc. 13th ACM Conf. Comput. Commun. Security*, 2006, pp. 336–345.

[75] J. Lee, H. Lee, and H. P. In, "Scalable attack graph for risk assessment," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, 2009, pp. 1–5.

[76] J. Homer, X. Ou, and M. A. McQueen, "From attack graphs to automated configuration management—An iterative approach," Kansas State Univ., Manhattan, KS, USA, Rep. 2008-1, 2008.

[77] R. E. Sawilla and X. Ou, "Identifying critical attack assets in dependency attack graphs," in *Proc. Eur. Symp. Res. Comput. Security*, 2008, pp. 18–34.

[78] H. Huang, S. Zhang, X. Ou, A. Prakash, and K. Sakallah, "Distilling critical attack graph surface iteratively through minimum-cost SAT solving," in *Proc. 27th Annu. Comput. Security Appl. Conf.*, 2011, pp. 31–40.

[79] B. M. Bowen, S. Hershkop, A. D. Keromytis, and S. J. Stolfo, *Baiting Inside Attackers Using Decoy Documents*. Heidelberg, Germany: Springer, 2009.

[80] V. E. Urias, W. M. Stout, and H. W. Lin, "Gathering threat intelligence through computer network deception," in *Proc. IEEE Symp. Technol. Homeland Security (HST)*, 2016, pp. 1–6.

[81] K. G. Anagnostakis *et al.*, "Detecting targeted attacks using shadow honeypots," in *Proc. USENIX Security Symp.*, 2005, p. 9.

[82] M. Crouse, B. Prosser, and E. W. Fulp, "Probabilistic performance analysis of moving target and deception reconnaissance defenses," in *Proc. 2nd ACM Workshop Moving Target Defense*, 2015, pp. 21–29.

[83] J. B. Hong and D. S. Kim, "Assessing the effectiveness of moving target defenses using security models," *IEEE Trans. Depend. Secure Comput.*, vol. 13, no. 2, pp. 163–177, Mar./Apr. 2016.

[84] P. Kampanakis, H. Perros, and T. Beyene, "SDN-based solutions for moving target defense network protection," in *Proc. IEEE 15th Int. Symp. World Wireless Mobile Multimedia Netw. (WoWMoM)*, 2014, pp. 1–6.

[85] S. Debroy, P. Calyam, M. Nguyen, A. Stage, and V. Georgiev, "Frequency-minimal moving target defense using software-defined networking," in *Proc. Int. Conf. Comput. Netw. Commun. (ICNC)*, 2016, pp. 1–6.

[86] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "OpenFlow random host mutation: Transparent moving target defense using software defined networking," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw.*, 2012, pp. 127–132.

[87] Z. Zhao, F. Liu, and D. Gong, "An SDN-based fingerprint hopping method to prevent fingerprinting attacks," *Security Commun. Netw.*, vol. 2017, p. 12, Feb. 2017.

[88] A. Chowdhary, S. Pisharody, A. Alshamrani, and D. Huang, "Dynamic game based security framework in SDN-enabled cloud networking environments," in *Proc. ACM Int. Workshop Security Softw. Defined Netw. Function Virtualization*, 2017, pp. 53–58.

[89] J. Hong, "The state of phishing attacks," *Commun. ACM*, vol. 55, no. 1, pp. 74–81, 2012.

[90] M. Thompson, N. Evans, and V. Kisekka, "Multiple os rotational environment an implemented moving target defense," in *Proc. 7th Int. Symp. Resilient Control Systems (ISRCS)*, 2014, pp. 1–6.

[91] C. Lei, D.-H. Ma, and H.-Q. Zhang, "Optimal strategy selection for moving target defense based on Markov game," *IEEE Access*, vol. 5, pp. 156–169, 2017.

[92] S. Neti, A. Somayaji, and M. E. Locasto, "Software diversity: Security, entropy and game theory," in *Proc. 7th USENIX HotSec*, 2012.

[93] I. El Mir *et al.*, "Software defined stochastic model for moving target defense," in *Proc. Int. Afro Eur. Conf. Ind. Adv.*, 2016, pp. 188–197.

[94] A. Clark, K. Sun, L. Bushnell, and R. Poovendran, "A game-theoretic approach to IP address randomization in decoy-based cyber defense," in *Proc. Int. Conf. Decis. Game Theory Security*, 2015, pp. 3–21.

[95] S. Sengupta, A. Chowdhary, D. Huang, and S. Kambhampati, "Moving target defense for the placement of intrusion detection systems in the cloud," in *Proc. Conf. Decis. Game Theory Security*, 2018, pp. 326–345.

[96] A. Chowdhary, S. Pisharody, and D. Huang, "SDN based scalable MTD solution in cloud network," in *Proc. ACM Workshop Moving Target Defense*, 2016, pp. 27–36.

[97] A. Chowdhary, A. Alshamrani, D. Huang, and H. Liang, "MTD analysis and evaluation framework in software defined network (MASON)," in *Proc. ACM Int. Workshop Security Softw. Defined Netw. Function Virtualization*, 2018, pp. 43–48.

[98] G. Tang, J. Pei, and W.-S. Luk, "Email mining: Tasks, common techniques, and tools," *Knowl. Inf. Syst.*, vol. 41, no. 1, pp. 1–31, 2014.

[99] D. Ucci, L. Aniello, and R. Baldoni, "Survey on the usage of machine learning techniques for malware analysis," *CoRR*, vol. abs/1710.08189, 2017.

[100] F. Skopik, G. Settanni, R. Fiedler, and I. Friedberg, "Semi-synthetic data set generation for security software evaluation," in *Proc. 12th Annu. Int. Conf. Privacy Security Trust (PST)*, Toronto, ON, Canada, 2014, pp. 156–163.

[101] T. Haq, J. Zhai, and V. K. Pidathala, "Advanced persistent threat (APT) detection center," U.S. Patent 9 628 507, Apr. 18, 2017.

[102] N. Carlini, C. Liu, J. Kos, Ú. Erlingsson, and D. X. Song, "The secret sharer: Measuring unintended neural network memorization & extracting secrets," *CoRR*, vol. abs/1802.08232, 2018.

[103] V. Benjamin, W. Li, T. Holt, and H. Chen, "Exploring threats and vulnerabilities in hacker Web: Forums, IRC and carding shops," in *Proc. IEEE Int. Conf. Intell. Security Informat. (ISI)*, Baltimore, MD, USA, 2015, pp. 85–90.

[104] E. Nunes *et al.*, "Darknet and deepnet mining for proactive cybersecurity threat intelligence," in *Proc. IEEE Conf. Intell. Security Informat. (ISI)*, Tucson, AZ, USA, 2016, pp. 7–12.

[105] M. Almukaynizi *et al.*, "Proactive identification of exploits in the wild through vulnerability mentions online," in *Proc. Int. Conf. Cyber Conflict (CyCon U.S.)*, Washington, DC, USA, 2017, pp. 82–88.

[106] J. Vukalović and D. Delija, "Advanced persistent threats-detection and defense," in *Proc. 38th Int. Conv. Inf. Commun. Technol. Electron. Microelectron. (MIPRO)*, 2015, pp. 1324–1330.

[107] C. Tankard, "Advanced persistent threats and how to monitor and deter them," *Netw. Security*, vol. 2011, no. 8, pp. 16–19, 2011.

Adel Alshamrani received the B.S. degree in computer science from Umm Al-Qura University, Saudi Arabia, in 2007, the M.S. degree in computer science from La Trobe University, Melbourne, Australia, in 2010, and the Ph.D. degree in computer science from Arizona State University, Tempe, AZ, USA, in 2018. He is an Assistant Professor with the College of Computer Science and Engineering, University of Jeddah, Jeddah, Saudi Arabia. He has eight years of combined work experience in information security, network engineering, and teaching while working in the Faculty of Computing and Information Technology, King Abdul Aziz University. His research interests include information security, intrusion detection, and software defined networking.

Sowmya Myneni received the M.S. degree in computer science from New Mexico State University in 2010. She is currently pursuing the Ph.D. degree in computer science with Arizona State University, Tempe, AZ, USA. Her research interests include network and information security specifically intrusion detection and prevention, cryptography, authentication and authorization, and wireless network security. Besides being a student, she is a certified Penetration Tester (GPEN) currently working as a Security Engineer.

Ankur Chowdhary received the B.Tech. degree in information technology from Guru Gobind Singh Indraprastha University in 2011 and the M.S. degree in computer science from Arizona State University, Tempe, AZ, USA, in 2015, where he is currently pursuing the Ph.D. degree. He was an Information Security Researcher with Blackberry Ltd., and an RSG and Application Developer with CSC Pvt. Ltd. His research interests include SDN, Web security, and network security and application of machine learning in the field of security.

Dijiang Huang received the B.S. degree from the Beijing University of Posts and Telecommunications, Beijing, China, in 1995 and the M.S. and Ph.D. degrees from the University of Missouri–Kansas City, Kansas City, MO, USA, in 2001 and 2004, respectively. He is an Associate Professor with the School of Computing Informatics and Decision System Engineering, Arizona State University, Tempe, AZ, USA. His research interests include computer networking, security, and privacy. He was a recipient of the ONR Young Investigator Program Award. He is an Associate Editor of the *Journal of Network and System Management* and an Editor of the IEEE COMMUNICATIONS SURVEYS AND TUTORIALS. He has served as an Organizer for many international conferences and workshops.