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 its associated entities, have long before expanded their target zone to include private and corporate sectors. These class of threats that every nation and organization wants to protect itself against are known as Advanced Persistent Threats. While nation sponsored attacks will always be marked for their sophistication, 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, they have, inadequate. As defenders strive hard to secure every endpoint and every link with in their networked system, attackers are finding new ways to penetrate into their target systems. With each day bringing new forms of malware with new signatures and behavior that's close to normal, a single traditional threat detection system would not suffice. These so called Advanced Persistent Threats are difficult to achieve as well as difficult to detect. While it requires time and patience to perform APT, solutions that adapt to the adapting behavior of APT attacker(s) are required. Several works have been published in 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 clean-up as one such solution demands complex correlation and behavior analysis of every event, user, system with in the network and across the network. Through this survey paper, we intend to bring before you 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 you with different case studies of APT attacks, different monitoring methods and deception methods to be employed for a fine grained control of security of a networked system. We conclude our paper with different types of challenges that one would face in defending against APT, and the opportunities for further research ending with a note on what we learned during our writing of this paper.**

***Index Terms*—Advanced Persistent Threat, Cyber-Security, Information security, Network security**

I. Introduction

THANKS to the strong emphasis on information security by security researchers across the world, security that once was proprietary to military and well-established organizations, has now started to become part of every organization. But this isn't just enough. Every day is introducing us to a new type of malware, a new design of attack form. There were days when an attacker or a group of attackers target 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 weren't trying to hide their actions. There are still these types of attacks, but there is a different breed of attacks that has become increasingly prominent over the last couple decades. This different class of attacks is what this paper is all about. This class of attacks is characterized by it's slow and low movement of a group of attackers to accomplish their goal, which is usually stealing the target's data with out getting caught. The term given to this class of attacks was 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 aren't 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 with in the organization network, gaining useful information as they move and exporting 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 as long as the funding organization needs. The attack only ends when it gets 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 victim of an APT attack, sometimes irrecoverable damage which is most common in the later case when the attack wasn't detected until all the organization's data fell into the wrong hands. The one question that an organization that is victim of an APT attack finds itself facing with is, why it wasn't able to detect the attack. Why, even after having security measures such as strong intrusion detection and prevention systems, did the attack go undetected. The answer to this question is what we provide in this paper.

The goal of an APT attack is not to just gather a target entity's data, but to do so 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 aren't usually detected by those signature based anti-virus software or intrusion detection and prevention systems. They gather every detail about the organization, what tools and techniques the organization uses, what applications does it host, what are the anti-virus, IDS/IPS used, and spend time in identifying the vulnerabilities in all these tools and creating malwares

The authors are with the School of Computing, Informatics, and Decision Systems Engineering, Arizona State University, Tempe, AZ 85281 USA (e-mail: adel.alshamrani@asu.edu; sowmya.myneni@asu.edu; ankur.Chowdhary@asu.edu; dijiang.huang@asu.edu)

that would use 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 cyber security firm reported several key findings of APT attacks performed by one of the largest APT organizations performing APT on a broad range of victims for long periods, starting from about 2006, by maintaining an extensive infrastructure of computers across the world. In it's M-Trends 2017 report, FireEye points out to the increase in the level of sophistication of financial attackers that is no longer any less compared to advanced state sponsored attacks. In support fo 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 year passing, the number of APT attacks being reported have 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 it's data. The defense methods should be employed at every phase of an APT attack.

The goal of this survey is to explicitly study the various techniques and solutions that were tailored to APT attacks detection, compare their characteristics, strengths, and weaknesses. In addition, this survey aims to point out APTs challenges and research opportunities.

II. Advance Persistent Threats

A. What is APTs?

Advanced Persistent Threat, as the name itself says, is not like a regular attack or attack done by a regular hacker. APTs are achieved often by a group of advance attackers that are well-funded by an organization or government to gain crucial data of 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: The APT attackers are capable to develop advanced tools by combining multiple attack strategies and launch multi-stages attacks.

Persistent: The APT attackers are highly persistent to achieve the goal and plan the evading techniques to avoid getting detected. This is usually done through persistent strategy maintained by the APT is "low and slow" approach.

Threat: The APT attackers focus precisely on specific organization to achieve their goals. They usually have the potential to harm an information system through destruction, disclosure, modification of data, and/or denial of service.

The attacker's goal often involves gaining access into the target, and ex-filtrating data about the target from the compromised systems. Once they break into the targets' system, they stay low to gather the information they are looking for and send it to their payer. APT attacks often involve several compromised nodes not just one unlike the regular attacks.

According to National Institute of Standards and Technology (NIST) [2], an APT attacker group : "(i) pursues its objectives repeatedly over an extended period of time. (ii) adapts to defenders' efforts to resist it. (iii) is determined to maintain the level of interaction needed to execute its objectives". Therefore, due to the aforementioned characteristics of APTs, attackers are capable to evade existing security systems and are difficult to prevent, detect and analyze [3].

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 foothold, internal network scanning, moving laterally from one system to another in the network to reach the target system that containing the sensitive data, and finally ex-filtrating this data to command and control center. Following ex-filtration, the attackers might chose to stay to continue extracting data as new data comes in or leave the system after cleaning up depending on the funding source's requirements. These multiple stages often involve, getting onto one of the systems of the network as a start, and then performing privilege escalations as necessary to reach the target system, followed by accessing sensitive data, and sending it over Organizations' Internet connection to the attackers' command and control center.

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

B. APT Attack Model: how does APT work?

APT attacks, as mentioned earlier, are well planned and highly organized so as to increase the probability of attack's success. And to be successful they perform attacks in multiple stages. To answer the *how* part of an APT attack, we take the help of attack trees. In [5], Schneier describes attack trees and their effectiveness in evaluating security of a system. With the goal of the attack as the root node, a properly constructed attack tree, though not limited to, 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 understand all the possible ways an attack could happen and accordingly place security measures for detecting or preventing the most risky attacks. In Figure 1, we have identified the different attack vectors and their combinations required by APT attackers in order to achieve their goal. The rectangular nodes represent APT stages, while elliptical nodes are the actions that the attackers can do to achieve their assigned attack goal in each of those stages. It is not necessary these

Table I: Comparison of traditional and APT attacks

	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

stages are found in every APT attack, but an APT attack usually is spread across these stages and they are:

1) Reconnaissance - The first step that APT attackers do is learn about the target. The more they learn about the target the higher is their rate of success.
2) Establish Foothold - This stage represents their successful entry into their target's computer and/or computer network.
3) Lateral Movement - Once they gain entry into the system, they need to locate the sensitive node(s) that hold sensitive data or are critical components of the organizations infrastructure which requires them move deeper into the network.
4) Exfiltration - When the attackers goal is to get the organization data, actions that comprise of retrieving and sending this data to the attackers' command and control center fall under this stage.
5) Cover Up - An APT attack would be successful and complete only when the attackers and the organization/entity that funded the attack cannot be identified/tracked requiring the attackers to remove any evidences that could trace back to them.

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. Figure 2 depicts the model of an APT attack performed in the above multiple stages.

1) Reconnaissance: One of the first steps attackers do is learn about their target. The more they understand the target, the more successful they could be with their attack. As part of this phase, attackers will do an extensive research to gather necessary information and intelligence about their target. The information that they gather during this stage is very vital to the attack's success. From details concerning individual employees such as their social life, habits, websites they often visit to details of the underlying IT infrastructure, such as the kind of switches, routers, anti-virus tools used, firewall, web servers, ports open, etc. the attackers would collect this information not only to establish 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 at site, port scanning, service scanning which refers to psychological manipulation of people into accomplishing goals that may or may not be in the target's best interest [4]. In addition, APT campaigns query publicly available repositories such as WHOIS and 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 network's layout [6]. In APT-based attack, reconnaissance usually is called passive, since attackers will not exploiting a victim, but instead are collecting data in preparation for a larger attack. Once APT actors collected enough information, they construct an attacking plan and prepare the necessary tools.

2) Establishing a Foothold:: 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.

Exploitation of Web 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), Open Source Vulnerability Database (OSVDB) [7], and NIST National Vulnerability Database (NVD) [8] 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 [9]. According to the reported study in [3], majority of APT attacks were based on known exploits. Therefore, it is essentially important to apply security patches shortly after vulnerabilities have been released.

Malware: According to Symantec's 2017 Internet Security Threat Report there were 357M malware variants in the year 2016 with email malware rate 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 Tree, 1, malware can be sent via spear-phishing, USB devices, web downloads.

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 send out emails with malware in it. These fradulent emails are cleverly crafted, well-enough to intrigue 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 foothold in the organization network.

Exploitation of known vulnerability: Known vulnerabilities are usually exposed and can be obtained from well-known vulnerability databases such as Common Vulnerabilities and Exposures List (CVE), Open Source Vulnerability Database (OSVDB) [7], and NIST National Vulnerability Database (NVD) [8] 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 [9]. According to the reported study in [3], majority of APT attacks were based on known exploits. Therefore, it is essentially important to apply security patches shortly after vulnerabilities have been released.

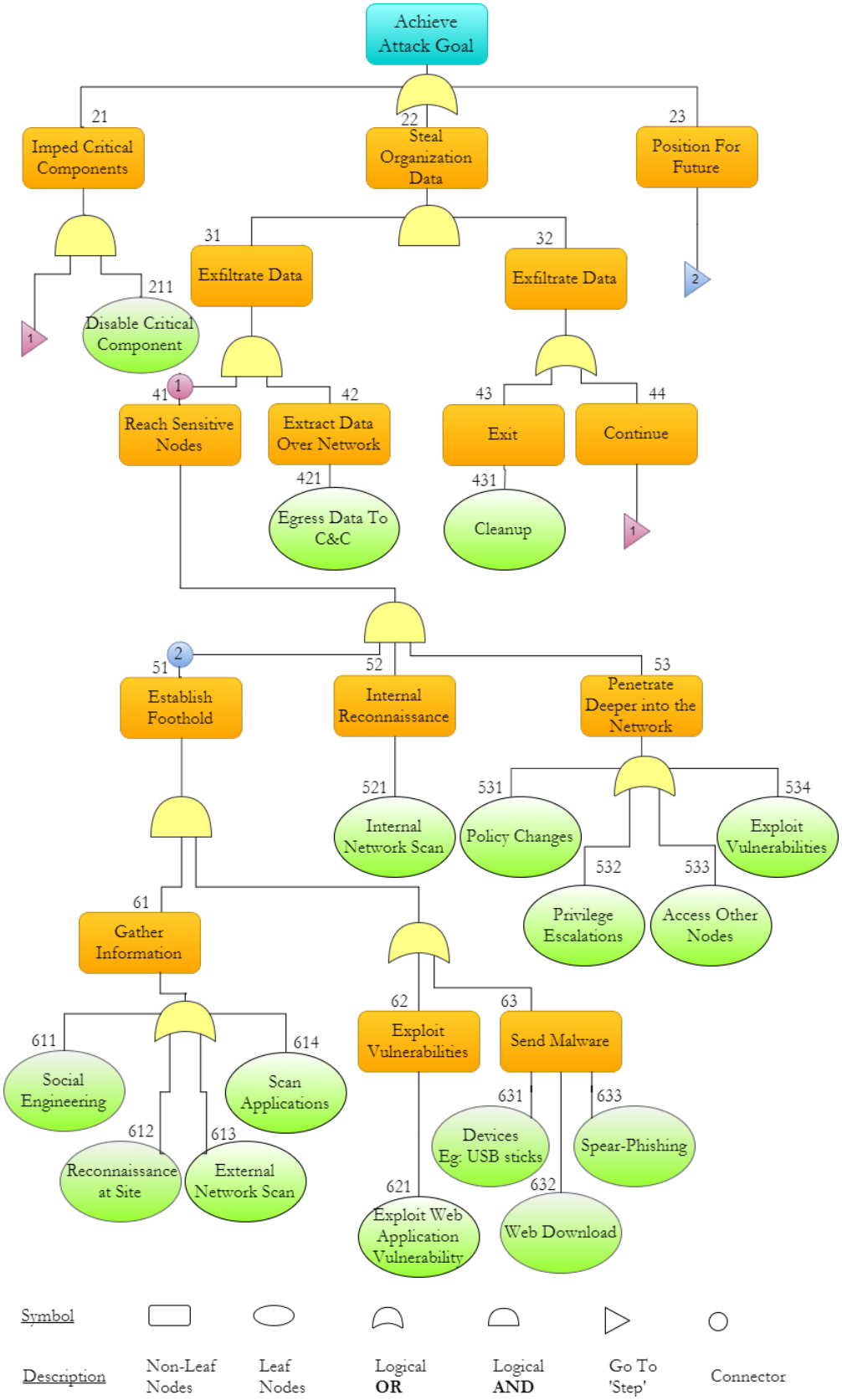

Figure 1: APT Attack Tree

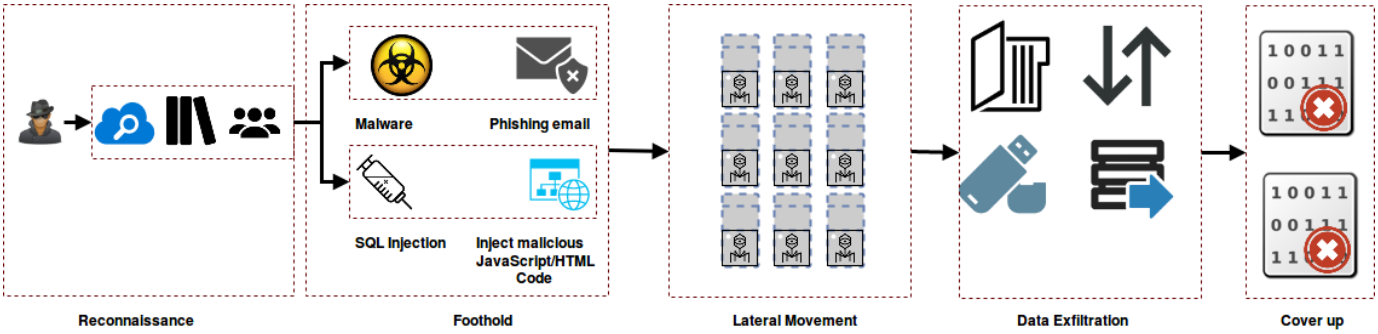

Figure 2: APT attack model

Zero-day vulnerability: A zero-day vulnerability is a software bug that either the software manufacturer is unaware of, or is aware of but wasn't able to fix it before the attackers could utilize it. APT attackers gather information about the organization's system components, such as the operating system versions, patches ran, software components installed on those systems including anti-virus, anti-malware components. They then go about identifying any vulnerabilities in those versions that could utilize to gain entry into the target's network. However, according to the reported study in [3], only few APT attacks were performed and achieved through zero-day vulnerability and the majority of APT attacks were based on known exploits.

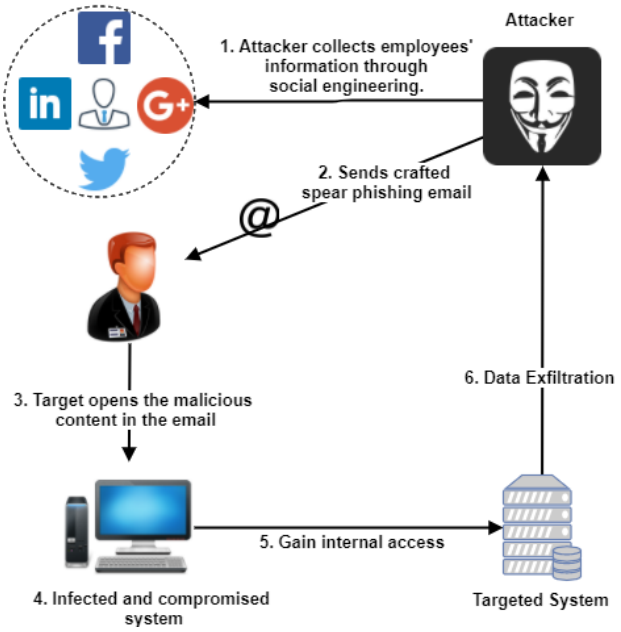

Figure 3: Spear phishing example

Lee and Lewis in [10] 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 involves constructing a 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 information helpful in analyzing the targeted malware. However, the problems with this solution is that there could be several attack nodes with out 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 an existence of APT attack.

Web Download: As mentioned earlier, spear-phishing emails could have malicious files attached to them that need to opened, or it could carry links to malicious websites that when employees visit, they unknowingly download malware. In addition, attackers might exploit vulnerabilities on one or more of the sites visited by the targeted employees and have the employees download the malware unknowingly when they visit the site. This later attack technique is called watering-hole attack.

Watering-Hole Attack: In case of watering-hole attack, the attacker can start checking those preferred websites and inject malicious contents to a vulnerable one. Considering the example in Figure 4 where an outsider attacker starts collecting information about the targeted organization X. We assume that he gets useful information about the organization's employees and what websites they heavily visit. The attacker then can start checking those preferred websites and inject malicious contents to a vulnerable one. Once the targeted user visits the infected website, the malicious code is loaded and the system got compromised. Now, the attacker has gained an access in the system through the malware, they can spread over the internal system and access more sensitive components such as servers. Attackers will be able to detect the applications running on the targeted machine and information that could be used later by an attacker to serve specific exploits. Finally, the targeted system's data will be exfiltrated. The presence of the malware on the compromised host can change its behavior by running unexpected processes, requesting unauthorized services, scanning the network, or attempting to authenticate to other hosts.

After sending out emails with malicious attachments or links that take to websites with malicious software, attackers patiently wait for the malware to run with in the organization's network that would open gates to their entry into 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 got the control of the system through the malware execution that exploits vulnerabilities in the system,

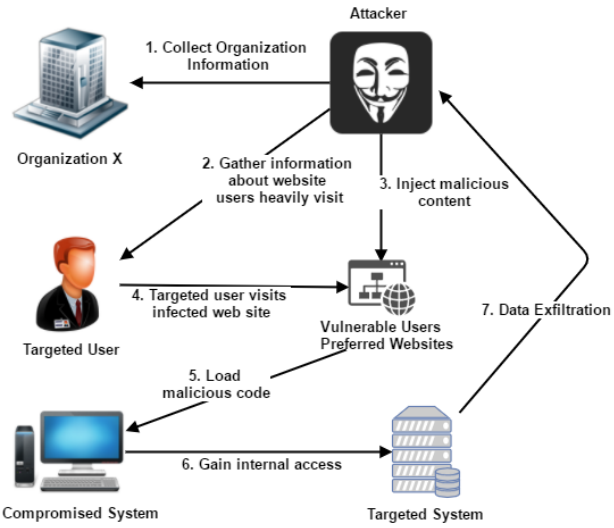

Figure 4: Watering-hole example

they keep low to go undetected to the next phase. At this point, APT attackers aim to build Command and Control (C&C) communication channel after infiltrating into the targeted network to deploy subsequent attacks. Most malware makes use of Domain Name System (DNS) to locate their domain name servers and compromised devices, so APT attackers can establish long-term connection to victims' devices for stealing sensitive data.

3) Lateral Movement: Now, once the attacker has gained an access in to the targeted system, he/she can spread over to other systems within the internal environment of the target. 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 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 [3]. Table II shows some techniques and methods used to accomplish lateral movement stage. Hash and password 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, and move stealthy within an environment. The main method for gathering hashes and passwords is to utilize appropriate applications that are able to dump this information from a system. Currently, mimikatz is 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 within 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 [3]. These tools are in use by both professional pen-testers and adversaries.

4) Exfiltration: When the attackers find the organization data they are in search of, they attempt to contact their command and control center and export this data to a remote location. Since most of the intrusion detection and prevention systems do ingress filtering and not outgress filtering, their data exfiltration could go undetected. Depending the organization's defense methodologies, the attacker's might intelligently split the data exfiltration into batches. Ullah et al.. in [22] summarize the latest data exfiltration incidents in 2017 as shown in Table III.

5) Cover up: The goal of an APT attack is not just stealing organization's data but to keep doing so until the attack has been lifted by the attack sponsor. The sponsor could chose to lift the attack once the data retrieved is what it wants or could have the attack still active to keep getting data as long as it 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 a permanent and continuous monitoring, the tools will usually be removed to cover up the tracks. This is usually achieved through establishing what is known as a backdoor so it makes a return easier.

C. 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 [3], [23].

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 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, non-government organizations.

A. Titan Rain

In 2003, a series of coordinated cyber attacks, later code-named $TitanRain$, have emerged that infiltrated several computers and networks associated with U.S. Defense Contractors

Table II: Techniques and methods of the APT campaigns ([3])

APT Group	Lateral Movement		
	Standard OS Tools	Hash and Password Dumping	Exploit Vulnerabilities
MsnMM (Naikon Group)[11]	✓		
Carbanak [12]	✓	✓	
Duqu 2.0 [13]	✓	✓	✓
Naikon APT [14]	✓		
EquationDrug (Equation Group) [15]			✓
Operation Cleaver [16]	✓	✓	✓
Shell Crew [17]	✓	✓	
Icefog [18]		✓	
Regin [19]	✓		
Anunak [20]	✓	✓	✓
Deep Panda [21]	✓	✓	

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	ESPORT ENTERTAINMENT (ESEA)	1.5 million	Locations, Login details, Email addresses, DOB, Phone NOs

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) [24] 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. But 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 being that 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 with in 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, has 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 with in 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 Ste7 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. 2 of the zero-days used in Stuxnet were not new in Stuxnet, they have been exploited earlier by other small malwares though weren't found at that time. After security researchers across the globe have digged 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 imped 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 [25] and [26]. But they all agree that Stuxnet was found to be like never before, a havoc that a digital code could create in physical world. It wasn't 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 then 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 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. Equifax APT Attack

Equifax, one of the three major credit bureaus, has been a target of APT attack in 2017. The Equifax data breach is one of the largest data breaches in history where the data of 143 million people were exposed for more than three months. As part of this breach, attackers were successful in retrieving the personal data of millions of Equifax customers. The breach reportedly happened due to an un-patched vulnerability in the APACHE STRUTS, CVE-2017-5638. NIST National Vulnerability Database (NVD) [8] assigned a score of 10.0 to this vulnerability, which is the highest score that can be assigned to a vulnerability. The APACHE STRUTS software was used by Equifax for one of its Portal websites. Attackers using this un-patched vulnerability were able to take control of this website, and thus penetrate into the system to extract the consumer data including Social Security numbers, home addresses, credit card numbers, drivers license numbers and birth dates.

IV. Classification of APT Defense Methods

As mentioned earlier, defending 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 network needs to be employed. Correlation of the events from these different defense measures will play 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 could be evaded. In addition, these layered defense measures give defenders time and a risk estimate that would help them with mitigation approach to employ.

L. Yang et al. in [27] 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. 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. That said, we now proceed with listing out the various defense methods that can be employed to defend against an APT attack, explaining the methodologies, and then presenting the current solutions for each of those methodologies along with pointing out the pros and cons of those solutions.

We categorize APT defense methods into 4 major categories, they are *Monitoring Methods, Detection Methods, and Deception Methods and Mitigation Methods*. Each category or class can be sub-categorized into different categories as shown in Figure 5. In the following subsections, each of these classes will be precisely explained.

A. Monitoring Methods

One of the basic and first step to defending APT starts with monitoring the entire network system. The system should be monitored at multiple points and multiple levels leaving no entry point un-monitored.

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

2) Memory Monitoring: One of the ways a malware can evade detection is by running from with in the memory of the end system rather than from a file. This so called fileless malware uses a process that is already running with in the memory to execute itself. As there is no separate process running in the background, it leaves no trace of it except the unexpected memory usage by a process which can be identified if monitored. Duqu 2.0 that infested Kaspersky labs in 2015, considered as step-brother of Stuxnet by some, ran with in the memory of already running process and thus bypassing 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 [28] 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 characteristics of key loggers, password thieves, which attempt to access the resources used by say, notepad that user is typing into, stealth backdoors that attempt to communicate with remote attackers. The common malicious activity that all these different types of malware exhibit is anomalous information access and processing behavior which the authors classified that anomalous behavior into anomalous information access, anomalous information leakage, excessive information access. For stealth backdoors which pertain to APT, the authors say that to go undetected the backdoors either use an uncommon protocol such as ICMP, create a raw socket, or intercept the network stack in order to communicate with remove adversaries. ICMP-based stealth backdoors will access ICMP traffic, raw-socket based stealth backdoors will access all the packets with the same protocol number. Example, a TCP raw socket will receive all TCP packets. The stealth backdoors intercepting the network stack will behave like network sniffer which would eavesdrop on the network traffic to obtain valuable information.

In [29], Virvilis and Gritzalis 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 unusual domains access, monitoring network connections, honeypots and honeynets, along with the standard host-based intrusions prevention systems as countermeasures for APT.

The authors of [30] 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.

In [31], Xu et al, 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.

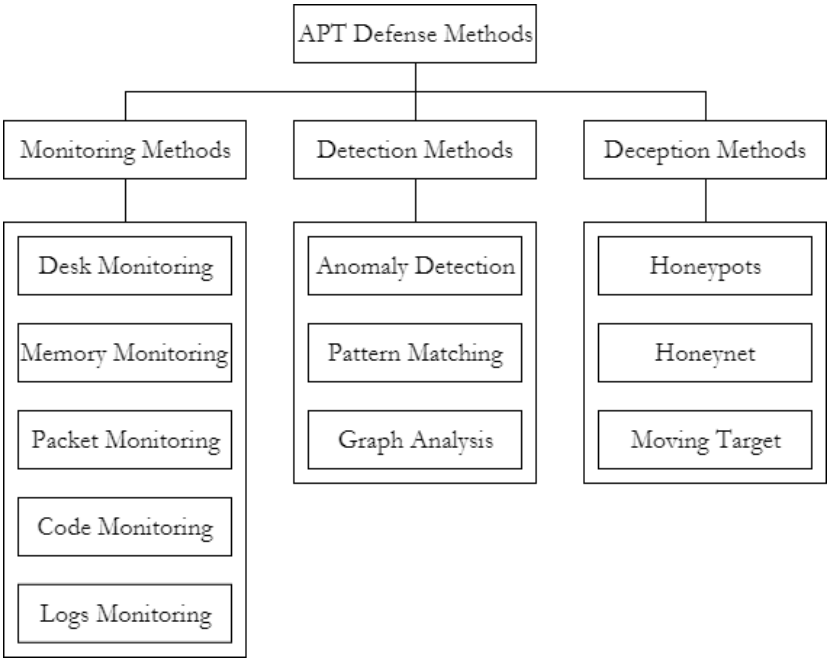

Figure 5: APT defense method classifications

Vaas and Happa in [32] 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, often multiple times, usually first time when the system has been compromised and repeatedly later for data transfer. Monitoring at end system level for, any network packets with new destination ip addresses, and packets with huge payloads, and large number of packets sent to the same destination ip address would help in identifying any suspicious behavior from with in an end system.

Marchetti and et al., in [33] proposed a framework that can detect, out of thousands of hosts, 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 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 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. These features could be numbytes (number of megabytes uploaded by internal hosts to external addresses), numflows (number of flows typically connections to external hosts initiated by internal hosts), numdst (number of external IP addresses related to a connection initiated by an internal host. According to authors, numbytes will allow to monitor deviations of uploaded bytes. Numflows will allow to monitor data transfers initiated by internal hosts. Numdst allows to identify anomalous behaviors that involve change in the number of distinct destinations contacted by each internal host. For example, if the number of external IPs contacted

within a given time window by an internal host remains stable while the number of uploaded bytes or connections greatly increases, it may correspond to a data exfiltration or APT-related activities.

In [34], authors 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 in [35] proposes 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.

Fu et al, in [36] 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 like a mirage. Every software developed, every code that you release can never be guaranteed to be error free. While having the code itself error free is quiet difficult, making sure that it is error free when running in different environments is not possible. These bugs are the means to attackers to penetrate into systems. While some of them could be known prior to the code release, there is always a possibility of unknown bugs. If the code is monitored at end system level, for its performance, and to make sure it runs with in its scope, neither utilizing unexpected resources or using up memory regions that otherwise aren't accessible, would lead to identifying a threat much earlier before it is spread to other systems.

5) Log Monitoring: Logs are an important part of not only forensic analysis, but when used appropriately can help in detecting or even preventing attacks in early stages. Correlation of these logs such as memory usage logs, CPU usage logs, application execution logs, system logs would give 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 aftermath for evidence of an attack.

One such paper that correlates data collected from different type of logs is [37]. Bohara and others in 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 with out any profiling the normal behavior of the system.

Shalaginov et al. in [38] analyze 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 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. In [39], the authors 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. in [40] discuss the kill chain attack model, and proposes 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 with in 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. in [41] propose 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

Techniques for detecting APT can be classified into the following group: *Anomaly based detection, detection by Pattern Matching, and Graph based detection.*

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 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 possible next steps.

Table IV shows the possible corresponding attack methods and detection and prevention techniques (defense) to each APT stage. However, these detection and prevention 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. Further, they could have these malwares behave as normal as possible with out raising any alarm to their behavior. Here comes the need to employ different learning methods that could detect these close-to-normal behaviors. Depending on the knowledge of the available data, these learning methods can be classified into

1) Supervised
2) Semi-Supervised and
3) Unsupervised

These learning methods have been part of machine learning field field for several years, applied to several domains for the purpose of detecting anomalous behavior from normal behavior.

Table IV: APTs stages and corresponding detection and prevention methods

Stages	Attack Methods	Defense Methods
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

Machine learning (ML) is defined as the ability of a machine to vary the outcome of a situation or behavior based on knowledge or observation. Using algorithms that iteratively learn from data, ML insights many technologies to analyze the data immediately as it is collected to accurately identify previously known and never-before seen new patterns.

Machine learning can be applied in different cases either the desired outcome is known, or the data is not known beforehand. In ML, the models can interact with the environment to learn from the collected data for different purposes. ML tasks are typically classified into three well-known forms of knowledge, depending on the nature of the learning available to a learning system. The most crucial part of anomaly based detection is identifying what constitutes normal behavior and what constitutes anomalous behavior. There are several works on identifying the difference between these. While some of them use trained data set representing the normal behavior of a system or network, some use models that learn about the system so as to identify the anomalous behavior.

Hodge in [42] has surveyed different outlier detection technologies. He classified outlier detection technologies into 3 types which are: First, Unsupervised clustering: this approach 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. Third type is: semi-supervised recognition or detection that models only normality or very rarely models abnormal data. This approach requires pre-classified data but only learns to recognize abnormality. Suitable for static or dynamic data as it only learns one class which provides the model of normality. It can learn the model incrementally as new data arrives, tuning the model to improve the fit as each new exemplar becomes available. It aims to define a boundary of normality. Unlike supervised classification, this doesn't require any training data for abnormality. Hodge classifies the outlier detection methodologies into the following: First, statistical which further can be classified into Proximity based (not feasible for high dimensionality data sets as it suffers from possible exponential growth of the data), parametric methods (suitable for large data sets, the model grows only with model complexity not data size, through they limit their applicability by enforcing a pre-selected distribution model to fit the data), non-parametric methods (flexible and autonomous), semi-parametric methods (combine the speed and complexity growth of parametric methods with the model flexibility of non-parametric methods). Second, neural networks

(generally non-parametric and model-based, and are capable of learning complex class boundaries) further classified into supervised neural methods (require a pre-classified data set to permit learning), and unsupervised neural methods (do not require any pre-classified data). Third, machine learning (Most statistical and neural methods require cardinal or at the least ordinal data to allow vector distances to be calculated and have no mechanism for processing categorical data with no implicit ordering). Fourth, hybrid systems which can involve more than one of the previously mentioned types. The author concludes the paper pointing out the fact that no single outlier detection methodology would suffice and often a combination of the methodologies have been used and been successful. The author also adds on how a developer wishes to handle outliers is very important, whether they wish to expunge (blacklist outliers) them from future processing in a diagnostic cluster or a retain them with an appropriate label in a accommodating clustering or a classification system. He ends up suggesting that they can chose an approach such as k-means or support vector machines that store only a minimal set of data that efficiently covers the distribution through a small subset of seminal exemplars.

Email spam has been known as a major method attackers use to launch APT. However, 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 spams.

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 is not the good option 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.

In [43], Chandola et. al. 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. They further discussed the different applications of anomaly detection in terms of the notion of anomaly, nature of the data, challenges associated with detecting anomalies and existing anomaly detection techniques. They point out that the huge volume of the data is a key challenge for anomaly detection in intrusion detection. While labeled data for normal behavior is usually available, labels for intrusions are not, thus making semi-supervised and unsupervised anomaly detection techniques as possible techniques in this domain. They also point out the fact that the anomalies keep changing over time as the intruders adapt their attacks to evade the existing intrusion detection solutions, thus making this as another key challenge to overcome when addressing intrusion detection. In addition to the anomaly detection techniques classified and discussed by Hodge an Austin in [42], the authors of this paper have discussed 2 other point anomaly detection techniques, Information Theoretic anomaly detection techniques and Spectral anomaly detection techniques, both of which can operate in unsupervised setting with the former one making no assumptions on the underlying statistical distribution for the data while the later one can automatically perform dimensional reduction making it suitable for handling high dimensional data sets.

Table V: APTs stages and corresponding AI/ML roles

Stages	AI/ML Role	AI/ML Techniques	Challenges
Reconnaissance	***	***	***
Accomplishing a foothold	Pattern Matching (classification)	Supervised ML	High False Positive Rate
Lateral movement	Grouping similar activities (Clustering), pattern matching (classification)	Supervised & Unsupervised ML	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

Table V summarizes the role of AI/ML techniques to defeat APTs. The following examples can be matched to different APT stages and detect their methods using 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 domains DNS 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 is not the good option 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) 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 in detecting one of the common APT stages which is *the attack expansion*.

D) Deeply analyzing 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) Detecting if, for example, CFO's computer makes unexpected financial transaction based on transaction's time, destination, etc. This is 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.

The authors of [44] 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 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.

The authors of [45] have 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 related work is quite interesting. IDS can be classified into host-based and network-based and hybrid. Another classification, misuse detection, and anomaly based detection. Misuse detection by comparing events with the security policies. These further classified into signature based and rule based. Anomaly based further can be classified into statistical based, distance based, rule based, profile based, and model based. Another classification for the same can be statistical, machine learning, neural network techniques, data mining techniques and computer immunology techniques. Anomalies can cover three different kinds such as *Point anomalies, Contextual anomalies, and Collective anomalies*. Anomaly detection methods can be classifier based, nearest-neighbor based, clustering based, statistical, information theoretic, and spectral. Neural networks, Bayesian networks, Support Vector machines based, and rule based fall under classifier based anomaly detection. Statistical anomaly detection technique assumes the input data to follow a stochastic model. 2 sub techniques can be parametric and non-parametric techniques. Coming back to their proposed scheme, it uses log data produced by various systems and components in ICT networks. From this log data, their solution extracts a system model, that is used further 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 continuously generated to detect anomalies that are consequence of realistic APT attacks.

The authors of [46] have proposed a model of APT detection problem as well as methodology to implement it on a generic organization network. Their solution considers 3 types of events, 1. candidate events, all events that are recorded by an organization logging mechanisms in any form, 2. suspicious events, events reported by security mechanisms as suspicious, or events associated with abnormal or unexpected activity, 3. attack events, events that traditional security systems aim to detect with regard to a specific attack activity. The authors use risk level and confidence indicators to evaluate the threats to attack goal. Thus, 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.

The authors of this paper [47] 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 common signature based approach brings 2 major drawbacks - pre-defined rules often being insufficient to detect unique or tailored attacks, and lack of rules that verify application specific operation sequences.

Existing technologies such as rule-based analysis require skilled analysts to involve in analyzing the behavior of malware and design rule-based solutions to predict similar behaviors in the future. However, human involvement to analyze malware behavior may take time and result in a gap between discovery and protection, which is enough to cause harm to an organization [21]. However, instead of having analysts manually analyzing and encoding malware detection and prevention tools, these tasks can be assigned to machine learning algorithms that continuously monitor, learn, train, and update a deployable machine learning models. Another major issue to consider when comparing human involvement and machine learning is the ability to detect minor behavior patterns. Machine learning algorithms can easily observe and detect such minor changes, where human analysts may not observe those changes.

The major idea behind detecting APT is to identify some unique features in the APTs behavior and then track the uniqueness of those features. The unique features are then used to identify any possible deviations from intended behavior and efficient defense mechanisms are formulated. Machine learning algorithms, such as Perceptrons, Neural Networks, Centroids, Binary Decision Tree, Deep Learning, etc, can help processing millions of data points every minute to establish normal behavior and compare data points to past behavior and identify anomalous differences in values. In machine learning technology, collecting data for statistically analysis

is an important step to start with. For APTs detection based mechanisms, malicious data and benign data usually combined to build reasonable dataset that contains multiple features of both malicious and non-malicious objects.

Siddiqui et al. in [48] state that many approaches have been developed to continuously monitoring and analyzing TCP/IP connections to extract multiple features that can be used by machine learning algorithms to utilize statistical and behavioral knowledge. However, authors believe that current related works have some limitations in terms of reducing false positives and false negatives. They propose a fractal based anomaly classification algorithm to reduce both false positives and false negatives. Their algorithm is based on fractal dimension representation of the network layer level command and control communication protocol of APTs. They use K-Nearest Neighbor (K-NN) machine learning algorithm. The dataset that was used in their study is a combination of two different data sources to cover both APTs traffic and non-malicious traffic. First, the APTs were collected from Contagio malware database [49]. Second, normal and non-malicious data is obtained from PREDICT internet data set repository [50] under the category of "DARPA Scalable Network Monitoring (SNM) Program Traffic". Authors apply two approaches to test the combined dataset, the first approach uses traditional supervised learning i.e. K-NN and the second approach uses correlation based fractal dimension. They set the value of k=3 neighbors for K-NN and they obtained better performance in terms of reducing both false positives and false negatives when using Correlation Fractal Dimension approach. This is reasonable due to the fact that the capability of the second approach to extract multiscale hidden information.

Nath et al. in [51] focus on comparing machine learning methods that have been used to statically analysis malwares. They mainly focus on four methods: n-Gram, Byte Sequence, OPCODES, and Portable Executable Header (PE-Header). N-Gram analysis is a method of analyzing byte sequences in a file where n stands for the number of bytes taken to form a single sample of malware. N-gram method can be taken in two ways either in the form of disjoint n-grams or over lapping n-grams. This process can generate distinct n-grams of multiple malware samples. Later, the distinct n-grams can be used for training a machine learning classifier such as naïve Bayes, support vector machine, decision tree, etc. Byte Sequence method is mostly used to identify the packed and encrypted malwares. It is based on the assumption that if a file is encrypted, then its byte sequence would be very much scrambled. When applying machine learning techniques such as measuring information entropy to identify if the entropy value is high then that indicate the file is encrypted and it is possible that the file is used to conceal the code in a malicious executable. OPCODE is a single instruction that can be executed by the CPU. The frequency of appearance of opcode sequence is the mainly feature machine learning classifiers look for. They summarized the best features of opcodes for classification such as like bt, fdvip, fild, fstcw, imul, int, nop, pushf, rdtsc, sbb, setb, setle, shld, std. PE-Header is a method used to extract header information "features". The major assumption here is that it is possible to predict the malware's behavior through analyzing the PE-Header.

Authors in [52] propose an approach to use machine learning techniques to contextually analyze network traffic alerts. Their approach aims to find APTs presence in the collected network traffic alerts. They have splitted their approach into three major steps which are: data collection, data analysis, and APTs exploration. During the data collection step, they collect raw data (network traffic) using WireShark tool. The result is a multivariate table where rows represent messages and columns represent attributes. They utilize the following protocol attributes: port numbers, ip addresses, flag data. A network traffic alert is modeled as a weighted vector of message value, where the weights describe to which extent the alert is considered malicious. They apply machine learning techniques on the collected network traffic alerts for the goal of splitting alerts into both messages and attributes. During the exploration step, the proposed approach considers three phases namely discovery, identification and confirmation. The idea behind these phases is that discovering what kind of alerts are discovered, what their causes, and finally build baseline to ensure the usability of their explorations. The proposed 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 in [53] presents a preliminary study on using deep learning-based technique for malware detection. 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, it performs much slower than the conventional models.

To detect security breaches that are designed to a specific target, et al. in [54] claim that deep packet inspection (DPI) and anomaly detection are indispensable. Authors propose an approach for network traffic analysis where they consider visualization and machine learning techniques. Therefore, system administrators can inspect and compare specific parts of the network traffic in parallel 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. 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.

Authors in [55] propose 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 by collected Symantec's 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 can be due to the limited amount from information that can be gathered from LinkedIn.

In [56] a ChainSpot framework was proposed for mining service logs for detection cyber security threats. It aims at summarize user's sequential behaviors while he/she is using application-layer services to discover deviations against one's normal patterns. The proposed framework was evaluated on a dataset that contains around 1000 employees. The dataset contains three types of data where two of them are service logs, active directory and proxy logs, while the other one is security operation center (SOC) event tickets, labeled with various security anomaly events by forensics experts providing ground truths specifying which IP related to certain accounts behaves abnormal or not. A total of 152.299 gigabytes logs data were collected over one month.

Paper [57] focuses 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 design three layers, to focus on the use of discriminative supervised/semisupervised models to identify behavior primitives.

Authors in [58] propose 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.

Paper [59] presents a big data analytics approach for anomaly detection. The approach's architecture contains four phases which are: data collection, data management, analytics trainer, and detection alert system. This approach mostly collects Network Log (NetL) and Process Log (PrcL), the NetL records all raw network data in form of packets travelling from/to the system, while the PrcL audits system processes in instant memory. The collected data are stored in local database for further inspection. The features selected from NetL contains three unique sets such as ethernet set which consists of source/destination physical addresses and the type of physical media. Second, IP set that includes destination and source IP addresses, type of service and protocol type. Finally, the third last set represents Application, with TCP type traffic with destination and source port along flag type. The features selected from PrcL component represent the existing process in system and belonging user ID, process ID, parent process ID, session, terminal type and command line arguments. Once the dataset has created of the collected features, the approach starts the correlation process to identify similarities among different processes in the system. The K-mean classifier was used with K=4. The final output of the classifier results in 9 different clusters, however, most of the data points are clustered in one cluster.

Marchetti et al. in [60] criticized that traditional defensive solutions such as signature-based detection systems and anti-viruses 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 to 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. The gathered information basically obtained from three sources which are: *network logs* that allows the identification of weak signals possibly corresponding to APT activities; *simplified assets list* that maps the members of the organization and their client devices. This information is useful to link technological and open source data; and finally from *OSINT information* which is collected from public open sources to identify and quantify the information that is available to APT attackers. To evaluate their framework, they collect and analyze network traffic generated within a large organization with 6,432 internal clients. They focus on three major stages of APTs which are foothold, lateral movement, and data exfiltration, and prioritize all internal clients that show suspicious activities.

Authors in [61] propose a machine learning model that used J48 decision tree classifier to detect phishing emails. Their model trained on 23 features that generated from an email's 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 on 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.

Authors in [62] 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).

Authors in [63] design an interactive system to bridge the gap of network management and anomaly detection. They design 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. To build their visualization tool, they use VAST Challenge 2013 Mini Challenge 3 dataset which contains network security data from an international marketing company that has around 1200 workstations and servers. The dataset contains of 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.

Paper [37] presented an intrusion detection approach that combines the network and host logs to find any malicious activity. Their approach involves unsupervised machine learning technique Clustering to host-based, network-based monitored data. Their threat model involves 2 categories of security incidents in enterprise network, first one includes network scan attacks and flooding attacks, and second includes detecting existence of malware on the host. They aim to detect both type of incidents by clustering the network logs and host logs to find dissimilarities. They extract 4 categories of features to capture the behavior and join the data across different monitors. Those feature categories are Identification features (they aggregate the raw log entries for a given IP address for each 1-minute window), Network traffic-based features (from firewall, number of unique source and destination ports used, the number of TCP connections built and torn down, and the number of distinct administrative services used such as Telnet, Finger, FTP), Service-based features (from firewall data for each source IP address - the number of accesses to workstations, the number of accesses to the domain controller servers, the number of accesses to database servers and the number of accesses to any other type of server), Authentication-based features (from system log data by counting the failed login attempts, logon attempts using explicit credentials, special privilege assignments to new logons, computer account changes, generation of Kerberos authentication tickets, NTLM authentication attempts, administrative logons, anonymous user logons, anonymous target user names, local interactive logons, remote desktop logons, requests for session keys, port numbers that are either zero or blank and logons by distinct process IDs). The extracted features are then refined to reduce redundancy through the use of Pearson correlation coefficient, and remove those that do not contribute towards clustering.

Kumar et al. [64] propose a framework to detect security intrusions using a hybrid approach of rules and machine learning techniques. They include multiple incorporating rules such as domains reputation, IPs reputation, files reputation, applications reputation, whitelists, and blacklists. This can help in detecting known malicious activities. Moreover, they apply the rules based techniques as filter on the machine learning output. They limit the implementation of this filter to only detect logins from unusual geographic locations using machine learning technique and apply the defined rules to reduce the false positive rate.

2) Sandboxing Techniques: The concept of sandboxing techniques is relatively straightforward. Usually a virtualized environment is designed with the goal to allow a malware to recognize it as an ordinary PC workstation. The virtualized environment is kind of isolated environment to safely test and monitor the behavior of a malware. Sandboxing techniques identify APTs by their behavior instead of looking for static characteristics which can be easily obfuscated [21]. Unfortunately, APTs can be set up to identify when they were operating in a virtualized environment. Thus, APTs author can adjust their techniques to evade sandboxes techniques.

Authors in [65] formulate a defense mechanism against APT based on the interaction between an APT attacker and a cloud system defender. They formulate the defense mechanism as a Colonel Blotto Game (CBG) to model the competition of two players under given resources constraints over multiple battlefields. The game is based on challenging tasks and it is two player zero-sum game and the player who allocates most resources to a battlefield wins the game, and the overall payoff of a player is proportional to the number of the won battlefields.

3) 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 et al. in [66] propose 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 [67] which was originally introduced as a Natural Language Processing (NLP) for behavior recognition in mobile sensing problems was utilized in their approach.

Database access logs: The goal of APT attackers is to stay quiet for as long as they can while extracting data from the target. This data in the organizations' databases point out to the fact that APT attackers would frequently, and/or periodically access the database to obtain new data and then transfer it. One way of monitoring this pre-exfiltration phase is through monitoring the database access through the database access logs.

Honeypot strategies: For attackers looking for data to extract once with in the organization's system, the honeypots setup by the organization with data to attract the attackers would be a bait to those attackers, specifically for the APT attackers. A period of frequent access to these honeypots could give them away.

4) Graph Analysis: Graph analysis is one of those fields that is being chosen for its ability to support analysis of complex networks and identifying sophisticated attacks. Johnson et al. in [68] have proposed a novel approach to measure the vulnerability of a cyber network through graph analysis. Their solution specifically detects those attacks that would involve lateral movement and privilege escalations using Pass the hash techniques to achieve the attack goal. The attack is detected by the use of a simple metric that measures on a graph how likely a node is reachable from another arbitrary node, potentially making the network vulnerable. This metric is dynamically calculated from the authentication layers during network security authorization phase and will enable predictable deterrence against attacks such as Pass-The-Hash (PTH).

Attack Graph has been used a 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 pre-condition 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 took by attacker in APT scenarios, as an ordered sequence of events that led to compromise of the system. Another advantage of using attack graph is the ease of estimation of attack cost and return of investment (ROI) for a particular

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

Table VI: Attack Graph and Attack Tree based Methods

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

C. Deception Methods

One of the characteristics of APT attacks is the level of sophistication employed to perform the attacks. We cannot guarantee that all our defense tools can protect us from the evolving malware and attack forms. But being prepared is one of the best defense methodologies. And here comes the deception technology. In this defense methodology, defenders deceive the attackers by creating baits in the forms 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.

The authors of the paper [77] have 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 2 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.

Anagnotakis et al. in [78] have 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 doesn't 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.

The authors in this paper [79] have 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 instrumented 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.

This paper [80] discusses the importance of reconnaissance defenses involving deception and movement. They point out to the facts 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 about potential targets in a network. And so, the authors proposed probabilistic models that given the benefit and cost associated with reconnaissance defenses help 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 which was better than defense by movement in the evaluated scenarios though having both would be 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* [81].

- **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.

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 used to prevent APT attacks in the table below:

The MTD techniques discussed in Table VII 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 [92] are changes in network policies on reconfiguration of network resources are other aspects that needs to be analyzed carefully before selecting appropriate MTD solution.

V. EVALUATION

One of the most critical aspect to ensure the effectiveness of APT attacks detection solutions is the evaluation part. In

Table VII: Moving Target Defense techniques against APT's

Paper	Details	MTD Technique	Classification
[82]	SDN-based solutions for Moving Target Defense	OS hiding, Network reconnaissance protection	Diversity, SDN, Network MTD, Host MTD
[83]	Target movement based on attack probability	VM migration	Shuffle, SDN, Network MTD
[84]	Openflow based Random host mutation	Physical IP mapping to corresponding virtual IP	Redundancy, SDN, Network MTD
[85]	Fingerprint hopping method to prevent fingerprint attacks	Game theoretic model for fingerprint hopping	Shuffle, SDN, Game Theory, Host MTD
[86]	Dynamic game based MTD for DDoS Attacks	Dynamic game for flooding attacks	Diversity, SDN, Game Theory, Network MTD
[87]	Security Models for MTD	Effectiveness analysis for MTD countermeasures	Shuffle, Diversity, Redundancy, Network MTD
[88]	Dynamic MTD using multiple OS rotation	Latency analysis, OS rotation	Diversity, Host MTD
[89]	Optimal MTD strategy based on Markov Game	Dyanmic game, MTD Hopping	Shuffle, Game Theory, Application MTD
[90]	Software Diversity and Entropy based MTD	Cost, usability analysis of software diversity	Diversity, Application MTD
[91]	Decoy based cyber defense using Randomization	IP address Randomization	Redundancy, Game Theory, Network MTD

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, then the challenge is that how 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 part of APT attacks detection methodologies lacks of 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 [93].

It has been noticed that most of 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 one (synthetic model). The real network scenario has advantages such as realistic test basis, however, it has disadvantages such as bad scalability in terms of user input, varying scenarios, privacy issues, attack on 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 the goal is to create a model which allows simple reproducibility. However, the drawback of using synthetic based model is that APT attacks are based on simplified attack scenarios and are deployed in controlled environments where no realistic noise involved in the collected data which is one of the major point APT attackers consider to stay undetected and move low and slow. Other APT studies use semi-synthetic which 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 applied [93].

The second important component is the feature selection which is a major aspect that affects the results when using machine learning to solve a task. Usually, the collected data are raw data and cannot be directly used to 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 is used in another solution. Usually, the problem formalization has an influence on this task and determine which features can be selected. For instance, when mining and investigation logs data, the available features are not similar to the features 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)[94]. Table VIII presents list of common features against mining techniques.

VI. Challenges

A. Infrastructure-based Challenges:

One of the major challenges makes this problem more severe is that cloud computing systems are continuing to grow, therefore, evaluating their vulnerability to cyber-attacks become more challenging and difficult. Large systems that build upon multiple platforms and diverse software packages and supports several modes of connectivity will contain security vulnerabilities that have not been noticed of even most diligent system administrator.

B. Attack-based Challenges:

The following reasons summarize why an APT attack goes undetected.

- Failure to detect the attackers' entry into the network.
- Failure to detect attackers' asset discovery process.
- Failure to detect the illegal access to the sensitive data.
- Failure to detect the extraction of sensitive data to a remote location.
- Failure to detect the post attack activities.

C. Internal Employees

As mentioned above that APT attacks involve gathering useful information about targeted organization such as collecting employees names, emails, addresses, etc. This is usually

Table VIII: Mining techniques against common features and targeted APT stages

	Common Features	Targeted APT Stage
Emails		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 [95]	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		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

done using social engineering techniques which rely on either naivete 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 either of the following 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 whom employees may share information with and how that information should be transmitted. Create official channels for security and IT personnel to contact staff, 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 employee information. Educate staff to not provide any information to outside people unless that is under known and approved procedures and how they should hold up on 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 reducing the chance of APT attacks.

D. Powerful Resources

In some cases, APT actors are sponsored by very large organizations, governments and countries. APTs are usually have financial or political motives, for example, Stuxnet, which took down Iran's nuclear program, and Hydraq. Therefore, they are usually powerful and have enough resources, well-funded, and have a long-term perspective on their mission. The APT actors are highly-skilled, determined and are able to design new, covert, and advanced attacking methods. In addition, APT attacks are inherent to the combined use of different attack vectors and evasion strategies. This makes detecting and mitigating APTs very difficult and shows why purely technological defenses cannot work. Therefore, there must be new defenses based on a combination of human analysis and automatic detection of stealthy movements likely formalizing an APT.

VII. Research Opportunities

In this section, we identify several opportunities that are worthy of investigation by the research community in order to develop APT attack detection mechanisms that meets the requirements and needs of current IT environments.

A. APT Stage-based Specification

Since APT attacks are done through multiple stages, *reconnaissance, foothold accomplishment, lateral movement, data exfiltration, and cover up*, each of these stages worth a separate study and investigation. The earlier you contain an APT, the easier would be the damage reversal. Therefore, Table **??** maps some opening research opportunities to APT attack model stages.

B. Multiple Sources of Information

Existing deployed security systems mostly are optimized for processing large amount of system data (e.g., event logs) and are therefore highly automated. This makes the detection of sophisticated, intelligent, and complex attacks difficult. Although these systems are able to identify individual characteristics of an attack, it is often necessary to perform an additional investigation of complex attacks to reveal all malicious activities. Thus, due to the stealthy nature of the APT attack, where the sequence of malicious activities are performed at low frequency, any proposed security system must be robust and accurately able to reveal important hidden details in the system that can lead to detecting intelligent malicious activities. Therefore, it is possible to design a solution that involves multiple components, such as system logs information, system vulnerability information, network configurations, machine configurations, intrusion detection alerts, etc, each with an ability to derive important observations that, when combined, can lead to detecting, tracing, locating, and optimally responding to potential APT attacks. These components look for parts of attack sequences that are visible within their own data.

C. Hacker Community Investigation

Investigating hacker community can help to identify vulnerabilities before being exploited. According to [96] some vulnerabilities have been discussed by black-hat community before publicly exposed by ethical organizations. Community

of hackers usually interact through a variety of means such as online forums. They also market and provide a new avenue to gather information about the cyber threat landscape. Hackers interact and communicate with each other through forums which are user-oriented platforms that have the sole purpose of enabling communication. It provides the opportunity for the emergence of a community of like-minded individuals regardless of their geophysical location. Administrators set up dark-web forums with communication safety for their members in mind. While structure and organization of dark-web-hosted forums might be very similar to more familiar web-forums, the topics and concerns of the users vary distinctly. Forums addressing malicious hackers feature discussions on programming, hacking, and cyber-security [97], [98]. In general, a wealth of cybercriminal knowledge is distributed among hacker forum participants, this allows individuals to advancing their capabilities to commit cybercrime by utilizing available and shared resources. Visiting different hacker communities and consuming their contents rapidly help hackers to gain knowledge and capability. The existence of such hacker communities is common across various geopolitical regions, including the US, Russia, the Middle-East, China, and other regions. This presents a growing problem of global significance. Research in this area has potential for high-impact on society [96].

D. Monitoring Methods

So far, we have discussed several solutions that would help detect an APT attack at its different stages. However, there is no one solution that would detect the malicious behavior at system level and use it to detect an APT. We now present a theoretical analysis solution that would detect APT in earlier stages. This solution is an unsupervised learning method that utilizes monitoring information collected at different system components. We believe, all the monitoring methods, CPU monitoring, Memory Monitoring, Log Monitoring, Code Monitoring will become a strong defense mechanism in protecting a system when the the results of all those methods obtained from each end system are compared with their past and intertwined with the results from other end systems of the organization's network to yield a map that projects any unusual patterns. To exemplify this, consider 2 end systems A and B. Let s_1^a and s_2^a be the processes running on A, and s_1^b and s_2^b be the processes running on B. Let's say m_i^a and p_i^a are the results from Memory Monitoring and Packet Monitoring respectively from system A at time t_i in a day d_k. Similarly consider m_i^b and p_i^b as memory monitoring and packet monitoring results from system B at time t_i in a day d_i. When m_i^a for s_1^a is compared with the past results,

$$\sum_{j=0}^{k} \sum_{q=0}^{i-1} d_j^a \; m_q^a$$

of s_1^a obtained over a period of days, say, gave an indication that s_1^a at time t_i has run for excessive time with unusual memory usage. Given this, say the packet monitoring results showed that process s_1^a has connected to one new IP address IP_x at time t_i, identified when compared with the packet monitoring results over the past. While the individual results m_i^a and s_i^a do not yield much information, when woven together, they give out a much sensible information, that points out a possibility of a fileless malware running under process s_1^a at time t_i that tried to connect to its Command and Control Center. Now, if this information obtained from end system A is intertwined with the information obtained from end system B, where end system B results indicate yet another new IP address IP_y from the same process s_2^a, the resulting information confirms the earlier possibility of a fileless malware spreading out to other systems through a vulnerability in application owning the process s_1 hiding by communicating with different IP addresses that turned out to be proxy servers. Thus by intertwining and mapping against the results across monitoring groups and across the systems in the organization's network, a finer detailed and sensible information can be obtained that would detect APT attacks much earlier stages just like in the above exemplification.

VIII. Discussion

Since the report of the first APT attack, several works came into existence, while some have studied APT in terms of malware, spear-phishing attacks, some in terms of exfiltrating data. From detecting a possible APT attack through collected reconnaissance information from social profiles 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 APT attack. Very few have studied and addressed defending APT attacks in complete. In [99], the authors have discussed APT attack stages and suggested educating users and system administrators about the attack vectors as first step followed by implementing stricter policies and static rules and to use software tools such as SNORT to detect anomalies. APT attacks are beyond a single tools or user awareness. And often, implementing stricter poicies not only difficult but also doesn't suffice for APT attacks. All the attackers need to do is steal an account that has permissions to several entities that they can penetrate into. Their work failed to realize the challenges in detecting an APT attack and the possibility of new attack vectors that evolve each day. In [3], the authors have analyzed several published reports of APT attacks and came out with a 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 of those APT attacks they studied were mostly known vulnerabilities and exploiting zero-day vulnerabilities are rarely involved. Chen and Desmet in [4] have studied APT attacks deeper than their several contemporary works, from analysis of APT stages through case studies of APT attacks, countermeasures to be taken and several detection methods when employed could help detect APT attacks. Though their work gives an overall idea of APT attack, it lacks study on defending against

APT attack by collecting data from different sources. In [100], Tankard et al. have studied APT attacks, explained the different stages, and discussed the detection techniques to employ for defending against APT attacks. This was one of the earlier works of APT attacks and it gives good foundation of what APT attacks are. However, this work like others discuss the attack vectors like we have studied and analyzed, 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 the several challenges involved in huge volume of data analysis. Therefore, in order to ensure the novelty and new contribution of our survey, we thoroughly compare our work with existing surveys as shown in Table IX.

IX. Conclusion

We discussed a class of attacks that are complex, persistent, and stealthy, referred to as Advanced Persistent Threats (APT) which often leverages the bot control and remotely access valuable information. Using multiple phases to break into a network the goal of an APT attack is go unnoticed, continuously and for long periods by bypassing the existing security detection systems and penetrating deeper and deeper into the network. Often entering into the organization's network via spear phishing or watering-hole-attacks via which malware is sent out, the diversity and stealthiness of APT makes it a challenging threat to detect and mitigate. In this paper, we presented how an APT attack is performed, different solutions at each stage of an APT attack that would help detect/mitigate the attack.

Acknowledgment

The authors would like to thank...

References

[1] D. McWhorter, "Apt1: exposing one of china's cyber espionage units," *Mandiant. com*, vol. 18, 2013.
[2] R. S. Ross, "Managing information security risk: Organization, mission, and information system view," *Special Publication (NIST SP)-800-39*, 2011.
[3] M. Ussath, D. Jaeger, F. Cheng, and C. Meinel, "Advanced persistent threats: Behind the scenes," in *Information Science and Systems (CISS), 2016 Annual Conference on*. IEEE, 2016, pp. 181–186.
[4] P. Chen, L. Desmet, and C. Huygens, "A study on advanced persistent threats," in *IFIP International Conference on Communications and Multimedia Security*. Springer, 2014, pp. 63–72.
[5] B. Schneier, "Attack trees," *Dr. Dobb's journal*, vol. 24, no. 12, pp. 21–29, 1999.
[6] A. K. Sood and R. J. Enbody, "Targeted cyberattacks: a superset of advanced persistent threats," *IEEE security & privacy*, vol. 11, no. 1, pp. 54–61, 2013.
[7] O. S. V. D. (OSVDB), "Open source vulnerability database (osvdb)," 2012.
[8] P. Mell, K. Scarfone, and S. Romanosky, "Common vulnerability scoring system," *IEEE Security & Privacy*, vol. 4, no. 6, 2006.
[9] M. Motoyama, D. McCoy, K. Levchenko, S. Savage, and G. M. Voelker, "An analysis of underground forums," in *Proceedings of the 2011 ACM SIGCOMM conference on Internet measurement conference*. ACM, 2011, pp. 71–80.
[10] M. Lee and D. Lewis, "Clustering disparate attacks: mapping the activities of the advanced persistent threat," *Last accessed June*, vol. 26, 2013.
[11] K. Baumgartner and T. M. C. M. Golovkin, "The earliest naikon apt campaigns,kaspersky lab," 2015.
[12] "Kaspersky labs - global research & analysis team, carbanak apt-the great bank robbery."
[13] "The duqu 2.0," Jun. 2015.
[14] K. Baumgartner and M. Golovkin, "The naikon apt," https://securelist.com/analysis/publications/69953/the-naikon-apt/.
[15] "Kaspersky labs - global research & analysis team, equation group:questions and answers," Feb. 2015, available online.
[16] Cylance, "Operation cleaver," Dec. 2014, available online.
[17] R. I. Response, "Shell crew," Jan. 2014.
[18] K. L. G. R. A. Team, "The 'icefog' apt: A tale of cloak and three daggers," Sep. 2013.
[19] "The regin platform - nation-state ownage of gsm networks," Nov. 2014.
[20] GROUP-IB and FOX-IT, "Anunak: Apt against financial institutions," Dec. 2014.
[21] D. Aplerovitch, "Deep in thought: Chinese targeting of national security think tanks," Jul. 2014, http://blog.crowdstrike.com/deep-thought-chinesetargeting-national-security-think-tanks/.
[22] F. Ullah, M. Edwards, R. Ramdhany, R. Chitchyan, M. A. Babar, and A. Rashid, "Data exfiltration: A review of external attack vectors and countermeasures," *Journal of Network and Computer Applications*, 2018.
[23] 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 *Communications (ICC), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1–6.
[24] Z. Ferrer and M. C. Ferrer, "In-depth analysis of hydraq," *The face of cyberwar enemies unfolds. ca isbu-isi white paper*, vol. 37, 2010.
[25] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security & Privacy*, vol. 9, no. 3, pp. 49–51, 2011.
[26] N. Falliere, L. O. Murchu, and E. Chien, "W32. stuxnet dossier," *White paper, Symantec Corp., Security Response*, vol. 5, no. 6, p. 29, 2011.
[27] L.-X. Yang, P. Li, X. Yang, and Y. Y. Tang, "Security evaluation of the cyber networks under advanced persistent threats," *IEEE Access*, 2017.
[28] H. Yin, D. Song, M. Egele, C. Kruegel, and E. Kirda, "Panorama: capturing system-wide information flow for malware detection and analysis," in *Proceedings of the 14th ACM conference on Computer and communications security*. ACM, 2007, pp. 116–127.
[29] N. Virvilis and D. Gritzalis, "The big four-what we did wrong in advanced persistent threat detection?" in *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on*. IEEE, 2013, pp. 248–254.
[30] I. Korkin and I. Nesterow, "Acceleration of statistical detection of zero-day malware in the memory dump using cuda-enabled gpu hardware," *arXiv preprint arXiv:1606.04662*, 2016.
[31] Z. Xu, S. Ray, P. Subramanyan, and S. Malik, "Malware detection using machine learning based analysis of virtual memory access patterns," in *2017 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2017, pp. 169–174.
[32] C. Vaas and J. Happa, "Detecting disguised processes using application-behavior profiling," in *Technologies for Homeland Security (HST), 2017 IEEE International Symposium on*. IEEE, 2017, pp. 1–6.
[33] M. Marchetti, F. Pierazzi, M. Colajanni, and A. Guido, "Analysis of high volumes of network traffic for advanced persistent threat detection," *Computer Networks*, vol. 109, pp. 127–141, 2016.
[34] N. Villeneuve and J. Bennett, "Detecting apt activity with network traffic analysis," *Trend Micro Incorporated Research Paper*, 2012.
[35] A. Vance, "Flow based analysis of advanced persistent threats detecting targeted attacks in cloud computing," in *Problems of Infocommunications Science and Technology, 2014 First International Scientific-Practical Conference*. IEEE, 2014, pp. 173–176.
[36] P. Hu, H. Li, H. Fu, D. Cansever, and P. Mohapatra, "Dynamic defense strategy against advanced persistent threat with insiders," in *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 2015, pp. 747–755.
[37] A. Bohara, U. Thakore, and W. H. Sanders, "Intrusion detection in enterprise systems by combining and clustering diverse monitor data," in *Proceedings of the Symposium and Bootcamp on the Science of Security*. ACM, 2016, pp. 7–16.
[38] A. Shalaginov, K. Franke, and X. Huang, "Malware beaconing detection by mining large-scale dns logs for targeted attack identification," in *18th International Conference on Computational Intelligence in Security Information Systems. WASET*, 2016.

Table IX: 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
[3]	✓	✓					✓		
[4]	✓	✓	✓	✓					
[22]	✓							✓	
[99]	✓			✓					
[100]	✓	✓		✓		✓			
Our Survey	✓	✓	✓	✓	✓	✓	✓	✓	✓

[39] T.-F. Yen, A. Oprea, K. Onarlioglu, T. Leetham, W. Robertson, A. Juels, and E. Kirda, "Beehive: Large-scale log analysis for detecting suspicious activity in enterprise networks," in *Proceedings of the 29th Annual Computer Security Applications Conference*. ACM, 2013, pp. 199–208.

[40] P. Bhatt, E. T. Yano, and P. Gustavsson, "Towards a framework to detect multi-stage advanced persistent threats attacks," in *Service Oriented System Engineering (SOSE), 2014 IEEE 8th International Symposium on*. IEEE, 2014, pp. 390–395.

[41] W. Niu, X. Zhang, G. Yang, J. Zhu, and Z. Ren, "Identifying apt malware domain based on mobile dns logging," *Mathematical Problems in Engineering*, vol. 2017, 2017.

[42] V. Hodge and J. Austin, "A survey of outlier detection methodologies," *Artificial intelligence review*, vol. 22, no. 2, pp. 85–126, 2004.

[43] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM computing surveys (CSUR)*, vol. 41, no. 3, p. 15, 2009.

[44] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *Ieee communications surveys & tutorials*, vol. 16, no. 1, pp. 303–336, 2014.

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

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

[47] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *computers & security*, vol. 28, no. 1, pp. 18–28, 2009.

[48] S. Siddiqui, M. S. Khan, K. Ferens, and W. Kinsner, "Detecting advanced persistent threats using fractal dimension based machine learning classification," in *Proceedings of the 2016 ACM on International Workshop on Security And Privacy Analytics*. ACM, 2016, pp. 64–69.

[49] M. Parkour, "Contagio malware database," 2013.

[50] DARPA, "Darpa scalable network monitoring (snm) program traffic (11/03/2009 to 11/12/2009)," pp. –, 2012.

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

[52] B. C. Cappers and J. J. van Wijk, "Understanding the context of network traffic alerts," in *Visualization for Cyber Security (VizSec), 2016 IEEE Symposium on*. IEEE, 2016, pp. 1–8.

[53] X. Yuan, "Phd forum: Deep learning-based real-time malware detection with multi-stage analysis," in *Smart Computing (SMARTCOMP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–2.

[54] B. C. Cappers and J. J. van Wijk, "Snaps: Semantic network traffic analysis through projection and selection," in *Visualization for Cyber Security (VizSec), 2015 IEEE Symposium on*. IEEE, 2015, pp. 1–8.

[55] P. Dewan, A. Kashyap, and P. Kumaraguru, "Analyzing social and stylometric features to identify spear phishing emails," in *Electronic Crime Research (eCrime), 2014 APWG Symposium on*. IEEE, 2014, pp. 1–13.

[56] J.-S. Wu, Y.-J. Lee, T.-E. Wei, C.-H. Hsieh, and C.-M. Lai, "Chainspot: Mining service logs for cyber security threat detection," in *Trustcom/BigDataSE/I SPA, 2016 IEEE*. IEEE, 2016, pp. 1867–1874.

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

[58] 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 *Security Technology (ICCST), 2015 International Carnahan Conference on*. IEEE, 2015, pp. 287–292.

[59] A. Razaq, H. Tianfield, and P. Barrie, "A big data analytics based approach to anomaly detection," in *Big Data Computing Applications and Technologies (BDCAT), 2016 IEEE/ACM 3rd International Conference on*. IEEE, 2016, pp. 187–193.

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

[61] S. Smadi, N. Aslam, L. Zhang, R. Alasem, and M. Hossain, "Detection of phishing emails using data mining algorithms," in *Software, Knowledge, Information Management and Applications (SKIMA), 2015 9th International Conference on*. IEEE, 2015, pp. 1–8.

[62] Y. Mehmood, U. Habiba, M. A. Shibli, and R. Masood, "Intrusion detection system in cloud computing: Challenges and opportunities," in *Information Assurance (NCIA), 2013 2nd National Conference on*. IEEE, 2013, pp. 59–66.

[63] T. Zhang, Q. Liao, and L. Shi, "Bridging the gap of network management and anomaly detection through interactive visualization," in *Visualization Symposium (PacificVis), 2014 IEEE Pacific*. IEEE, 2014, pp. 253–257.

[64] R. S. S. Kumar, A. Wicker, and M. Swann, "Practical machine learning for cloud intrusion detection: Challenges and the way forward," *arXiv preprint arXiv:1709.07095*, 2017.

[65] M. Min, L. Xiao, C. Xie, M. Hajimirsadeghi, and N. B. Mandayam, "Defense against advanced persistent threats: a colonel blotto game approach," in *Communications (ICC), 2017 IEEE International Conference on*. IEEE, 2017, pp. 1–6.

[66] X. Yan and J. Zhang, "Early detection of cyber security threats using structured behavior modeling," *ACM Transactions on Information and System Security*, vol. 5, 2013.

[67] H.-K. Peng, P. Wu, J. Zhu, and J. Y. Zhang, "Helix: Unsupervised grammar induction for structured activity recognition," in *Data Mining (ICDM), 2011 IEEE 11th International Conference on*. IEEE, 2011, pp. 1194–1199.

[68] J. R. Johnson and E. A. Hogan, "A graph analytic metric for mitigating advanced persistent threat," in *Intelligence and Security Informatics (ISI), 2013 IEEE International Conference on*. IEEE, 2013, pp. 129–133.

[69] K. Ingols, R. Lippmann, and K. Piwowarski, "Practical attack graph generation for network defense," in *Computer Security Applications Conference, 2006. ACSAC'06. 22nd Annual*. IEEE, 2006, pp. 121–130.

[70] M. Albanese, S. Jajodia, and S. Noel, "Time-efficient and cost-effective network hardening using attack graphs," in *Dependable Systems and Networks (DSN), 2012 42nd Annual IEEE/IFIP International Conference on*. IEEE, 2012, pp. 1–12.

[71] S. Jha, O. Sheyner, and J. Wing, "Two formal analyses of attack graphs," in *Computer Security Foundations Workshop, 2002. Proceedings. 15th IEEE*. IEEE, 2002, pp. 49–63.

[72] X. Ou, W. F. Boyer, and M. A. McQueen, "A scalable approach to attack graph generation," in *Proceedings of the 13th ACM conference on Computer and communications security*. ACM, 2006, pp. 336–345.

[73] J. Lee, H. Lee, and H. P. In, "Scalable attack graph for risk assessment," in *Information Networking, 2009. ICOIN 2009. International Conference on*. IEEE, 2009, pp. 1–5.

[74] J. Homer, X. Ou, and M. A. McQueen, "From attack graphs to automated configuration management—an iterative approach," *Kansas State University Technical Report*, 2008.

[75] R. E. Sawilla and X. Ou, "Identifying critical attack assets in dependency attack graphs," in *European Symposium on Research in Computer Security*. Springer, 2008, pp. 18–34.

[76] H. Huang, S. Zhang, X. Ou, A. Prakash, and K. Sakallah, "Distilling critical attack graph surface iteratively through minimum-cost sat solving," in *Proceedings of the 27th Annual Computer Security Applications Conference*. ACM, 2011, pp. 31–40.

[77] B. M. Bowen, S. Hershkop, A. D. Keromytis, and S. J. Stolfo, "Baiting inside attackers using decoy documents." Springer.

[78] V. E. Urias, W. M. Stout, and H. W. Lin, "Gathering threat intelligence through computer network deception," in *Technologies for Homeland Security (HST), 2016 IEEE Symposium on*. IEEE, 2016, pp. 1–6.

[79] K. G. Anagnostakis, S. Sidiroglou, P. Akritidis, K. Xinidis, E. P. Markatos, and A. D. Keromytis, "Detecting targeted attacks using shadow honeypots." in *Usenix Security Symposium*, 2005.

[80] M. Crouse, B. Prosser, and E. W. Fulp, "Probabilistic performance analysis of moving target and deception reconnaissance defenses," in *Proceedings of the Second ACM Workshop on Moving Target Defense*. ACM, 2015, pp. 21–29.

[81] J. B. Hong and D. S. Kim, "Assessing the effectiveness of moving target defenses using security models," *IEEE Transactions on Dependable and Secure Computing*, vol. 13, no. 2, pp. 163–177, 2016.

[82] P. Kampanakis, H. Perros, and T. Beyene, "Sdn-based solutions for moving target defense network protection," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on a*. IEEE, 2014, pp. 1–6.

[83] S. Debroy, P. Calyam, M. Nguyen, A. Stage, and V. Georgiev, "Frequency-minimal moving target defense using software-defined networking," in *Computing, Networking and Communications (ICNC), 2016 International Conference on*. IEEE, 2016, pp. 1–6.

[84] J. H. Jafarian, E. Al-Shaer, and Q. Duan, "Openflow random host mutation: transparent moving target defense using software defined networking," in *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 2012, pp. 127–132.

[85] Z. Zhao, F. Liu, and D. Gong, "An sdn-based fingerprint hopping method to prevent fingerprinting attacks," *Security and Communication Networks*, vol. 2017, 2017.

[86] A. Chowdhary, S. Pisharody, A. Alshamrani, and D. Huang, "Dynamic game based security framework in sdn-enabled cloud networking environments," in *Proceedings of the ACM International Workshop on Security in Software Defined Networks & Network Function Virtualization*. ACM, 2017, pp. 53–58.

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

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

[89] 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.

[90] S. Neti, A. Somayaji, and M. E. Locasto, "Software diversity: Security, entropy and game theory."

[91] A. Clark, K. Sun, L. Bushnell, and R. Poovendran, "A game-theoretic approach to ip address randomization in decoy-based cyber defense," in *International Conference on Decision and Game Theory for Security*. Springer, 2015, pp. 3–21.

[92] A. Chowdhary, S. Pisharody, and D. Huang, "Sdn based scalable mtd solution in cloud network," in *Proceedings of the 2016 ACM Workshop on Moving Target Defense*. ACM, 2016, pp. 27–36.

[93] F. Skopik, G. Settanni, R. Fiedler, and I. Friedberg, "Semi-synthetic data set generation for security software evaluation," in *Privacy, Security and Trust (PST), 2014 Twelfth Annual International Conference on*. IEEE, 2014, pp. 156–163.

[94] T. Haq, J. Zhai, and V. K. Pidathala, "Advanced persistent threat (apt) detection center," Apr. 18 2017, uS Patent 9,628,507.

[95] D. Ucci, L. Aniello, and R. Baldoni, "Survey on the usage of machine learning techniques for malware analysis," *arXiv preprint arXiv:1710.08189*, 2017.

[96] V. Benjamin, W. Li, T. Holt, and H. Chen, "Exploring threats and vulnerabilities in hacker web: Forums, irc and carding shops," in *Intelligence and Security Informatics (ISI), 2015 IEEE International Conference on*. IEEE, 2015, pp. 85–90.

[97] E. Nunes, A. Diab, A. Gunn, E. Marin, V. Mishra, V. Paliath, J. Robertson, J. Shakarian, A. Thart, and P. Shakarian, "Darknet and deepnet mining for proactive cybersecurity threat intelligence," in *Intelligence and Security Informatics (ISI), 2016 IEEE Conference on*. IEEE, 2016, pp. 7–12.

[98] M. Almukaynizi, E. Nunes, K. Dharaiya, M. Senguttuvan, J. Shakarian, and P. Shakarian, "Proactive identification of exploits in the wild through vulnerability mentions online," in *Cyber Conflict (CyCon US), 2017 International Conference on*. IEEE, 2017, pp. 82–88.

[99] J. Vukalović and D. Delija, "Advanced persistent threats-detection and defense," in *Information and Communication Technology, Electronics and Microelectronics (MIPRO), 2015 38th International Convention on*. IEEE, 2015, pp. 1324–1330.

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

PLACE PHOTO HERE

Adel Alshamrani Biography text here.

Sowmya Myneni Biography text here.

Ankur Chowdhary Biography text here.

Dijiang Huang Biography text here.