J Supercomput (2019) 75:4543–4574
https://doi.org/10.1007/s11227-016-1850-4

A comprehensive study on APT attacks and countermeasures for future networks and communications: challenges and solutions

Saurabh Singh[1] · Pradip Kumar Sharma[1] · Seo Yeon Moon[1] · Daesung Moon[2] · Jong Hyuk Park[1]

Published online: 7 September 2016

Abstract Recently in the connected digital world, targeted attack has become one of the most serious threats to conventional computing systems. Advanced persistent threat (APT) is currently one of the most important threats considering the information security concept. APT persistently collects data from a specific target by exploiting vulnerabilities using diverse attack techniques. Many researchers have contributed to find approaches and solutions to fight against network intrusion and malicious software. However, only a few of these solutions are particularly focused on APT. In this paper, we introduce a structured study on semantic-aware work to find potential contributions that analyze and detect APT in details. We propose modeling phase that discusses the typical steps in APT attacks to collect the desired information by attackers. Our research explores social network and web infrastructure exploitation as well as communication protocols and much more for future networks and communications. The paper also includes some recent Zero-day attacks, use case scenarios

✉ Jong Hyuk Park
jhpark1@seoultech.ac.kr

Saurabh Singh
singh1989@seoultech.ac.kr

Pradip Kumar Sharma
pradip@seoultech.ac.kr

Seo Yeon Moon
moon.sy0621@seoultech.ac.kr

Daesung Moon
daesung@etri.re.kr

1 Department of Computer Science and Engineering, Seoul National University of Science and Technology (SeoulTech), Seoul, Korea

2 Network Security Research Team, Electronics and Telecommunications Research Institute, Daejeon, Korea

and cyber trends in southeastern countries. To overcome these challenges and attacks, we introduce a detailed comprehensive literature evaluation scheme that classifies and provides countermeasures of APT attack behavior. Furthermore, we discuss future research direction of APT defense framework of next-generation threat life cycle.

Keywords APT · Threat · Vulnerability · Exploitation · Zero-day attack

1 Introduction

In 2006, the United States Air Force (USAF) analysts came up with the term APT to facilitate discussion of intrusion activities with their unclear civilian counterparts [1]. Thus, the military teams could discuss the attack characteristics yet without revealing classified identities and explain the components of the terminology.

Advanced The attacker has the ability to bypass the detection and capability to gain and maintain access to well-protected networks and the confidential information that is contained within them. The hacker has usually an adaptive nature and is well resourced.

Persistent The persistent nature of the threat makes it difficult for the defender to protect against it. It is very difficult to prevent access to your system network, when the attacker has successfully gained access to your computer network; it is almost very difficult to remove.

Threat The hackers have to maintain the level of capability to gain access the sensitive information that is stored electronically.

APT was originally invented by a community involved in cyber-espionage to steal information for monetary gains [2]. Experienced computer exploitation adversaries' practice has a great number of attacks in the organizations repository to successfully penetrate even the most heavily shielded and secured networks to steal treasured company intellectual capital and government secrets. APTs target a particular enterprise unlike other usually found malware, which infects random of millions of boxes. The singular motive here is to gain access financial benefit by inducing damage to cyber infrastructure. How do you know if your organization is the target of an APT? If it is, how can you adequately categorize and incorporate this threat while maintaining the continuity of operations. APTs are the pinnacle of computer system threats and anticipate for them is just as important as identifying that you have become a target and anticipating them is as important as realizing that you have become a target [3–7].

Bodmer et al. [8] define the APT objective—the end goal of threat, timeliness—time spent poking into and accessing your system; risk tolerance—the threat will go to remain undetected; resources—the level of knowledge and tools weigh to the event; attack discovery points—the counts of points where the event originated; numbers involved in the attack—how many internal and external systems in the network were involved in the event.

The aim of this paper is to provide a systematic review of proposed approaches and solutions that facilitate the target attack detection. The paper provides a brief introduction of APT phases and its behavior pattern. The paper also includes the different types of APT and Zero-day attacks including cyber trends in the world. A

special focus is on semantic aware approaches that are helpful to find out the targeted attacks and defeating traditional signature-based detection method. In the complicated network of cyber-attack, the defenders need to consider the network-based, host-based and another hybrid monitoring tool to capture the suspicious activity that can later be analyzed. Finally, the paper includes existing models and technologies that are well categorized by different parameter. The discussion and future generation of APT threat life cycle effectively help to understand the detection process and analysis in future.

The structure of the paper is as follows; Sect. 2 introduces the APT phases and its behavior. Section 3 provides different APT attacks, Zero-day attacks, use case based and cyber trends in the world. In Sect. 4, a detailed study with the specific background is provided and reviews the paper related with advanced threats. Section 5 summarizes the paper with discussion and proposes the future generation threat life cycle to better understand the behavior of targeted attack, which further helps to detect and analyze the targeted attack. We conclude our research in Sect. 6.

2 APT taxonomy and pattern

APT modeling phase is composed of reconnaissance, breach, infiltration, exfiltration, and stealth persistence as shown in Fig. 1.

2.1 APT modeling phase

Reconnaissance In this step, attackers gather information about the target organization resources, employees and relationships with other entities that can be leveraged to reach the target [9]. The attacker should consider about the parameter of defense system and access information while scanning the network. Then, they build employees

Fig. 1 APT modeling phase

profiles using social networks such as 'LinkedIn, Facebook, etc.' and set up the malicious service delivery via various channel to gather information of targeted system.

Breach In targeted attacks, adversary typically enters the network of the target organization through different methods using stolen authentication data, SQL injections, and malware for targeted attacks. This step includes the exploitation of vulnerabilities present in the network system. The intruder controls the infected host using command and control service remotely. To enter the target system, attackers sometimes use social engineering and phishing methods to motivate an employee to click a link or view and run an attached file by carelessness or mistake. They use Zero-day vulnerabilities as well [9]. In the Stuxnet case, four Zero-day vulnerabilities were used simultaneously [10,11].

Infiltration An intruder who enters the target system collects knowledge on the system and automatically searches for confidential data. Attackers move horizontally in the network and identify the servers storing the sensitive information, users having the legitimate access privileges and set up the strategy to collect and export the targeted information. Examples of information collection targets include unprotected important data, software, networks, hardware, exposed confidential documents, and additional resources [12,13]. To do this, additional attack tools are downloaded, and systems and data obtained including network configuration, user IDs and passwords are researched and analyzed.

Exfiltration In the APT attacks, unauthorized attackers take control of the target system and leak a range of confidential data, including intellectual property rights. They also damage software and hardware systems [14,15]. Secret data are transferred to attackers through web mail, as a packet, or as a compressed file. It is unauthorized transfer of sensitive information from a target's network to an external location which the threat actor controls. After exploring the data of interest, the APT usually gathers the data into an archive and then compresses and encrypts the archive. This enables them to obscure the content of the archive from deep packet analysis and data loss prevention techniques.

Stealth persistence APT attacks don't take place overnight; instead first it plan and spans a period of time. The attacker's malicious code goes through a rigorous round of proving such that it is not detectable and the existence of the attacker is impossible to tap by the organization. Such an attacker would require an immense amount of patience to get the fruitful result of his efforts. Daily based audits and monitoring to find out a correlated burst of actions can help enterprise to detect the APT and mitigate them on time before any data breach.

2.2 Behavioral pattern

System users are infected by the website Malware can be sneaking into your computer system when you are using a malicious website. The website contains malicious files which automatically downloaded by clicking somewhere on the website. At this stage, users might be completely unaware that they are infected and the malware has already started destroying the system. Spyware, Viruses ransomware and another kind of malware attacks have become so sophisticated that they could bring out the chaotic

situation on your systems without showing any symptoms of its presence until it is too late. Through this infection, confidential information is stolen and a trapdoor is built. Using confidential information, the attacker can authenticate the nodes in the system to expand the attack infrastructure.

E-Mails focusing a specific target In this kind of APT behavior, an attacker sends a malicious e-mail which contains a virus to a specific target. Once the receiver opens the e-mail, the computer of the receiver will be directly infected with malware and a backdoor connecting to the Command and Control (C&C) server will be built. Finally, the data saved in the system will be continuously leaked and update the information of victim to the attacker. By sending the malicious e-mail to target in a specific organization is called an attack on the specific target [2].

Websites are misused as websites where malware passes some authenticated websites are attacked by attackers and make them websites to pass the malware [16]. As a result, using authenticated website attackers are able to spread or move their malware in the system. It is a very critical situation for the users because they become assailants as well as victims.

Using media system virus infection Auxiliary memory mediums of viruses, like a USB drive, that infiltrate into a computer system via different paths. When a virus enters a system, the infection will extend to the system network and a backdoor connecting to the C&C server will be built. Accordingly, the assailant will get control of the system and server and also assorted data and information in the system.

Integrated DDoS In coordinated DDoS attack, various infections isolate their parts relying upon their capacities and work efficiently. Incorporated DDoS attacks cause over-burden on a web server or a circuit being used, and can send a lot of spam. When this sort of attacks happens, it is exceptionally hard to dissect data on the assault. On the off chance that such an assault is done in various ways, it will be an extremely troublesome risk to handle [17].

Attacks on management systems One of the best cases of an attack on an organization system is Stuxnet, which was an attack on a nuclear power plant in Iran. This attack drew consideration in the light of the fact that customized attack program work in the objective offices was utilized. The functional attributes of this attack were to exploit information trade between the open system required for work and inside administration framework with a USB.

3 APT attacks

This chapter deals with different APT attack and discusses exploitation of targets.

3.1 Attack and exploit the targets

Once complete to collect information about the target, attackers move on to building a threat model. After collecting information, they analyze the gathered information to develop a profile of the target and his or her environment. They even construct a

replica of the target system so that they can test various penetrations without revealing that an attack is forthcoming.

Attack modeling provides significant information about the weaknesses in an organization's network and employees that could be exploitable. It is a critical step in successfully executing targeted attacks.

The final phase is self-explanatory attackers launch the attacks, building on the information gathered and the profile developed. In general, the goal is to load malware onto a target's machine and use that platform to extract information. Targeted attacks can vary significantly in how they are executed, but they have some common patterns.

Drive-by downloads and spear phishing Attackers use drive-by download attacks to get the target to download malware from the Internet [18]. To do this, the user is daunted to visit a compromised website that hosts a hidden Iframe that redirects to the user's browser to yet another malicious domain that runs a browser exploit pack (BEP). It exploits vulnerabilities in the user's browser to download malware directly into the system [19]. Spear phishing is the primary means of directing a targeted user to a drive-by download site. It simply targeted phishing—personal and business information in an email to convince a user to visit a compromised website, which typically is done via an embedded link in the message. Customized spear-phishing attacks can involve obfuscation to aid in bypassing the automated defenses. If spear phishing targets a big fish, such as an executive, it is sometimes called "whaling". Botnets provide a handy mechanism for launching phishing anonymously, especially when targeting a group of users [20]. However, the built-in anonymity of botnets can also be useful when targeting individuals.

The process to launch this attack is as follows: Initially, attackers begin by collecting email addresses to initiate the spear phishing [21]. These addresses may be publicly available, but high-value targets can have private email addresses that will be more susceptible to phishing. An affluence of online outlets offers them for a price. If attackers have time and resources, the data mining of raw, bulk email dumps can yield useful addresses that might not have shown up through another kind of approaches. The attacker also searches the online resources and websites to find targeted user email addresses. Once the desired email addresses are founded, the attacker initiates an automated process of sending an email with malicious attachments.

The technique of sending the malicious email attachments persists as an effective attack vector. High-value organizations have developed the software to verify email attachments, but file formats such as PDF XLS, PDF or DOC can obtain through with embedded malicious code.

One advantage of using email is that it usually slips past peripheral security devices such as firewalls, malware protection system and intrusion detection systems. In this case, security depends on later attachment checking. Once the user inside the organization opens the email, many levels of security have already been bypassed. The malicious code now attacks vulnerable software in the system to expand the exploitation—it can even download more malicious content from remote parties. The idea is to slip something small and seemingly innocuous through the defenses and then upload more infectious code.

In targeted attacks, the exploitable code is usually designed to download a Remote Administration Toolkit that allows the attacker to manage the exploited system remotely. These toolkits are sophisticated software with a variety of integral tools to manage systems across an intranet. A single compromised host might be infecting the other machines on the network, as internal defenses are often weaker than external ones. Now, the attacker is inside the fortress. An excellent example of this type of attack was performed against RSA: attackers used a malicious XLS file embedded with an adobe flash exploit [22].

Exploiting web infrastructure Web application security flaws play an important role in targeted attacks. Two popular techniques, SQLI and XSS, have been used to conduct mass online attacks in which attackers exploit the exact vulnerability of servers across the Internet. Specifically, attackers exploit SQLI vulnerabilities by extracting database information and use the subsequent information to conduct additional attacks [23]. Generally, attackers combine SQLI and XSS in a hybrid attacks known as SQLXSSI, which updates a database of vulnerable websites with malicious Iframes via SQLI. When a user unknowingly visits a vulnerable website, the content is retrieved from the database, which consists of Iframes pointing to a malicious domain serving malware. In another scenario, an SQL injection attack can extract the password of domain, letting malware be installed directly.

One example is Lizamoon mass SQLI attack, in which an attacker targeted Microsoft servers running ASP.NET and exploited SQL injections using search engines to inject malicious code (Iframes pointing to malware) in the websites. Visiting users to those infected websites were assisted with malware [24]. MySQL.com, Goal.com, and content delivery networks such as DoubleClick have all been exploited recently to serve malware [25].

Exploiting communication protocols Attackers exploit several communication protocols over the Internet to bypass the normal flow of operations during an attack. The attackers can compromise SMTP servers configured as open broadcast and used them to spread the spear-phishing emails. Insecure HTTP and FTP servers might be used as storage warehouses to host malicious programs. Many Attackers can exploit the DNS protocol to redirect the legitimate traffic to a malicious site by changing DNS entries. Malware on a system can tweak the DNS entries in the host configuration file system or perform DLL injection methodology to redirect a browser to different domains [26]. Attackers have executed DNS cache-poisoning attacks, in which a server-side cache is filled with rogue DNS entries that can redirect the user's browser [27,28]. Of course, some of these activities can impact much more people than the targeted individual or group, which can increase the probability of detection.

Online social network exploitation The exponential growth of online social networks has provided an ample source of personal data and also has opportunities for social engineering. In social networks system, users connect to each other and share their information [29]. In the perception of targeted attacks, they provide to the attackers with an opportunity to exploit trust among friends. For example, a suggested link from a friend is more likely to be opened. Broad range attacks on social networks signify the potential with respect to targeted attacks.

Exploiting co-location services Many different services can exist in a single location and become more vulnerable. Their numbers are growing, so they are being exploited more frequently that can be useful in targeted attacks.

Virtual hosting, the provision of serverSelecting a Web hosting service, is beneficial from a business perspective, but when an attacker compromises one vulnerable website, it might be possible that he or she can take full control of the entire hosting server. Keeping such a server provides multiple places to host malware. Attackers can use two approaches to exploit target servers using virtual hosting to gain access to them. One, an attacker can manually write scripts to inject malicious Iframes to infect all hosts runs on the server. In another approach, an attacker can install a remote administration shell such as C-99 by compromising vulnerable website.

The cloud provides another platform for hosting malware. If infected, targeted users could compromise the cloud services of thousands of customers. IsecLab's security analysis of Amazon's cloud, Amazon Web Services (AWS), showed exactly this, by highlighting the state of AWS insecurity with respect to security vulnerabilities and exploitation [30].

Rogue Wi-Fi services and open or weak wireless networks provide yet another attack surface. Weaknesses here allow for information gathering or hosting of malware for drive-by downloads. For example, the soft AP/virtual Wi-Fi functionality in Windows 7 can be turned into a rogue Wi-Fi access point, which is an unauthorized network capable of communicating with the hosts in a network without explicit permission from the administrator. These soft APs are usually hidden because of the Windows Port Address Translation feature, which allows several networks to run behind a single IP address. Consequently, stealthy infections can be initiated using Peer-to-Peer (P2P) protocols.

Bluetooth services may be exploited or subverted to gather the information or allow access for hosting malware. In 2004, Cabir was the first proof of concept that demonstrated the practicality of Bluetooth malware [31].

Finally, the instant messaging and online chatting are some other mediums to spread the malware on the internet. As with social networks, this approach can exploit the trust or faith of friends, family and colleagues relationship to increase the chance of users clicking malicious links.

Physical attacks Hardware devices provide yet another attack surface. USB sticks are ubiquitous and frequently shared among individuals—yet another easily crossed trust boundary. Malware can copy itself onto USB sticks that spread malware whenever the USB stick is used in another system. This technique is especially useful to infect the machines that are not connected to the Internet. Shared memory devices such as DVDs, CDs, and memory cards can be carriers as well. Recently, teensy human-interface devices were used to make physical attacks by user-assisted attackers to execute arbitrary programs using a USB stick on target machines, including smartphones [32]. By design, a tiny device can emulate itself as a keyboard or a mouse. Once connected with the CPU, it can acquire keystroke information and execute payloads.

Recent researches have focused on hardware preloaded with backdoors [33]. A backdoor provides a window for installing malware. An obvious advantage of preloaded hardware is that it bypasses all Internet security because it is embedded in the

hardware and moves into an environment as an inherent component of the machine. Hardware-based backdoors have the capability to access the kernel and use a direct memory access engine.

3.2 Zero-day attacks

A Zero-day attack exploits an anonymous computer which has software or hardware security vulnerabilities. Attackers are exploited Zero-day vulnerabilities through different vectors [34].

The term 'Zero-day' is derived from the age of the exploit that happens before or on the first (or 'zeroth') day of a developer's knowledge of the exploit or bug. This meaning that there is no known information on security fix because developers are unaware of the vulnerability or threat. A Zero-day threat is also known as a day-zero attacks or zero-hour attacks [35].

There are many definitions for Zero-day attacks with little difference between them. Some professionals define Zero-day attacks that attack some vulnerability which has not been patched while the others say as the attack that takes advantage of the security vulnerability on the same date when the vulnerabilities become publically. Table 1 illustrates some recently happen Zero-day attacks [35,36].

It can generate some complicated problem before administrator realizes something going wrong. There is no patch available so it is very difficult to tackle these types of attacks even when developers of the company or an organization are aware of the issue. Prevention is the only possible way to deal with such kind of attacks for any type of networks. If the network administrator knows that how many such attacks can be possible, then he made some changes in his administrator rights [9,37,38].

3.3 Use cases of APT attacks

Stuxnet The objective of Stuxnet is a system destruction of a nuclear power plant [39]. Stuxnet is a worm and as such it has self-replicating capabilities [40]. Additionally, it is elaborately crafted to avoid detection from the operating system. So, it is very difficult to detect and provide an initial response. Stuxnet succeeded in remotely controlling over the supervisory control and data acquisition (SCADA), which manages the infrastructure within the plant [41]. There are many assumptions about the intrusion methods. However, most of the opinions point the cause of APT attacks. As shown in Fig. 2, Stuxnet mainly exploits vulnerabilities such as Windows shell LNK vulnerability, Windows Server Service vulnerability, Windows print spooler vulnerability, and Shared Network Service Vulnerability. In addition to exploiting those vulnerabilities, Stuxnet can be attached to a removable storage device (External hard drive or USB). Therefore, Stuxnet can be used to perform an APT attacks in systems which are separated from the network [42].

Spear phishing Spear phishing is a combination of 'Spear' and 'Phishing', which means that the attack is based on tricking the user. Firstly, the attacker sends malicious emails disguised as trusted information to related members. The idea is to steal the confidential information of individuals or organizations also to secretly spy on them.

Table 1 Some list of Zero-day attack in 2016 (January to June) [35,36]

Zero-day initiative	Common vulnerabilities and exposures	Description
ZDI-15-665	CVE: CVE-2015-8823	Apple OS X libATSServer heap-based buffer overflow remote code execution vulnerability
ZDI-16-360	CVE: CVE-2016-1797	Apple OS X fontd sandbox escape vulnerability
ZDI-16-359	CVE: CVE-2016-1094	Adobe Reader DC FlateDecode use-after-free remote code execution vulnerability
ZDI-16-355	CVE: CVE-2016-0186	Microsoft Edge JavaScript unshift method uninitialized memory remote code execution vulnerability
ZDI-16-352	CVE: CVE-2016-1859	Apple Safari GraphicsContext use-after-free remote code execution vulnerability
ZDI-16-231	CVE: CVE-2016-0159	Microsoft Internet Explorer CTableLayout AddRow out-of-bounds write remote code execution vulnerability
ZDI-16-229	CVE: CVE-2015-6065	Microsoft Internet Explorer CAttrValue double-free remote code execution vulnerability
ZDI-16-210	CVE: CVE-2016-0226	IBM Informix portmap service privilege escalation vulnerability
ZDI-16-199	CVE: CVE-2016-1961	Mozilla Firefox nsHTMLDocument setbody use-after-free remote code execution vulnerability
ZDI-16-197	CVE: CVE-2016-1645	Google Chrome Pdfium JPEG2000 out-of-bounds write remote code execution vulnerability
ZDI-16-190	CVE: CVE-2016-1008	Adobe Acrobat Pro DC DLL planting remote code execution vulnerability
ZDI-16-186	CVE: CVE-2016-0113	Microsoft Internet Explorer CTravelEntry use-after-free remote code execution vulnerability
ZDI-16-165	CVE: CVE-2016-0060	Microsoft Edge text node type confusion remote code execution vulnerability
ZDI-16-164	CVE: CVE-2016-2396	Dell SonicWALL GMS virtual appliance multiple remote code execution vulnerabilities
ZDI-16-161	CVE: CVE-2016-0973	Adobe Flash URLRequest use-after-free remote code execution vulnerability
ZDI-16-126	CVE: CVE-2016-0855	Advantech WebAccess dashboard viewer openwidget directory traversal information disclosure vulnerability
ZDI-16-364	CVE: CVE-2016-4360	Hewlett Packard Enterprise LoadRunner virtual table server import_csv denial of service vulnerability
ZDI-16-347	CVE: CVE-2016-1820	Apple OS X IOAudioFamily buffer overflow privilege escalation vulnerability
ZDI-16-335	CVE: CVE-2016-4496	Panasonic FPWIN Pro ReleaseBuffer Integer overflow out-of-bounds write remote code execution vulnerability
ZDI-16-329	CVE: CVE-2016-1095	Adobe Reader DC JPEG2000 out-of-bounds read information disclosure vulnerability
ZDI-16-248	CVE: CVE-2016-4351	Trend micro mail encryption gateway SQL injection remote code execution vulnerability

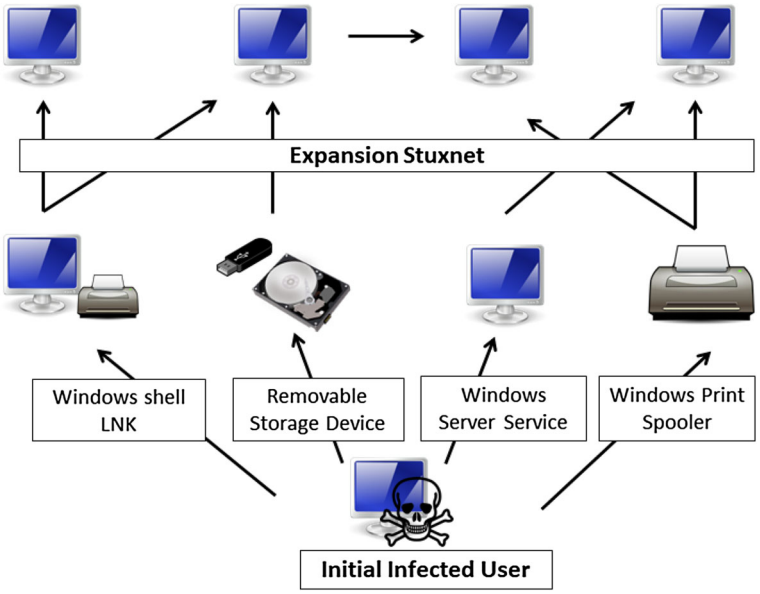

Fig. 2 Attack scenarios and structures in stuxnet

Secondly, the message content tricks the recipient into downloading the attached file. As a result, the recipient is infected due to the downloaded file. After infection, the attacker attempts to remotely control and steal the data [43]. Spear phishing E-mail includes a normal file and a malicious file with a double extension. If the malicious file with the double extension is executed, this is creating and executing in the normal document file path. At the same time, the normal document file running secretly installs an additional malicious file called "conhost.exe" into a temporary path where the user may fail to recognize it. This malicious code waits for additional commands from a remote address. Lastly, the attacker can remotely control or provide additional commands to steal information or install additional malware variants [44].

Duqu On the other hand, Duqu only has the ability to collect information. This avoids the monitor using the digital signature of the C-Media to look as legitimate [45]. This object is the information gathering stage for the attacks.

This Zero-day vulnerability attacks by attaching itself to an MS document file in order to hide, then it is delivered via e-mail. It usually infects after the user views the Word document which contains a TTF font file, where Duqu is hidden. Information about this vulnerability is registered as CVE-2011-3402. Vulnerability in MS Windows Components has been found that can allow running code on the vulnerable system for the attack. Duqu gathers information on the process, network, file, input key, other logs, etc. The collected information is then stored and encrypted with ~DQ (num).tmp in the log file. When exchanging information to the remote address, it runs the download and upload that the data to a JPG file [46].

Watering Hole Watering Hole originally comes from the ambush of a lion near a puddle of water. In the cyber-attack context, it means waiting for the user to take the trap.

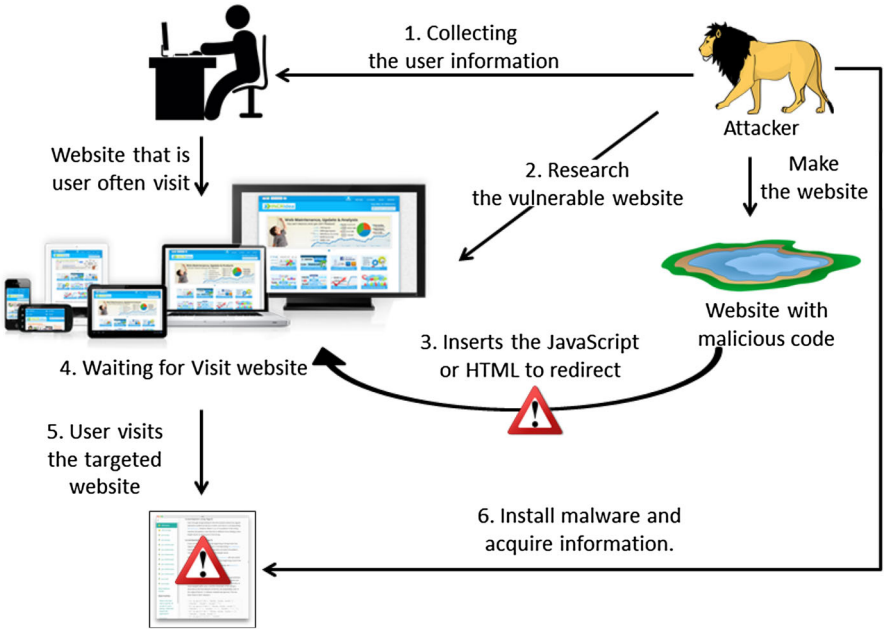

Fig. 3 Watering hole scenario and architecture

This attack uses the APT process. First, the attacker collects user information which is determined as the target. Then, it finds out the websites that the user frequently visits. Second, the attacker needs to find vulnerability on one of the frequently visited websites and hack it. After that, the attacker inserts JavaScript or HTML to the website to redirect to a particular website which is controlled by the attacker and contains the Zero-day vulnerability. Finally, the attacker only waits for the target to come to the Watering Hole (prepared website). Once the attack is successful, the attacker elopes and closes the used website to avoid tracking. This attack using Zero-day vulnerability is not considered novel because the method is similar to the one used by Stuxnet and Duqu. However, Watering hole is in fact a more advanced and intelligent targeted attack. As such, it poses a higher risk. Besides, this attack is the form 'hit and run' APT attack to avoid the risk of detection and tracking [47]. Figure 3 illustrates Watering Hole.

3.4 Cyber trends in Southeast Asia

Southeast Asia faces various unique challenges regarding cyber security. The region's extraordinary pace of economic development and growing military expenditures constitute the two major reasons why APT actors target governments and businesses. APT groups are looking to obtain intelligence to provide their sponsoring government with diplomats, military, and economic advantages across the negotiating table. Many Countries in this region continuously are facing the risk that persistent territorial dis-

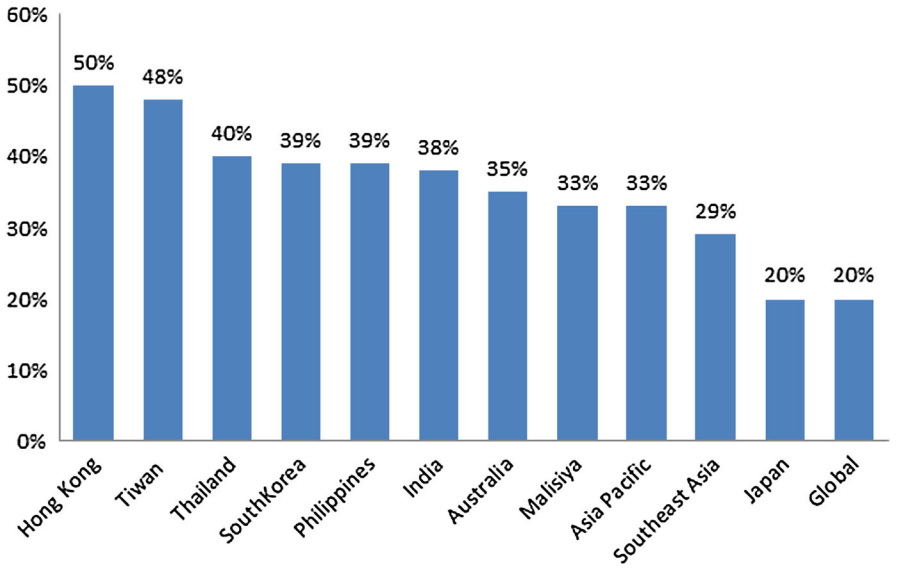

Fig. 4 Cyber trends in Southeast Asia

putes, specifically across the South China Sea, will expand into the cyber operations. China, Brunei, Philippines, Taiwan, Vietnam, and Malaysia all contest territory in the area. These disagreements have cyber trends in Southeastern Asia lingered for decades in some cases. Rival governments generally employ the APT groups to conduct the cyber espionage to obtain valuable political and military intelligence. The fire eye report routinely observes the APT groups stealing information that deals with South China Sea disputes and their economic effects from the networks of governments and companies involved [48,49].

As shown Fig. 4, the report says during the first half of 2015, about 29 % of their customers in Southeast Asia have detected malware associated with APT groups. It remains higher than our global average of about 20 %. Philippines and Thailand are among those countries in Southeast Asia most frequently targeted with malware associated with APT groups [50,51].

4 Classification of APT attack behavior and countermeasures

APT attacks are a form of the internal system intrusion. Firstly, malware infects particular hosts using USB or network vulnerabilities. Secondly, malicious code is downloaded and it is spread through the contamination server. And the next step is to discover the target system through the spreading of malware. Lastly, the vital information of target system is leaked out through malware. The APT attacks can be classified as external attacks, physical attacks and structural attacks. It may be summarized as follows, the external attacks makes use of another program except target like, virus, Trojan horse, and worm attacks and so on [52,53]. The physical attacks use weak points

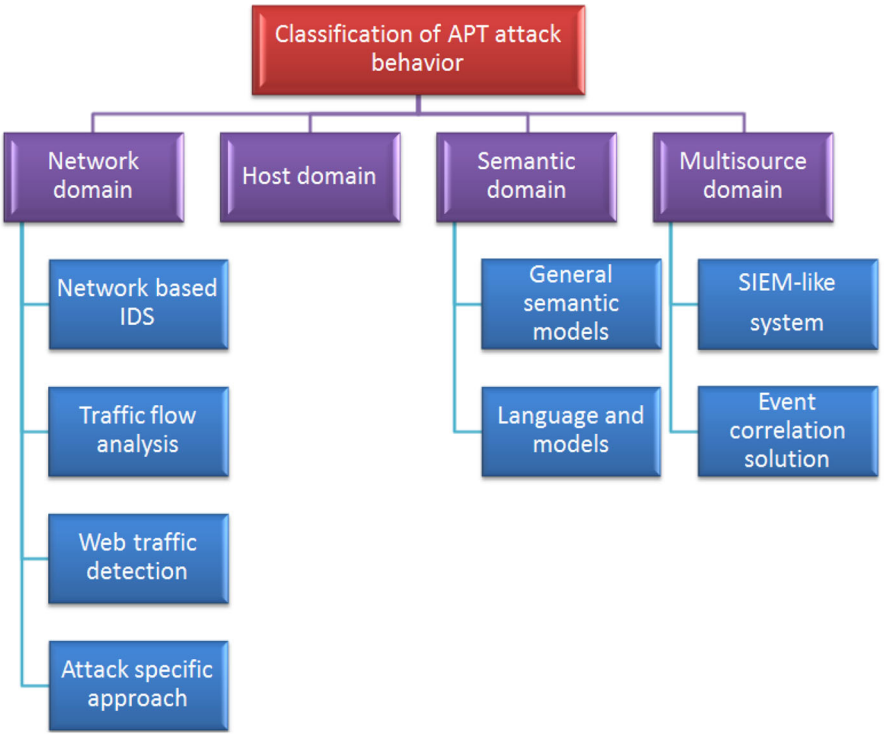

Fig. 5 Classification of APT attacks behavior

of source code such as buffer overflow and SQL injection. And the structural attacks exploit the vulnerability of the systems design and architecture such as authentication procedures, weak points of modularization, and protocol [54,55]. In this survey, APT attacks behavior and their countermeasures are categorized into network-based, host-based, purely semantic and multi-source that cannot be imputed to a specific domain. Figure 5 shows an overview of the full classification used in below subsections.

4.1 Network domain

Network-based techniques mainly include standard network-based intrusion detection systems (NIDS), traffic flow analysis solutions, specific attack detection systems as well as detection models for malicious web traffic.

In NIDS, network traffic analysis methods come in two distinct schemes [56]; Traffic flow-based detection methods and packet inspectors that examine the payload of packets. Instead of individual packets, it focuses on communication patterns. Such patterns usually include destination and source IP addresses, port numbers, transmission duration and timestamps, as well as the number of packets and amount of data sent. Flow monitoring methods like compatible IOS routers typically export IPFIX [57] or Netflow [58] records and passes on them to a central node for examination.

Detailed packet scrutiny requires significant processing power and may cause overload on even the most powerful networks. Because of this reason, many solutions focus mainly on the information in header and analyze only specific packets such as HTTP requests and responses appropriate for malicious web traffic detection. Flow-based methods are usually quicker but ignore packet payloads. Additionally, they are dependent on external analysis methods that consolidate and interpreted the collected data. Both methods may use anomaly-based and pattern-based detection. Substantial amounts of data are sent throughout the out-of-hours period; on the other hand, could comprise a data leakage anomaly.

A rising number of data providers use flow-based detection method to supplement traditional packet inspection. It is often applied to identify DoS attacks and other suspicious exchanging information patterns. On the other hand, packet scanning focuses at identifying potentially malicious payload data.

Numerous network-based methods use openly available datasets to experiment the potentiality of an NIDS. Examples include the KDD [59] and 2000 DARPA set [60]. On the payload side, the Client engine and ADMmutate [61] are often used to produce shellcode.

4.1.1 Network-based intrusion detection systems

General NIDS are presented in a huge number of papers. Some of those include a semantic module. For example, Abdoli and Kahani [62] illustrate a distributed IDS that can excerpt semantic relations between attacks via an ontology based on the Semantic Web [63], an accumulation of W3C formats for data exchange. Alike solutions [64,65], they make use of SPARQL [66] for querying and Protégé [67] for ontology design. Their system uses Java-based JENA [68] agents to gather ports, IPs, protocols, and link status information.

Several papers are inspired by the static malware detection analysis method proposed by Christodorescu et al. [69]. In research [70], the authors illustrate two sliding-window-based techniques exercised to automatically generate malware signatures that are pre-defined patterns, a fixed-size system and a more pliable variable length system which stops at certain.

Hirono et al. [71] presented another, more architecture-centered technique. They proposed a distributed IDS that makes use of a transparent proxy able to examine internal network traffic in a confined environment. The system mainly applies signature-based detection to identify spamming, malware propagation, or DoS attacks. Suspicious binaries are robotically extracted and send out to a dynamic malware analysis sandbox. Snort [75] integrated by an off-the-shelf virus scanner can be used to make the decision whether a sample is suspicious or not. In the same way, Andersson [73] exercises the Snort IDS for primary decision-making. In [71], Hirono et al. do not try to identify shellcode but rather target on the taking out and analysis process of the flagged binaries.

Association of intrusion events is an essential part of most multi-agent IDS schemes. Chien et al. [74] brought in a primitive-attacks (PA)-based correlation framework capable of identifying multi-stage attacks. IDS alerts from a variety of tools such as Snort [75] are gathered, stored and evaluated. Through time window association

complemented by IP and port matching, the method is able to discover such as network scans and DoS attacks.

Zhu and Ghorbani [76] presented an alert association technique that also considers attacker approaches. Their categorization includes a neural network as well as a support-vector machine (SVM) [72] scheme. Their scheme suggests a connection of alerts: For example, corollary of one event is mapped to the necessities of another to build scenarios that consider both association probability and strength. In addition to matching ports and IP similarities, the frequency of alerts is evaluated.

Recently, AlEroud and Karabatis [77,78] presented a layered attack detection scheme that makes use of context and semantics through a semantic network describing relations between attacks. Their technique applies conditional entropy theory (CET). The uncertainty of one event given another event to generate attacks in context profile that filter out non-relevant events [79].

4.1.2 Traffic flow analysis solutions

Flow-based techniques are gradually used more to detect network attacks. Sperotto et al. [56] presented a fine overview of traffic flow IDS schemes. Over the years, numerous different resolutions have been proposed.

Münz and Carle [80] presented TOPAS, a packet and traffic flow analysis scheme compatible with IPFIX and Cisco NetFlow. TOPAS offers a framework for user-defined detection components operating in real time. The data exercised are netflow specific in nature: destination and source ports/IP addresses as well as protocols are considered. The scheme's detection algorithm includes threshold-based detection through pre-defined values that necessitate to be exceeded, principal component classifiers (PCC) to identify anomalies in multivariate time series, outlier detection via the assessment of a sample to normal behavior, previously learned, and rule learning through a categorization extracted from these "evil" and "good" training sets.

Vance's work [81] is one of the few techniques that concentrate directly on APT; He illustrates a flow-based monitoring method that uses statistical analysis of gathered network traffic data to identify anomalies. He makes use of change detection via sketch-based measurement [82] to recognize flows that hint at control and command traffic, exfiltration, or data mining activities. Timing, packet size, and volume are of main interest; the individual baseline and subsequent analysis consider traffic throughput and packet, the number of concurrent flows, RST and TCP/IP SYN packets, current time and the flow duration.

Flow-based intrusion detection is depicted in [83]. The authors' scheme uses probabilistic semantic link networks (SLN) substitutable to graphs utilizing similarity values to illustrate node connections. The target's IP address, duration and time of communication, as well as different other features such as flags and protocols, are mined from network traffic and the related flows. Similar characteristics are translated into link weight among the respective nodes of the graph.

4.1.3 Web traffic detection and analysis systems

Web service attacks are a frequent part of the reconnaissance stages of a targeted attacks or intention to publicly reveal sensitive information. Business-critical communications can be accessed via exposed web portals that permit privileged users to configure or monitor back end systems. Because of their high visibleness, web servers are also regular targets of defacement attacks. Because of these reasons, numerous security solutions concentrate on the detection of malicious HTTP traffis.

Razzaq et al. [84] illustrated a technique that is able to detect and categorize web application attacks. Threats are specified through semantic rules that set up the context: both attacks effect and common application attributes such as protocol use are assessed. The proposed system analyzes the user part of HTTP request to detect authentication bypass attacks, probing, cookie stealing attacks, DoS, and so on. The authors proposed an ontological model [85] by making a use of a description logic based on OWL [86,87] and authorized through OntoClean [88]. The inference policies were implemented using the Apache JENA framework [68].

Previous work of Razzaq et al. [89] explained another interesting semantic technique. The authors brought up an application-level ID using a Bayesian filter to discover malicious scripts in HTTP protocol traffic. Parameters, URL, cookies, etc. are considered. This 'spam filter' allows values to particular keywords and produces a score. Those attacks are matched with attack description stored in an external database.

Another semantic intrusion detection scheme is proposed by Sangeetha and Vaidehi [90]. It classifies malicious behavior as rules that include frequency of occurrence, source, and parts of the HTTP packet content. Rules are played as BNF grammar [91]. In [92], Fuzzy Cognitive Mapping is applied for attacks prediction. The authors' interpreter and traffic sniffer scheme were enforced as part of a client–server infrastructure.

SpuNge [93] is a technique explicitly designed to identify targeted attacks via behavior clustering and industry/location association. The analysis target lies on URLs—its framework is capable of identifying machines that are portion of the same attacks. This is attained through string similarity measurement and hierarchical clustering utilizing the Levenshtein distance [94].

Thakar et al. [95] also aimed at the analysis and mining of traffic patterns. Their work rotates around the extraction of signatures that can afterward be used by an IDS. In [96], the authors log SOAP traffic and take out information such as client IP addresses, ports, identifiers, and certain strings in HTTP packets (requests and response). The data are then clustered via an SVM-based classifier [72].

Zarras et al. [97] brought in BotHound a detection technique for malware communication over HTTP. The scheme automatically builds models for benign and malicious petitions and categorizing new traffic in real time. The prime goal is to find out both traffic and C2 communicating with the help of suspicious HTTP templates and sequences of HTTP headers like header chains and that include content data such as ports, IP addresses, transported file types, and more.

4.1.4 Attacks-specific approaches

There are numerous network attacks detection techniques that pay attention to a specific type of threat. Because of their eminence, DoS/DDoS attacks are of special interest. In [98], one solution is proposed for such attacks by Gamer et al. Their proposed attacks detection scheme is located on routers and centering on DDoS attacks and malware spread. Network traffic is sampled and then classified into several stages: at the granularity level, Gamer's scheme only considers the overall number of packets and tries to discover anomalous changes in volume. Level two differentiates DDoS from worm propagation by analyzed target subnets. Protocol anomalies indicating specific attacks are discovered in stage three. This example considers the anomalous ratio among outgoing and incoming packets that usually accompanies DDoS attacks.

'Vanguard' [99] is a detection method addressing random-interval and low-rate DoS attacks. Luo at al.'s technique is formal in nature: they proposed a detection method for polymorphic DoS attacks that records anomalies in TCP traffic. The decision has mainly based on the ratio between incoming and outgoing data/ACK packets. Vanguard has been enforced as Snort preprocessor plug-in, introducing a more specific and improved tested technique than Gamer's model [98]. Nevertheless, its fine attention on low-rate DoS attacks confines its use.

On the ontology side, the scheme by Ansarinia et al. [100] offers an interesting take on the recognition of DDoS attacks. The authors model consequences like unwanted disclosure and prerequisites of such attacks and automatically produce attacks ontology based on Mitre's CAPEC [101], CVE [102], and CWE [103] threat information. IDS and events logs are aggregated and converted to a solitary format: CEE [104]. Finally, it is possible to ontologically illustrate how vulnerability is constituted of certain weaknesses and how exploiting them directs to a successful attacks.

4.2 Host domain

Host-based solutions can be unstated as detection methods running on the endpoint. We recognized memory-based techniques, numerous behavioral analysis and detection systems, host-based intrusion detection systems, function call monitoring solutions, and more. For many of the tools, malicious software is something of a common denominator:

The majority of cyber-attacks involve malware transferred onto the system to carry out its threatening deed. Malware can usually be a software that performs something that induces damage to a computer, network, or user [105]. Examples include Trojan horses, viruses, worms, backdoors, spyware, scareware, or rootkits. Malicious software is well known to exploit vulnerabilities of the system it is designed to run on.

Malware motives include financial gain through blackmail or fraud, and it is commonly delivered to a huge number of recipients in the assumption that a sufficient number of users unknowingly will install it on their machines. Malware used for targeted attacks is much more sophisticated and usually includes additional components. For example, more complex command, control communication, and long term persistent installation that can be noticed in other malicious softwares. APT malware is

designed to attack a specific operating system, device, or specific application version. This influences the choice of evasion routines, dropping technique, and attacks against specific protective measures employed by the victim. While this fact makes such software risky to only a limited number of systems, the harm caused by targeted malware can be much harsher.

Malware analysis is probably the most broadly used and arguably most significant class of host-based schemes. Solutions range from analysis suites built up to determine a sample's common maliciousness to the interpretation of activities or clustering of prospective malware into families. Egele et al. [106] and Idika et al. [107] offer a survey of dynamic malware analysis tools and illustrate various detection and monitoring approaches. In [108], Wagner et al. elaborated on some of the tools and evaluated the information they provide. There is a huge amount of information to be collected from various malware analysis methods, each offering precise insight into the functionality and nature of a malicious program. This includes the virus description, packer information like packer designations and common compression information about the model, header and file information and code sections, function and library imports, CPU instructions and their related assembly operations, API and system calls executed by the model, file system operations indicating the modification, deletion, and creation of files, as well as interpreted process, thread, registry, and network operations.

Numerous data providers surveyed below concentrate on the analysis or detection of malware. We usually define malware data providers as tools that make use of static or dynamic analysis techniques as well as behavior-based and signature-based detection approaches to gathering information about a conceivably malicious piece of software [108].

The static analysis illustrates schemes that do not need the sample under scrutiny to be really executed. Based on the depth of analysis, a file may be checked for its fundamental properties like checksum, file type, easily extractable information such as import DLL information or null-terminated strings, or be fully disassembled [104]. The analysis environment, such as virtual machine, bare metal, or emulation, plays an insignificant role for static analyses—the analyst simply prefers a platform compatible with the tools of his/her preference. The dynamic analysis goes a step more and executes the file on a dedicated host premises. A variety of tools then monitor the execution and log appropriate information into an execution trace. The analysis environment is necessary for the dynamic technique, since the nature of data logged depends on both the techniques as well as on the platform used to capture system events.

On the detection part, we differentiate between behavior-based and signature-based techniques.

Signature-based techniques are best known for their prominent role in traditional intrusion detection systems and antivirus software. A so-called signature or definition is created to illustrate a parts or entire file of the code that are well known to be malicious [108]. The detection software then measures up the appearance of a packet or file to this set of known signatures. Signature-based detection has numerous shortcomings [109].

Firstly, confusion methods usually use metamorphic or polymorphic mutation to produce an ever-growing number of malware variants that are dissimilar in appearance, but similar in functionality. This directs to bloated signature databases and, finally, to an overall slowdown of the detection method. Secondly, signature-based methods only detect malware which has already been known and analyzed; new species or previously unidentified variants are frequently overlooked.

Behavior-based methods, on the other hand, concentrate on software behavior or specific system activities usually captured through dynamic analysis. Malicious events are defined through behavioral or patterns anomalies. Since the behavior-based technique can be semantics aware, it is largely immune to befuddlement [109].

Another big area of host-based threat mitigation is intrusion detection technique. IT systems intrusion illustrates the act of accessing a local or network-based assets without proper authorization or approval. It can essentially be defined as computer trespass leaving the targeted system vulnerable to sabotage or theft [110]. In many cases, the line between malicious software and black-hat hacking becomes blurred [111]. Most usual intrusions involve tools or malware designed to mislead security measures of the target system.

On the defense part, intrusion detection is mainly concerned with activities deemed unlawful by the system operator [110]. These may consist of digital breaking and entering but usually encompasses all 'malicious actions' targeted at resource abuse or privilege acquisition. Literature separates two classes of IDS: host based and network based [112]. The core differences are the nature and number of sources utilized to detect adversary activity. Host-based systems attempt to discover attacks taking place on the various machine in which the sensor component is installed on. Network-based IDS, on the other hand, analyze the traffic transmitted among systems and usually utilize a number of sensors distributed all over the network.

There are two techniques to detect malicious activity through IDS: pattern detection and anomaly detection [110]. Anomaly detection is based on the system that illegal activity or action manifests as anomaly and that it can be recognized through measuring the variation of certain key metrics. This may comprise the extreme use of system functions, high resource consumption at unusual times, or other behavior conflicting from a defined baseline.

Misuse or pattern detection, on the other hand, is based on pre-defined patterns. Information of an attacker's techniques or the probable consequences of an attack is converted into behavior sequences that can be observed by an IDS looking for their happening [110].

Many of the reviewed papers explain various host-based techniques to locate, identify, classify, or analyze malicious software. Both semantics-aware and semantics-based schemes are used.

4.3 Semantic domain

Unlike the other two categories, this section focuses strictly on formal definitions and general ontology models. Only schemes that specifically revolve around ontologies, formal definitions, and other semantic schemes are reviewed here. Semantics-aware

solutions designed to offer attack detection on the network or host can be found above in the respective subsections.

A number of models and some selected data providers concentrate on semantics-based detection of malicious behavior or cyber-attacks in general. Mostly, the term 'semantics' is used to refer to the process of assigning meaning to the specific patterns of network packets or system functions. Other solutions recognize sequences of code that generate identical results.

4.3.1 General semantic models and ontologies

Landwehr et al. [113] were the first one among researchers who presented a taxonomy of computer program security flaws. Their work laid the foundation for later attack models. Raskin and Nirenburg [114] eventually presented a semantic technique to information security with respect to the unification of nomenclature and terms. Nowadays, several application-independent semantic models can be found in the literature.

Razzaq et al. [85] present a general scheme to ontology-based attack detection and claim its suitability for web application security purposes. Although they also propose a domain-specific model [84], the paper mainly brings up general ontology engineering methodologies such as 'Methontology' [115]. A layered model for ontology design is introduced.

The paper by Anagnostopoulos et al. [116] is a practical example for the application of semantics to common intrusion scenarios. The authors attempt to classify and anticipate attacker intentions through a Bayesian classifier and a conditionally probabilistic inference algorithm. Their semantic scheme includes both illegitimate and legitimate actors, the formal characterization—labeled behavior—of the actor, activities in the form of events, commands issued, and an overall state of attack actuated by specific commands.

On the other hand, Yan et al. [117] concentrate on the conversion of raw sensor alerts data into a machine-understandable format to allow easier data fusion. They advertise the practice of a Principal-subordinate Consequence Tagging Case Grammar (PCTCG) that includes location, object, 'is part of' and 'has object' rules, method, cause, attacks stage and consequence of an intrusion.

4.3.2 Languages and models

Several languages form the base for many semantic models. Meier [118] presents a good overview of techniques by brought in a model of attacks signature to utilize on pattern detection systems. Zimmer and Unland [119] offer a model for the semantics of database events based on a meta-model. The author identifies some types of information appropriate for misuse detection. Exploit languages utilized to encode attacker actions, event languages that signify the information to be analyzed by an IDS such as NAOS, SNOOP, ACOOD, and HiPAC, detection languages used to describe state-transition-based languages like IDIOT IDS, STATL, signatures rule-based P-BEST, response languages, algebraic languages such as ADeLe, Sutekh, and LAMBDA, and report languages such as IDMEF [120].

For example, Totel et al. [121] associate events from different IDS sources to combat the typically high number of false positives. The authors developed a language ADeLe, specifically tailored to describe attacks and exploits from the target's perspective to the intrusion response. Their correlation model offers occurrence, recurring events, event sequences, and time constraints. While ADeLe is not an analysis or data provider system, its event association capabilities make it specifically useful for describing multi-stage attacks.

4.4 Multi-source domain

Multi-source techniques typically focus on event correlation and data fusion. The main categories within this domain are log correlation systems and security information and event management (SIEM) like solutions.

Logs are recorded files generated by a wide range of applications and devices. Their prime purpose is non-repudiation through system diagnostics and event auditing. Log analysis models retrieve log files and evaluate their contents. This can again be done using anomaly or pattern detection and will generally yield simplified alerts or anomalous log entries.

Log analysis is generally utilized in conjunction with multi-source event fusion. Solutions such as SIEM systems take log files of such as traffic flow analyzers, several intrusion detection systems, and OS event logs to visualize or correlate attacks. SIEM systems typically do not monitor assets on their own; they merely alerts, process logs, and other monitoring reports created and supplied by other tools. Multi-source analysis is the appropriate thing to attack interpretation system presently in the market.

SIEM-like and SIEM systems event fusion tools have become more and more important in today's cyber-defense. SIEM development has engendered open source solutions like OSSIM [122] and various commercial products. Combined with the evaluation of multi-source event aggregation and correlation, conventional log files are a promising new technique to understanding cyber-attacks.

4.4.1 SIEM-like systems

SIEM-like systems are archetypal of the multi-source domain. One of the initial multi-source techniques was introduced by Gorodetski et al. [123]. They present a paper on Multi-Agent System (MAS) technology for IDS involving attacks simulation and intrusion detection learning. The authors formally described the mapping attacker intentions to actions. The proposed simulator considers data from the OS audit trail, network traffic data, application audit data, and system logs. Combined attacks with shared source IP addresses are found via pattern matching on pre-processed input streams.

Bhatt and Gustavsson [124] presented a common APT attack model following the intrusion kill chain [125]. The logging component collects security events and various logs and sends them for analysis. Malware forensics is part of the framework but not discussed in detail. A dedicated intelligence unit is accountable for event correlation and searching.

The system proposed in [126] goals to build real-time situational awareness to detect multi-stage attacks through semantic event fusion. Event streams from intrusion detection modules are interrelated with pre-defined alert templates and mapped to several categories such as intrusion or scan, services (FTP, web), consequences (DoS), and protocol stacks. Attack criticalness is also modeled. Unlike Bhatt's mainly time-based approach [124], Mathew et al.'s bases for association are similarities of semantically linked events and IP addresses. The authors implemented their scheme using the fusion engine INFERD [127] and the model editor FUME [128] on an emulated OSIS network [129].

Atighetchi et al. [130] introduce 'Gestalt', a cyber-information management scheme that simplifies the access to event data stored on different systems. The focus lies on forensics: the actual techniques and methods essential to access the data are abstracted and defined using a novel cyber defense language (CDL). Sadighian et al. [131] offer an alert fusion method that incorporates public vulnerability data (NVD, CVE) and related attacks information such as host settings, network configurations, user-specific configurations, and application requirements. These data are recovered from a dedicated configuration management system. The gathered information is then transformed to a unified format and linked with common IDS alerts or signals. A pre-populated set of ontologies for context information, vulnerabilities, and alerts is used as basis for the subsequent decision process. In contrast to other solutions, Sadighian's scheme concentrates on alerts gathered by various sensors but depicting the same event. This potentially reduces irrelevant and redundant information.

4.4.2 Event correlation solutions

The second attention of multi-source data analysis is a correlation. In contrast to SIEM systems, the focus here lies more on the primary correlation algorithms and less on data management or generation considerations. This promises high collaboration potential among solutions of the two categories.

In [132], the authors present the application of data mining methods to identify patterns in data. A consolidation of association analysis, which focus to discover interesting relationships, and similarity-based propagation analysis is applied to fuse log events into semantic incidents. User data, events, and host data from both the antivirus software and OS are used as input for classification.

Langeder [133] presents a framework for dynamic threat identification and combines it with a proof-of-concept categorization comparing decision trees, SVM, and Bayes.

Patterned rules are extracted from a training environment and contain attributes such as time, HTTP status and request Information, users, IP addresses, ports, and more. Best results were realized with the SVM method. However, processing performance was only evaluated for small data sets. Further domain testing is essential to generically compare the several classification techniques.

Strasburg et al. [134] present S-MAIDS, a semantic model for correlation, automated tuning, and response selection in IDSs based on noticeable attack indicators, the authors call 'signals'. Each signal is breaking down into a characteristic/domain

Table 2 Existing research on general category and knowledge generation techniques on APT

Type	Classification	Parameter	References
Category	Domain	Host	[73,106–108]
		Network	[62,77,80,83,84,93,95,99,100]
		Multi-source	[124,126,130,132,133]
		Semantic domain	[85,113,116,117,121]
	Goal	Prediction	[77,83,116]
		Intelligence	[83,93,95,116,124,130]
		Correlation	[77,83,84,93,126,130,132,133]
		Detection	[73,77,80,83,84,89,99,100]
		Analysis	[73,80,97,124]
	Threat type	Malware	[62,84,124,132]
		Host intrusion	[62,73,77,83,93,116,124,130]
		Network intrusion	[62,80,83,93,95,99,100,116,124,126,130,133]
Knowledge generation techniques	learning	Supervised	[76,77,123,132]
		Un supervised	[133]
		Logical	[116,117,132]
	Clustering	Neural network	[76]
		Bayesian	[77,116,123,133]
		Decision tree	[133]
		Markov	[117]
	Threat type	Behavior graph	[76,133]
		Semantic network	[77,116]

such as a type constraint like integer, the high-level protocol used such as TCP, and a value like 80. The proposed scheme is formalized by OWL. Cross-system correlation was evaluated using IIS log messages and the output of a Netflow-aware model. As a reasoning-based ontology, S-MAIDS need predefined attack responses to be introduced in the knowledge base.

Table 2 summarizes the existing researches on general categorization for a list of papers and with knowledge generation capabilities on APT.

5 Open issues for APT attacks

5.1 Research issues

We evaluate and discuss our research paper in this section. The main goal of the paper was to provide an overview of approaches and methods that could be employed to strengthen an organization's defense against APT and targeted attacks in general.

Effective solutions tailored to the APT realm are very rare, presenting researchers with sufficient opportunity to develop dedicated solutions that transport the conventional cyber-defense mechanisms to this advanced threat domain.

Particularly, there are two open research issues addressed by this paper:

Issue 1. Which models, formal definitions, frameworks and existing tool to describe information system attacks?
Issue 2. Which type of promising approaches to APT detection and how can they be classified?

In answer to issue 1, we have identified various models, behavior pattern frameworks, formal definitions, and tools that describe information system attacks. The techniques and solutions they employ can serve as the foundation for the design or technical implementation of a system proficient to identifying the targeted attacks on the various levels: the host, the network, or a custom combination thereof. Fortified with this informative knowledge, future research should be focused and prioritized in accordance with the user's specific needs in the format of data, specific approaches to gathering and monitoring as well as knowledge generation capabilities.

Many literature and studies useful to analyzing targeted attacks are identified as well as their APT stages affinity tag. This addresses research question issue 2 ("What are promising approaches to APT detection and how can they be classified?") by promoting solutions to deem promising for APT detection and analysis. APT attacks behavior and their countermeasures are categorized into network-based, host-based, purely semantic and multi-source that cannot be imputed to a specific domain. There is still scope for improvement in the areas of reinforcement as well as self-evident and operational semantics. By familiarity with significance, information era was resolved to be of most extreme significance. Many papers suggest functionality to enable the extraction of certain behavioral particularities, which support the process of learning typical APT activities.

5.2 Proposing next-generation threat life cycle

Figure 6 shows the process of next-generation threat life cycle. The following steps gives you the detail of stages:

First phase It is reconnaissance process in which the Spearheading defined target is the primary goal of the initial intrusion. The attackers have started by defining a target and examine the attacks vectors and all entry points and potential vulnerabilities are enumerated.

The custom tools are developed with high-quality control and they are tested to bypass security mitigations. Also attackers make the schedule in day and time of the initial attacks is determined.

Second phase The vulnerability is exploited and attacks vectors could be Zero-days, SQL Injection, Design flaws, VoIP and PTSN, or spear phishing campaigns. Then, intruder gains limited access to an endpoint device that can be further used to elevate

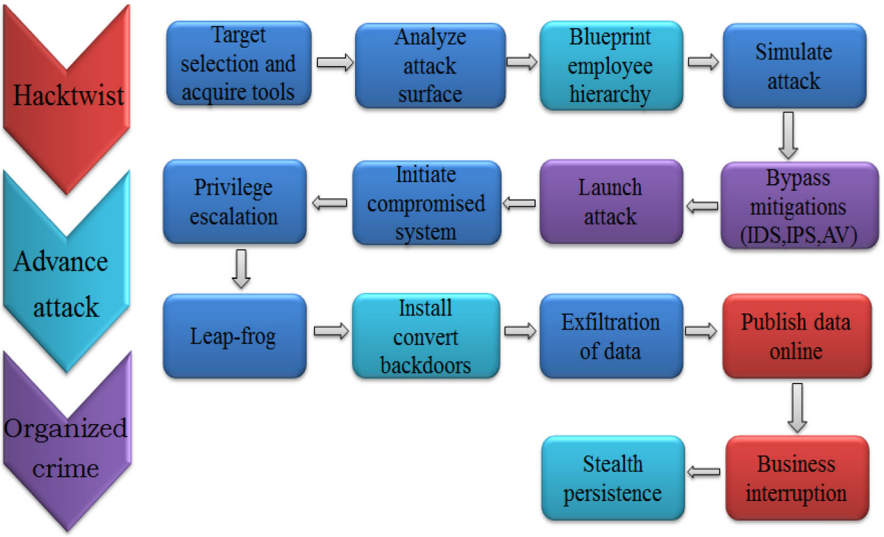

Fig. 6 Next-generation threat life cycle

privileges. The compromised endpoints are now used as a communications radar and information gathering device.

Third phase In this phase, the privileges are escalated after obtaining the secrete stolen credentials from the compromised endpoint. The initial compromised end point is now a pivoting point for a lateral movement or leapfrogging. Then, attackers spread their access, pursuing data worth exfiltration. Internal vulnerabilities of system are exploited to enhance access privilege and stealth.

Fourth phase In this phase, the e-mail and backup server data are retrieved, databases are breached, and VoIP conversations are eavesdropped or manipulated. And remote access tools, Key loggers and scripts such as Dark Comet, Dark Shades, web shells, Poison Ivy, and custom shell codes are deployed to exfiltrate the data. Data are quietly exfiltrated through genuine exit points and due to anti-signature evasion through file undetectable (FUD) no alarms are triggered in the system.

Fifth phase In this phase, the covert channels are created by building countless backdoors which can be any erratic technique. The evidence is abolished or corrupted using Anti-Forensics techniques. The attackers (hacktivists, targeted, criminal) exfiltrated data can be used to operate dumps to public forums. This phase is aiming for business interruption, fraudulent transactions, extortion, or competitive advantage.

6 Conclusion

APTs are sophisticated, specific and evolving threats, yet certain patterns can be identified in their process. APT first gained attention in the previous decade, and damages caused by the APT have recently started to become more severe. In this paper, we

studied the modeling phase of APT and its behavior pattern. The paper also included the different types of APT and Zero-day attack including cyber trends in the world. We studied approximately 50 articles that introduce models, method, framework or different goals that could potentially contribute to defense against APT. To get the knowledge generation through learning and classification, we surveyed articles which focus on clustering, learning and extraction techniques.

With ontologies and common semantics-based methods, an increasing number of attack models have observed their way into the field of information security. The focus on specific techniques changed little over the years. We have observed that most of the articles are processed system events for a number of categories, including learning techniques. Even the use of semantics-based approaches did not significantly increase. This shows that the focus is slowly shifting towards the detection and analysis of targeted attacks.

To simplify prioritization, the familiarized categorization enables researchers to conveniently browse for a best suited solution to their particular endeavor. In the last, the paper described the discussion and future direction, which summarized the paper and next-generation threat life cycle of APT, complemented by design concept of defense framework to detect and analyze a targeted attack.

Acknowledgments This work was supported by the Institute for Information and communications Technology Promotion (IITP) Grant funded by the Korea government (MSIP) (No. R-20160222-002755, Cloud based Security Intelligence Technology Development for the Customized Security Service Provisioning.

References

1. Chen P, Desmet L, Huygens C (2014) A study on advanced persistent threats. In: ifip International Conference on Communications and Multimedia Security, pp 63–72
2. Jeun I, Lee Y, Won D (2012) A practical study on advanced persistent threats. Computer applications for security, control and system engineering. Springer, Berlin, Heidelberg, pp 144–152
3. Moon D, Im H, Lee JD, Jong Park H (2014) MLDS: multi-layer defense system for preventing advanced persistent threats. Symmetry 6(4):997–1010
4. Tankard C (2011) Advanced persistent threats and how to monitor and deter them. Netw Secur 8:16–19
5. Sood AK, Enbody RJ (2013) Targeted cyberattacks: a superset of advanced persistent threats. IEEE Secur Priv 11(1):54–61
6. Friedberg I, Skopik F, Settanni G, Fiedler R (2015) Combating advanced persistent threats: from network event correlation to incident detection. Comput Secur 48:35–57
7. Ask M, Bondarenko P, Rekdal JE, Nordbø A, Bloemerus P, Piatkivskyi D (2013) Advanced persistent threat (apt) beyond the hype. Project report in IMT4582 Networn security at Gjovin University College. Springer. https://andynor.net/static/fileupload/434/S2_NetwSec_Advanced_Persistent_Threat.pdf. Accessed 11 May 2016
8. Bodmer S, Kilger M, Carpenter G, Jones J (2012) Reverse deception: organized cyber threat counter-exploitation. McGraw Hill Education. https://www.mhprofessional.com/details.php?isbn=0071772499. Accessed 24 June 2016
9. Bilge L, Dumitras T (2012) Before we knew it: an empirical study of zero-day attacks in the real world. In: Proceedings of the 2012 ACM Conference on Computer and Communications Security. ACM, pp 833–844
10. Zetter K (2011) How digital detectives deciphered Stuxnet, the most menacing malware in history. Wired Mag 11:1–8
11. Falliere N, Murchu L, Chien E (2015) W32.Stuxnet.Dossier. https://www.symantec.com/content/en/us/enterprise/media/security_response/whitepapers/w32_stuxnet_dossier. Accessed 10 May 2016

12. Mustafa T (2013) Malicious data leak prevention and purposeful evasion attacks: an approach to advanced persistent threat (APT) management. In: Electronics, Communications and Photonics Conference (SIECPC), Saudi International. IEEE, pp 1–5
13. Information-technology Promotion Agency, Design and Operational Guide to Cope with advanced persistent threats. Japan (IPA) (2011). https://www.ipa.go.jp/security/english/third.html. Accessed 25 Apr 2016
14. Smith AM, Toppel NY (2009) Case study: using security awareness to combat the advanced persistent threat. In: 13th Colloquium for Information Systems Security Education, pp 64–70
15. Hoglund G (2009) Advanced persistent threat, what APT means to your enterprise. http://www.issa-sac.org/info_resources/ISSA_20100219_HBGary_Advanced_Persistent_Threat.pdf. Accessed 22 Mar 2016
16. Dixon CJ, Pinckney T (2013) Indicating website reputations based on website handling of personal information. US Patent no. US 2006/0253583 A1
17. Bhatti AT (2015) Integrated analysis on case study of steve gibson ddos attack may 4th, 2001: performance of testing tools and in the context of business. Int J Res Comput Appl Robot 3(7):8–12
18. Cova M, Kruegel C, Vigna G (2012) Detection and analysis of drive-by-download attacks and malicious JavaScript code. In: Proc. 19th Int'l Conf. World Wide Web, ACM
19. Sood AK, Enbody RJ (2011) Browser exploit packs death by bundled exploits. In: Proc. 21st Virus Bulletin Conf
20. Spear-Phishing, watering hole and drive-by attacks: the new normal. Invincea, Inc. https://www.invincea.com/wp-content/uploads/2013/10/Invincea-spear-phishing-watering-holedrive-by-whitepaper-2013.pdf. Accessed 20 June 2016
21. Kim CH, Kim S, Kim JB (2016) A study of agent system model for response to spear-phishing. Int Inf Inst Tokyo Inf 19(1):263
22. Branco R (2011) Into the darkness: dissecting targeted attacks. Qualys Blog. https://blog.qualys.com/securitylabs/2011/11/30/dissecting-targeted-attacks. Accessed 16 July 2016
23. Appelt D, Nguyen CD, Briand LC, Alshahwan N (2014) Automated testing for SQL injection vulnerabilities: an input mutation approach. In: Proceedings of the 2014 International Symposium on Software Testing and Analysis. ACM, pp 259–269
24. Huang W, Hsiao C, Lin N (2011) Mass meshing injection: Sidename.js (now cssminibar.js) ongoing. Armorize Malware Blog. http://blog.armorize.com/2011/06/mass-meshing-injectionsidenamejs.html. Accessed 14 June 2016
25. Huang W, Hsiao C, Lin N (2011) Malvertising on google doubleclick ongoing. Armorize Malware Blog. http://blog.armorize.com/2011/08/k985ytvhtm-fake-antivirus-mass.html. Accessed 26 July 2016
26. Zhang YL, Xia GS (2013) The SSL MIMT attack with DNS spoofing. In: Applied Mechanics and Materials, vol. 385. Trans Tech Publications, pp 1647–1650
27. Wang Z (2014) POSTER: on the capability of DNS cache poisoning attacks. In: Proceedings of the 2014 ACM SIGSAC Conference on Computer and Communications Security. ACM, pp 1523–1525
28. Yuan L, Chen CC, Mohapatra P, Chuah CN, Kant K (2013) A proxy view of quality of domain name service, poisoning attacks and survival strategies. ACM Trans Internet Technol (TOIT) 12(3):9
29. Yamada A, Kim THJ, Perrig A (2012) Exploiting privacy policy conflicts in online social networks. Technical Report: CMU-CyLab-12-005, Carnegie Mellon University
30. Balduzzi et al M (2012) A security analysis of Amazon's elastic compute cloud service. In: Proc. 27th Ann. ACM Symp. Applied Computing, ACM
31. Ferrie P, Szor P (2004) Cabirn fever. Virus Bulletin Magazine
32. Stavrou A, Wang Z (2011) Exploiting smart-phone USB connectivity for fun and profit. In: BlackHat DC Conf
33. Rutkowska J (2009) Thoughts about trusted computing. In: EuSecWest Conf
34. Wang L, Jajodia S, Singhal A, Cheng P, Noel S (2014) k-zero day safety A network security metric for measuring the risk of unknown vulnerabilities. IEEE Trans Dependable Secure Comput 11(1):30–44
35. Recent Zero-day exploits and vulnerabilities. https://www.fireeye.com/current-threats/recent-zero-day-attacks.html. Accessed 13 June 2016
36. What is a zero-day vulnerability? http://www.pctools.com/security-news/zero-day-vulnerability/. Accessed 6 July 2016
37. https://en.wikipedia.org/wiki/Zero-day_(computing). Accessed 15 May 2016

38. Choi J, Choi C, Lynn HM, Kim P (2015) Ontology based APT attack behavior analysis in cloud computing. In: 2015 10th International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA), pp 375–379
39. James PF, Rohozinski R (2011) Stuxnet and the future of cyber war. Surv Glob Polit Strat 53(1):23–40
40. Karnouskos S (2011) Stuxnet worm impact on industrial cyber-physical system security. In: 37th Annual Conference on IEEE Industrial Electronics Society, pp 4490–4494
41. Langner R (2011) Stuxnet: dissecting a cyber warfare weapon. IEEE Secur Priv 9(3):49–51
42. Falliere N, Murchu LO, Chien E (2011) W32.Stuxnet Dossier, Symantec security response, Version 1.4
43. Parmar B (2012) Protecting against spear-phishing. Comput Fraud Secur 2012(1):8–11
44. Caputo DD, Pfleeger SL, Freeman JD, Johnson ME (2014) Going spear phishing: exploring embedded training and awareness. IEEE Secur Priv 12(1):28–38
45. Faisal Mohammad, Ibrahim Mohammad (2012) STUXNET, DUQU and beyond. Int J Sci Eng Investig 1(2):75–78
46. Bencsáth B, Pék G, Buttyán L, Félegyházi M (2012) The cousins of Stuxnet: Duqu, Flame, and Gauss. Future Internet 4(4):972–1003
47. Chen P, Desmet L, Huygens C (2014) A study on advanced persistent threats. Commun Multimed Secur 8735:63–72
48. http://www.enterpriseitnews.com.my/malaysia-organizations-more-likely-to-be-targeted-with-cyber-attacks-fireeye-report/3.4ref. Accessed 10 June 2016
49. https://www.fireeye.com/current-threats/annual-threat-report.html3.4ref. Accessed 19 June 2016
50. http://www.computerweekly.com/news/4500260196/Cyber-attacks-an-increasing-concern-for-Asean-countries. Accessed 10 May 2016
51. http://www.computerweekly.com/news/4500260196/Cyber-attacks-an-increasing-concern-for-Asean-countries. Accessed 5 July 2016
52. Davis J, Clarck A (2011) Data preprocessing for anomaly based network intrusion detection: a review. Comput Secur 30:353–375
53. Kai HM, Liu XJ, Liu YF, Zhou L (2011) Reducing false negatives in intelligent intrusion detection decision response system. Appl Mech Mater 128:676–681
54. Sommer R, Paxson V (2010) Outside the closed world: on using machine learning for network intrusion detection. In: IEEE Symposium on Security and Privacy, Oakland
55. Zhou C, Leckie C, Karunasekera S (2010) A survey of coordinated attacks an collaborative intrusion detection. Comput Secur 29:124–140
56. Sperotto A, Schaffrath G, Sadre R, Morariu C, Pras A, Stiller B (2010) An overview of IP flow-based intrusion detection. IEEE Commun Surv Tutor 12(3):343–356. doi:10.1109/SURV.2010.032210.00054
57. Trammell B, Claise B (2015) Specification of the IP flow information export (IPFIX) protocol for the exchange of flow information. https://tools.ietf.org/html/rfc7011. Accessed 29 July 2015
58. Cisco: Cisco IOS NetFlow. http://cisco.com/c/en/us/products/ios-nx-os-software/ios-netflow/index.html. Accessed 29 July 2015
59. University of California: KDD Cup 1999 Data. http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html. Accessed 29 July 2015
60. MIT Lincoln Laboratory: DARPA Intrusion Detection Evaluation. http://www.ll.mit.edu/ideval/data/. Accessed 29 July 2015
61. Julisch K, Kruegel C (2005) Detection of intrusions and malware, and vulnerability assessment. In: Proceedings of 2nd International Conference, DIMVA Vienna, Austria, July 7–8. Springer, New York
62. Abdoli F, Kahani, M (2009) Ontology-based distributed intrusion detection system. In: Computer Conference, 2009. CSICC 2009,14th International CSI. IEEE, pp 65–70
63. W3C: Semantic web. http://www.w3.org/standards/semanticweb/. Accessed 29 July 2015
64. Chiang HS, Tsaur WJ (2009) Ontology-based mobile malware behavioral analysis. Da-Yeh University, Changhua
65. Huang HD, Chuang TY, Tsai YL, Lee CS (2010) Ontology-based intelligent system for malware behavioral analysis. In: Fuzzy Systems (FUZZ), IEEE International Conference on, pp 1–6
66. W3C: SPARQL 1.1 Overview. http://www.w3.org/TR/sparql11-overview/. Accessed 29 July 2015
67. Stanford Center for Biomedical Informatics Research: Protégé. http://protege.stanford.edu/. Accessed 29 July 2015
68. Apache Software Foundation: Apache JENA. https://jena.apache.org/. Accessed 27 July 2015

69. Christodorescu M, Jha S, Seshia S, Song D, Bryant RE (2005) others: Semantics-aware malware detection. In: Security and Privacy, IEEE Symposium, pp 32–46
70. Scheirer W, Chuah MC (2008) Syntax vs. semantics: competing approaches to dynamic network intrusion detection. Int J Secure Netw 3(1):24–35
71. Hirono S, Yamaguchi Y, Shimada H, Takakura H (2014) Development of a secure traffic analysis system to trace malicious activities on internal networks. In: Proceeding of IEEE 38th Annual Conference on Computer Software and Applications Conference (COMPSAC). IEEE, pp 305–310
72. Cortes C, Vapnik V (1995) Support-vector networks. Mach Learn 20(3):273–297
73. Andersson S, Clark A, Mohay G, Schatz B, Zimmermann J (2005) A framework for detecting network-based code injection attacks targeting Windows and UNIX. In: Computer Security Applications Conference, 21st Annual, p 10
74. Chien SH, Chang EH, Yu CY, Ho CS (2007) Attack sub plan based attack scenario correlation. Int Conf Mach Learn Cybern 4:1881–1887
75. Cisco: Snort.Org. https://www.snort.org/. Accessed 10 Jan 2015
76. Zhu B, Ghorbani AA (2005) Alert correlation for extracting attack strategies. Ph.D. thesis, Citeseer
77. AlEroud A, Karabatis G (2013) A system for cyber attack detection using contextual semantics. In: 7th International Conference on Knowledge Management in Organizations: Service and Cloud Computing. Springer, New York, pp 431–442
78. He P, Karabatis G (2012) Using semantic networks to counter cyber threats. In: Intelligence and Security Informatics (ISI), IEEE International Conference on, pp 184–184
79. Shannon CE (2001) A mathematical theory of communication. ACM SIGMOBILE Mob Comput Commun Rev 5(1):3–55
80. Münz G, Carle G (2007) Real-time analysis of flow data for network attack detection. In: Integrated Network Management, 2007. IM'07. 10th IFIP/IEEE International Symposium on, pp 100–108
81. Vance A (2014) Flow based analysis of advanced persistent threats detecting targeted attacks in cloud computing. In: Info communications Science and Technology, 2014 1st International Scientific-Practical Conference Problems of, pp 173–176
82. Krishnamurthy B, Sen S, Zhang Y, Chen Y (2003) Sketch-based change detection: methods, evaluation, and applications. In: Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement, pp 234–247
83. Aleroud A, Karabatis G (2014) Context infusion in semantic link networks to detect cyber-attacks: a flow-based detection approach. IEEE, pp 175–182
84. Razzaq A, Latif K, Ahmad HF, Hur A, Anwar Z, Bloodsworth PC (2014) Semantic security against web application attacks. Inf Sci 254:19–38. doi:10.1016/j.ins.2013.08.007
85. Razzaq A, Anwar Z, Ahmad HF, Latif K, Munir F (2014) Ontology for attack detection: an intelligent approach to web application security. Comput Secur 45:124–146. doi:10.1016/j.cose.05.005
86. McGuinness DL, Van HF (2004) OWL web ontology language overview. W3C Recomm 10(10):101
87. Meier M (2004) A model for the semantics of attack signatures in misuse detection systems. In: Information security. Lecture notes in computer science, vol 3225. Springer, New York, pp 158–169
88. Guarino N, Welty CA (2009) An overview of OntoClean. In: Handbook on ontologies. Springer, New York, pp 201–220
89. Razzaq A, Ahmed HF, Hur A, Haider N (2009) Ontology based application level intrusion detection system by using Bayesian filter. In: Computer Control and Communication, 2009. IC4 2nd International Conference on, pp 1–6
90. Sangeetha S, Vaidehi V (2010) Fuzzy aided application layer semantic intrusion detection system—FASIDS. Int J Netw Secur Appl 2(2):39–56
91. Farrell JA (2015). http://www.cs.man.ac.uk/~pjj/farrell/comp2.html#EBNF. Accessed 29 July 2015
92. Kosko B (1986) Fuzzy cognitive maps. Int J Man Mach Stud 24(1):65–75
93. Balduzzi M, Ciangaglini V, McArdle R (2013) Targeted attacks detection with spunge. In: 11th Annual International Conference on Privacy, Security and Trust (PST), 2013, pp 185–194
94. Levenshtein VI (1966) Binary codes capable of correcting deletions, insertions, and reversals. Sov Phys Doklady 10:707–710
95. Thakar U, Dagdee N (2010) Pattern analysis and signature extraction for intrusion attacks on web services. Int J Netw Secur Appl 2(3):190–205. doi:10.5121/ijnsa.2010.2313
96. W3C: SOAP Version 1.2 Part 1: Messaging Framework (Second Edition). http://www.w3.org/TR/soap12/ . Accessed 22 July 2015

97. Zarras A, Papadogiannakis A, Gawlik R, Holz T (2014) Automated generation of models for fast and precise detection of HTTP based malware. In: 12th Annual International Conference on. Privacy, Security and Trust (PST), pp 249–256
98. Gamer T, Scholler M, Bless R (2006) A granularity-adaptive system for in-network attack detection. In: Proceedings of the IEEE/IST Workshop on Monitoring, Attack Detection and Mitigation, pp 47–50
99. Luo X, Chan EW, Chang RK (2006) Vanguard: a new detection scheme for a class of TCP-targeted denial-of-service attacks. In: Network Operations and Management Symposium, NOMS, 10th IEEE/IFIP, pp 507–518
100. Ansarinia M, Asghari SA, Souzani A, Ghaznavi A (2012) Ontology-based modeling of DDoS attacks for attack plan detection. In: 2012 6th International Symposium on Telecommunications (IST), pp 993–998
101. MITRE Corporation: CAPEC-Common Attack Pattern Enumeration and Classification (CAPEC). https://capec.mitre.org/. Accessed 22 Sept 2015
102. MITRE Corporation: CVE-Common Vulnerabilities and Exposures (CVE). https://cve.mitre.org/. Accessed 22 Sept 2015
103. MITRE Corporation: CWE-Common Weakness Enumeration. https://cwe.mitre.org/. Accessed 22 Sept 2015
104. MITRE Corporation: Common Event Expression: CEE, A Standard Log Language for Event Interoperability in Electronic Systems. https://cee.mitre.org/. Accessed 29 July 2015
105. Sikorski M, Honig A (2012) Practical malware analysis: the hands-on guide to dissecting malicious software. No Starch, San Francisco
106. Egele M, Scholte T, Kirda E, Kruegel C (2012) A survey on automated dynamic malware-analysis techniques and tools. ACM Comput Surv (CSUR) 44(2):6
107. Idika N, Mathur AP (2007) A survey of malware detection techniques. Technical report 286, Department of Computer Science, Purdue University, USA
108. Wagner M, Fischer F, Luh R, Haberson A, Rind A, Keim D, Aigner W, Borgo R, Ganovelli F, Viola I (2015) A Survey of Visualization Systems for Malware Analysis. In: EG Conference on Visualization (EuroVis)-STARs, pp 105–125
109. Dornhackl H, Kadletz K, Luh R, Tavolato P (2014) Malicious behavior patterns. In: IEEE 8th International Symposium on Service Oriented System Engineering (SOSE), pp 384–389
110. Kumar S, Spafford EH (1994) A pattern matching model for misuse intrusion detection. In: Proceedings of the 17th National computer Security Conference, pp 11–21
111. Peyman K, Ali AG (2005) Research on intrusion detection and response: a survey. IJ Netw Secur 1(2):84–102
112. Wagner D, Soto P (2002) Mimicry attacks on host-based intrusion detection systems. In: Proceedings of the 9th ACM Conference on Computer and Communications Security, pp 255–264
113. Landwehr CE, Bull AR, McDermott JP, Choi WS (1994) A taxonomy of computer program security flaws. ACM Comput Surv (CSUR) 26(3):211–254
114. Raskin V, Hempelmann CF, Triezenberg KE, Nirenburg S (2001) Ontology in information security: a useful theoretical foundation and methodological tool. In: Proceedings of the Workshop on New Security Paradigms, pp 53–59
115. FernándezL M, Gómez-Pérez A, Juristo N (1997) Methontology: from ontological art towards ontological engineering. In: AAAI Symposium on Ontological Engineering, American Association for Artificial Intelligence
116. Anagnostopoulos T, Anagnostopoulos C, Hadjiefthymiades S (2005) Enabling attack behavior prediction in ubiquitous environments. In: Pervasive Services, 2005. ICPS'05, Proceedings of International Conference on, pp 425–428
117. Yan W, Hou E, Ansari N (2004) Extracting attack knowledge using principal-subordinate consequence tagging case grammar and alerts semantic networks. In: Local Computer Networks, 29th Annual IEEE International Conference on, pp 110–100
118. International secure systems lab: anubis-malware analysis for unknown binaries. https://anubis.iseclab.org/. Accessed 29 July 2015
119. Zimmer D, Unland R (1999) On the semantics of complex events in active database management systems. In: 1999, Proceedings of 15th International Conference on, Data Engineering, pp 392–399
120. Debar H, Curry D, Feinstein B (2015) The Intrusion Detection Message Exchange Format (IDMEF). https://www.ietf.org/rfc/rfc4765.txt. Accessed 29 July 2015

121. Totel E, Vivinis B, Mé L (2004) A language driven intrusion detection system for event and alert correlation. In: Proceedings at the 19th IFIP International Information Security Conference. Kluwer Academic, Toulouse, Springer, New York, pp 209–224
122. Alienvault: OSSIM: The Open Source SIEM | AlienVault. https://www.alienvault.com/products/ossim. Accessed 29 July 2015
123. Gorodetski V, Kotenko I, Karsaev O (2003) Multi-agent technologies for computer network security: attack simulation, intrusion detection and intrusion detection learning. Comput Syst Sci Eng 18(4):191–200
124. Bhatt P, Yano ET, Gustavsson P (2014) Towards a framework to detect multi-stage advanced persistent threat sattacks. Proceeding of IEEE 8th international symposium on service oriented system engineering (SOSE). IEEE, pp 390–395
125. Hutchins EM, Cloppert MJ, Amin RM (2011) Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Lead Issues Inf Warfare Secur Res 1:80
126. Mathew S, Upadhyaya S, Sudit M, Stotz A (2010) Situation awareness of multistage cyber attacks by semantic event fusion. In: Military Communications Conference, 2010-MILCOM 2010. IEEE, pp 1286–1291
127. Stotz A, Sudit M (2007) Information fusion engine for real-time decision-making (INFERD): a perceptual system for cyber attack tracking. In: Information Fusion, 2007 10th International Conference on, pp 1–8
128. Mathew S, Giomundo R, Upadhyaya S, Sudit M, Stotz A (2006) Understanding multistage attacks by attack-track based visualization of heterogeneous event streams. In: Proceedings of the 3rd International Workshop on Visualization for Computer Security, pp 1–6
129. GlobalSecurity.org: Open Source Information System (OSIS). http://www.globalsecurity.org/intell/systems/ osis.htm. Accessed 29 July 2015
130. Atighetchi M, Griffith J, Emmons I, Mankins D, Guidorizzi R (2014) Federated access to cyber observables for detection of targeted attacks. In: Proceeding of IEEE on Military Communications Conference (MILCOM), IEEE. pp 60–66
131. Sadighian A, Zargar ST, Fernandez JM, Lemay A (2013) Semantic-based context-aware alert fusion for distributed Intrusion Detection Systems. In International Conference on, Risks and Security of Internet and Systems (CRiSIS), pp 1–6
132. Gabriel R, Hoppe T, Pastwa A, Sowa S (2009) Analyzing malware log data to support security information and event management: some research results. In: Proceeding of IEEE First International Conference on Advances in Databases, Knowledge, and Data Applications (DBKDA). IEEE, pp 108–113
133. Langeder S (2014) Towards dynamic attack recognition for SIEM. Ph.D. thesis, St. Poelten University of Applied Sciences
134. Strasburg C, Basu S, Wong JS (2013) S-MAIDS: a semantic model for automated tuning, correlation, and response selection in intrusion detection systems, In: Proceeding of IEEE 37th Annual Conference on Computer Softwareand Applications Conference (COMPSAC). IEEE, pp 319–328