Employing Link Analysis for the Improvement of Threat Intelligence Regarding Advanced Persistent Threats

Corey T. Holzer
Purdue University
cholzer@purdue.edu

J. Eric Dietz
Purdue University
jedietz@purdue.edu

Baijan Yang
Purdue University
byang@purdue.edu

Abstract

Over the past decade, the Advanced Persistent Threat (APT) has risen to forefront of cybersecurity threats. APTs are a major contributor to the billions of dollars lost by corporations around the world annually. The threat is significant enough that the *Navy Cyber Power 2020* plan identified them as a "must mitigate" threat in order to ensure the security of its warfighting network.

This paper examines the current state of APT malware detection and the challenge this represents, existing research into improving APT detection. It examines the concept of link-analysis data mining and its application in intelligence gathering in various environments, and proposes how link analysis may be employed in intelligence gathering in terms of APTs. Finally, it outlines the methodology of the authors' planned research into improving signature development with the goal of improving the detection of APT malware through understanding the relationships between various APT malware, their behavior patterns, and other characteristics that APTs share.

Introduction

This paper and its related research are intended to address how intelligence about the entire range of APTs can be used as a means for expanding the intelligence that we have on a single APT. Just as criminologists and members of the intelligence community have used link analysis to understand the relationships between organizations and individuals and improved knowledge regarding a single individual by looking at the whole, we posit that it should be possible to learn more about individual APTs by studying their relationships to the family of APTs.

The paper is broken down into nine sections. The first section will focus on defining Advanced Persistent Threats, both as cyber-attacks and as organizations. This section also

includes an examination of the cost, both financially and to data, that these attacks represent. The second section examines the several phases of an APT attack. The third section provides a brief history of attacks attributed to APTs. The fourth section explores why APTs are difficult to detect with a particular focus on the challenge of detecting malware. With the sketch of APTs and the challenges of their detection complete, the paper moves into its fifth section where the authors will briefly explore related research. Section six defines and evaluates Open Source Intelligence. Section seven defines the purpose of ontology development. The eighth section is a review the use of link analysis as an intelligence development tool. The ninth, and final section, presents the current research's problem statement, goals, and planned methodology.

Defining the Term Advanced Persistent Threat

The United States Air Force first coined the phrase Advanced Persistent Threat in 2006 [1–3]. They created the term in order to facilitate discussions about the characteristics, activities involved, and the classification of these types of attacks [2].

It is proper to begin this research endeavor by defining how Advanced Persistent Threat will be used for our purposes. The threat is *Advanced* in that adversaries can employ tactics that cover the full spectrum of cyber-attacks [1,2,4,5]. It is *Persistent* in that he is not an opportunistic hacker searching for easily infiltrated systems [1]. Instead, the persistent attack is one that will operates over an extended period of time with a pre-determined target and desired end state for the cyber-attack [1,5,6].

Today, the terms Advanced Persistent Threat and APT are used to refer both to the groups that execute the attack and as a collective term for all of the elements that go into the attack [4]. While the present research focuses on APT as an attack, we will use the term to refer to the organizations within this work. It is therefore necessary that we briefly discuss the context of APT as an entity to afford our readers and to provide a more holistic understanding of the term.

The APT as an Attack

The APT attack involves multiple methods, tools, and techniques to compromise the target and achieve what is usually a long-term objective [7]. Later in the paper we present the phases involved in an APT attack within the framework of the Lockheed Martin Cyber Kill Chain. It is important to note that as the APT attack phases should not be considered as discrete events where one phase ends and the next begins [7]. Keep in mind that there can be overlap as the attack propagates across the target network. It is also important to understand that while we use the term persistent it is not intended to mean that the attack is constantly active or that connections between the APTs' C2 server and compromised hosts is constant [2,8,9].

The sophistication and complexity of APT attacks make it hard for organizations to recognize one particular element as being only one piece of a larger plan [2,10]. In this game of cat and mouse the attacker has the advantage of having unlimited time, resources, the

victim organization's prioritization of its business processes, and less fear of prosecution when the attack takes place across international borders [11].

The APT as an Entity

In addition to describing the attack, the term APT is used to describe the organizations that execute these attacks. In this context the discussion is about organizations that are well funded, well organized, and patient [4]. They can infiltrate a network and remain hidden while monitoring it for a specific target or data to exfiltrate [5]. Their goal is stealthy execution instead of the kind of attack that draws attention to the person or persons committing the crime.

While it is understood that those responsible for APT attacks are well organized and that they possess significant funds, cybersecurity professionals must not confuse this with meaning that they are sponsored by state actors [2]. Cybersecurity professionals attribute some APTs to state sponsored actors but state sponsorship in not a required component of the definition [12,13].

With APT defined let us explore the costs, both in financial and national security terms, associated with the threat.

The Cost of the APT Threat

On a regular and increasingly frequent basis companies, government organizations, and industries are reporting breeches of their network and the extraction of thousands if not millions of data records. Despite the security measures these organizations put in place to ensure the security of the data customers provide to them. For example, in 2011 RSA spent $66 million USD to undo the damage caused by an APT attack [14]. In a 2015 study by the Ponemon Institute, the researcher estimated that it can cost up to $161 each record lost [15]. When one considers that some APT attacks can compromise millions of user records the cost could potentially bankrupt businesses.

The threat is not limited to consumer market. The US government's Office of Personnel Management reported a data breech in 2014 which involved 25,000 or more personnel records of government employees [16]. Breeches like this could present a risk to national security. The Stuxnet attack in 2010 had the potential of causing significant damage to nuclear facilities which could place lives and national infrastructures at risk as well [17,18]. It is for this reason that the U.S. President issued an Executive Order in 2013 calling for the development of a Cyber Resiliency Framework [19]. It also prompted the U.S. Navy to declare APTs as a "must mitigate" threat [20].

The Phases of an APT Attack

With the definitions and examination of costs complete, let us delve into the several phases that an APT attack employs in order to obtain the attacker's desired end state. One of the more widely accepted description of the APT Attack comes from Lockheed Martin. The

"Lockheed Martin Cyber Kill Chain" breaks the attack into seven phases. We will use their model as a means of discussing Lockheed Martin's definition of each phase. The seven phases are as follows:

- Reconnaissance
- Weaponization
- Delivery
- Exploitation
- Installation
- Command and Control
- Actions on Objective

Additionally, we will include elements that other cybersecurity professionals and professional organizations include when discussing the APT Attack.

Reconnaissance is the selection and identification of the desired target. In this stage the APT is footprinting the target organization and collecting information including but not limited to names, positions, email addresses, physical locations, operating systems, etc. [8]. Through this information gathering process the APT determines who has ownership of the desired information that they seek to steal [2]. The APT will determine which employee to compromise as well as a potential means for doing so.

In the Weaponization phase, the APT puts together the code that they will use to compromise a target system [8]. This will often involve the use of existing and proven code but, if needed, APTs will adapt or modify code in order to address a specific configuration or defensive challenge [2,4,8]. When using code designed for the specific target, the code has no anti-virus signature which the target company might use to detect it [21].

In the Delivery phase, the APT transmits the weapon to the targeted system [8]. Lockheed Martin identifies the most common delivery methods as email attachments, websites and removable media. In addition to those three, Ask, et.al. [2] identified social media as another means for launching at attack against an individual within the target organization. For the attack to move beyond this phase, the targeted individual most click on the link, attachment, or application for the attack to move into the next phase [11].

Exploitation involves compromising the host machine on the network. It is where the weaponized tool is triggered [8]. The exploitation can be of a flaw in the operating system or an individual application on the host [2,8].

The next phase of the attack is the Installation phase. Installation refers to the installation of a Remote Administration Tool (RAT) or backdoor that the APT can use to gain control of the target's computer [2,8]. Once the victim triggers the malicious code (e.g. by clicking the malicious link, opening the infected file, or visiting the compromised site, etc.) the code reaches back to its Command and Control (C2) server and provides the attacker with useful information about the target network's environment that could be useful in executing the

later stages of the APT attack [2]. Once installed the RAT can also lay dormant until the C2 server connects to it [2,22].

The Command and Control phase begins once the infected host beacons the C2 server [8]. Attackers need to maintain access to the victim's network means that each communication with a compromised system [11]. During this phase the APT will seek to obtain elevated privileges on the system and will install additional software to facilitate the attack (i.e., encryption) on compromised system and network [2]. While the initial Installation is software is designed to exploit a zero-day vulnerability the additional software is likely to be commonly known software that may even be approved to operate on the network for legitimate activities (e.g., SSH, SecureFTP, etc.) [2].

The final stage in Lockheed Martin's APT Kill Chain is the Actions on Objective phase. During this phase the APT is actively going after the data that they originally identified as their target [8]. The APT uses previously installed software to determine the network layout including, but not limited to, mapping the hosts of networked drives, database servers, domain controllers, PKI, etc. [2]. The goal here is to footprint the network and to establish a network account and elevate the privileges for that account [2]. During this phase, the APT will also seek to compromise more hosts in order to strengthen its foothold in the target network. The extraction of the target data may also be accomplished using custom encryption and/or tunneling within other protocols to hide the data from security professionals [21].

Conventionally, malware will remove itself once its task is complete or is discovered and removed by antivirus software [2]. The APT, however, is designed to stay invisible. It maintains persistence by reaching back to the C2 server for updates to the malicious code [2]. Changing code enables the APT attack to avoid detection. Mandiant's APT Attack model includes cleanup as part of this phase [12,23]. However, it is more likely that the APT will leave some software in place in order to facilitate quicker access if they wish to exfiltrate new information in the future. The security firm Mandiant has data demonstrating that a group identified as APT1 has left software in place to re-access a target network months, and even years, later [9,12,13].

With an understanding of the APT attack established, we will next look at some examples of cyber-attacks that were qualified as APTs. This is by no means intended to be an all inclusive list. It is intended to demonstrate the variety of attack elements and the variety of targets.

A Brief History of APT Attacks

With the APT attack method complete, let us examine some historical cyber-attacks which employ tactics attributable to an APT. This list is meant to be a brief example to illustrate the variety of implementations and targeted organizations.

Shady RAT - With earliest evidence indicating that this APT collected data in mid-2006, it is possible that it was stealing data even earlier than the logs provide [24]. Evidence collected by McAfee indicates that, unlike other APTs discussed here, this APT was used against a wide range of individuals and organizations in multiple industries. Initial installation took

place via a spear-phishing email. The attachment triggered the download and installation of malware that, in turn, created a backdoor communication channel with its C2 server. In four of 71 instances where Shady RAT gained control of a target system, it remained persistent for 24 or more months [24].

Night Dragon - This attack targeted the Global Energy Business Community [25]. Commencing in 2009, Night Dragon employed social engineering and spear-phishing attacks to exploit vulnerabilities and compromise Microsoft Active Directory machines. The initial targets were extranet web servers and then internal desktops and servers to gain elevated permissions within the hosts and target network. Finally, the APT harvested sensitive proprietary and project-financing related information sending that information back to C2 servers on the Internet [25].

Poison Ivy Attack on RSA – The APT identified two specific independent groups of RSA employees and crafted a spear-phishing campaign tailored to the target employees' job functions [26]. The email contained a spreadsheet which executed code that leveraged an Adobe Flash vulnerability in order to inject a Poison Ivy RAT which, in turn, established a hard to connect reverse connection to the APT's C2 servers. The data stolen included RSA's SecureID two-factor authentication products [27]. Executive Chairman Art Coviello issued an open letter to customers in which he acknowledged that the stolen information "could potentially be used to reduce the effectiveness of a current two-factor authentication implementation as part of a broader attack [27,28]."

Icefog - This APT attack has been used numerous times starting in 2011 with most of the attacks targeting organizations in Japan and South Korea [29]. Kaspersky Lab's Global Research & Analysis Team (GREAT) researched the attacks and determined that their targets were supply chains of "government institutions, military contractors, maritime and ship-building groups, telecom operators, satellite operators, industrial and high technology companies and mass media [30]." The APT achieved their initial insertion through spear-phishing campaigns and attachments which exploited known vulnerabilities in the host's operating system [30]. GREAT identified at least 6 variations in the manner in which Icefog exfiltrates data [29]. They are as follows (with designations established by GREAT) [29]:

- The "old" 2011 Icefog — sends stolen data by e-mail
- Type "1" "normal" Icefog — interacts with C2 servers
- Type "2" Icefog — interacts with a script-based proxy server that redirects attacker commands
- Type "3" Icefog — observed to use a certain kind of C2 via a different means of communication
- Type "4" Icefog — another C2 variation with a different means of communication
- Icefog-NG — communicates by direct TCP connection to port 5600

Stuxnet - In 2010 this worm was weaponized with the specific goal of impacting systems that run Supervisory Control And Data Acquisition (SCADA) engineered by Siemens [18]. The worm was initially uploaded via USB to a Windows workstation and started spreading across

the target network without impacting any systems that did not run the SCADA software. Once it entered a machine where the SCADA was present it would connect with its C2 server and receive software updates [18]. The worm then compromised the system and began gathering information in order to get the elevated permissions needed to take control of centrifuges making them fail. The software would also provide false information back to the user giving the appearance that everything was functioning normally [18].

New York Times Attack – In 2013, the New York Times announced that their network was compromised through the installation of malware which led to the extraction of the network's user database [31]. In its 2014 "M-Trends: Beyond the Breach" Report, Mandiant stated that the suspected APT group took specific steps to change their cyber operations following the disclosure in order to "better hide its activities [32]."

Trojan.APT.Seinup – Discovered in 2013, this Trojan targeted Google Docs. The APT attack leveraged this legitimate cloud based Software as a Service (SaaS) in order to leverage the legitimate Secure Socket Layer (SSL) of Google Docs to protect their malicious communications [33].

With this brief sample the reader can see that detecting APTs is a challenge for cybersecurity professionals. Next we will examine some of the technical reasons that make it so difficult.

Challenges in Detecting APTs

With the previous examples providing a context for understanding how APTs function we can now address what challenges exist in detecting APTs. While APTs can employ a variety of known attack elements (e.g., phishing, malware, etc.) which can be detected by current security measures, attackers are still able to execute their attacks unnoticed. The question for security professionals, therefore, how are these attackers employing these detectable tools in a manner that leaves them undetected.

Challenges Specific to APT Detection

Conventional means of intrusion detection often fail to detect APTs because they are implemented to mitigate risks associated with automated viruses and worms, not the focused manually-operated attack of an APT [8]. In its annual *M-Trends*, Mandiant [34], estimated that only 24% of APT malware is detected by security software. This is due to multiple factors. The target organization's decision not to inspect outbound packets [2,11,35]. Data is encrypted [2,35]. The affected machines send data to a trusted source [35].

Anti-detection Methods Employed with Malware

As discussed in the previous section, APTs commonly employ malware as a means to gain their initial foothold into a target network and to gain elevated permissions on host machines. Malware comes in many forms including Trojan horses, worms, rootkits, scareware, spyware, and viruses [36,22]. Regardless of which family the malware belongs it is software designed to remain undetected on an infected system [37].

Malware developers must overcome the various forms of detection the forensics analysts use to reveal the presence of malware as described above. Malware developers have varied methods at their disposal to defeat detection. These means fall into several areas:

Anti-emulation analysis - malware developers employ techniques which detect that their malware is running in a virtual environment and the malware will either stay dormant or use deception code to provide a false signature to forensic experts trying to dissect the malware [37].

Anti-online analysis - companies offer third party malware analysis services online. However, there are limits to how well these services work because the online environment may not match conditions to trigger the malware or it may act differently than it would in a real-world network [37].

Anti-analysis - Anti-analysis refers to changing the code such that it becomes harder to read during the analysis process [37,38]. These techniques target the way analysis is conducted. Code is deceptively transformed such that the analysis tools cannot establish a signature for the malware [37]. De-obfuscation methods of analysis fail because that analysis happens on the files whereas the malware's transformation back into identifiable malicious code happens in memory as part of the unpacking process [36].

Anti-hardware - Malware developers can use check to determine whether the malware is being analyzed based on signatures of CPU usage and register usage during the debugging session [37].

Anti-debugger, anti-disassembler, and anti-tools - In the same way that malware can detect if the operating system is running in a virtual environment, malware developers can design their malware to detect if the code is being debugged, disassembled, or examined by other tools [37].

Anti-memory - In case the malware analyst is clever or experienced enough to dump memory as a means of defeating anti-analysis measures, the malware developer can use anti-memory measures in order to frustrate the forensics analyst's effort. For example, the developer can have the packer that unpacks the code into memory to delete code as soon as it is executed [37].

Anti-process - Anti-Process techniques are designed to mitigating the attempts by forensic analysts debugging of running processes. The technique changes the entry point from to a different one which foils the debugging effort [37].

Packers and protectors - Obfuscation and its subset, packing, are techniques used by malware developers to make static analysis more difficult for the forensics experts [22,37]. Obfuscation is a means of hiding or disguising code [22]. Packing uses compression and a wrapping program as a means of disguising the true purpose of program [22]. Even more

challenging for analysts and malware detection is recursive packaging which obfuscates code in multiple layers of recursive compression [36].

Metamorphic or Polymorphic - This type of malware is constructed in such a manner that is can re-engineer or recode itself [9,22]. This recoding can take place each time it propagates or is distributed through a network. This type of malware hinders the use of signature-based protection tools [9].

Prior Related Research

This section will look at some of the research that focus on the use of ontologies for malware analysis and the development of fuzzy logic and cognitive agents for use in detecting APT attacks.

In Mundie and McIntire's [39] research they sought to develop an ontology for Malware Analysis. Their work was motivated by four challenges in the business of malware analysis: (1) security teams and their customers were wasting time negotiating requirements because they did not "speak the same language;" (2) human resources departments couldn't hire the right malware analysts because they could not properly explain job requirements; (3) certification programs did not have a standardized way to assess the abilities of malware analysts; and (4) information sharing within the malware analysis community is impeded by a lack of shared foundation [39].

Their work employed six increasingly complex levels of knowledge representation [39]:

- Controlled Vocabulary – collection of preferred terms.
- Taxonomy - hierarchically related terms in a controlled vocabulary.
- Static Ontology – an ontology that describes static aspects of the world.
- Dynamic Ontology – an ontology hat describes changing aspects of the world.
- Intentional Ontology – a subjective ontology based on the motivation of agents.
- Meta-model – An ontology template that can generate ontologies by filling-in the parameters.

Their work yielded a vocabulary of approximately 270 malware analysis terms and a taxonomy outlined in World Wide Web Consortium's Web (W3C) Ontology Language (OWL). Mundie and McIntire built their initial ontology using the email archive of a malware analysis team. They also included various recognized textbooks in the malware analysis field and some Internet resources [39].

Huang, Loia, Lee, and Kao [40] research sought to apply fuzzy logic and ontologies for their application of inferring knowledge about malware, and designing an intelligent decision making system whose behavioral rules can be used to detect viruses and other malicious programs. As with Mundie and McIntire's research they employed OWL to build their malware behavior ontology [40].

To test the effectiveness of their decision making system they evaluated its performance against 30 "attendance records" from the National Center for High Performance and Computing (NCHC) malware repository. Their reported results and conclusion indicate that the employment of fuzzy logic and ontology was feasible and usable for a malware behavioral analysis system [40].

Meckl, Tecuci, Boicu, and Marcu [41] attempted to improve cyber defense against APTs using an operational semantic approach. The motivation for their work was to reduce false positives thus increasing the efficiency and reducing the costs associated with automated APT analysis. To achieve this, they are proposing developing collaborative cognitive agents that can apply updates based on new intelligence. In theory, the work of this present study could potentially inform this agent [41].

These and potentially other research efforts have some overlap with our current effort, but we contend that this does not invalidate our efforts. Instead, we hold that our research could augment these research efforts and others like them.

Defining and Evaluating Open Source Intelligence

With related research understood, we can start to explore the independent elements that will go into the present research. Therefore, the next three sections will address Open Source Intelligence (OSINT) which will be used to establish our ontology and as our data for text-mining and link analysis, which will be discussed in the following two sections.

Intelligence organizations and law enforcement at all levels of government use data and information found in open sources for decades [42,43]. Traditionally, OSINT was characterized by searching through publicly available sources of information to include books, journals, magazines, etc. [42,44]. Steele formalizes the definition of OSINT as "information that has been deliberately discovered, discriminated, distilled and disseminated to a select audience [43]."

Steele [43] notes that the change to OSINT is the result of three distinct trends (1) the proliferation of the Internet; (2) the consequence of the related "information explosion" of published useful knowledge which is experiencing an exponential growth; and (3) the availability of formerly restricted sources of information resulting from failed states and newer non-state threats. Best [45] acknowledges that the challenge with OSINT is not the collection of information but the filtering and distillation of the retrieved content into meaningful metadata that can be analyzed.

The Role of the Ontology

An ontology is designed to establish a common vocabulary and give practitioners the ability to easily share concepts [39]. The development of an ontology is further intended to streamline the process of information sharing and to avoid problems of misrepresentation or miscommunication resulting from parties using incompatible terminology [40]. Previous

research in the area of malware analysis and APT detection will be discussed in the section on Prior Related Research.

The Role of Link Analysis

Also referred to as relationship extraction, it is the identification of relationship between two units whether they are individuals, organizations, entities, etc. [45]. The relationship between them is derived from the context in which the reference is used. For example, "A is the child of B," "A is the CEO of B," "Company C merged with Company D," etc.

The research performed my Ben-Dov, et.al. [46] explores the use of link analysis and inference from that analysis can produce increased knowledge. Their efforts were an extension of prior done by Davies, and Swanson and Smalheiser. One of the biggest challenges to their research is that contemporary link-analysis tools operated on structured data [46]. They overcame the obstacle by processing text through information extraction or text mining as a precursor to the link analysis.

For their research, Ben-Dov, et.al. [46] used two different methods for text mining that the present research must consider. Co-occurrence links uses pattern matching to determine if target phrases exist within a sentence. However, the mechanism does not apply any semantics or syntactic logic to determine how the two terms relate to one another. The second method that they employed Semantic links work by connect noun phrases and verb identification with the application of linguistic and semantic constraints. They accomplished this by using Declarative Information Analysis Language developed at ClearForest Labs [46].

Mapping the Present Research Effort

With definitions, history, and a review of prior APT research complete, it is time to layout the approach that the authors intend to apply in the present endeavor. We will start by identifying our research question and hypotheses. Finally, we will lay out the planned methodology.

The Research Question and Hypotheses

The goal of the current research is to use open source intelligence pertaining to known APTs to establish an APT ontology and then employ the ontology and link analysis with the goal of increasing the amount of intelligence about individual APTs as aggregated from the whole knowledge base.

Based on the research goal outlined above, we intend to test the following hypotheses:

- **The Null Hypothesis (H_0):** Link analysis will provide the same amount of intelligence about APTs as what is currently available.
- **The Alternate Hypothesis (H_a):** Link analysis will improve the intelligence about APTs as what is currently available.

The Planned Methodology

The methodology consists of two phases. Figure 1 provides a graphical representation of the description that follows.

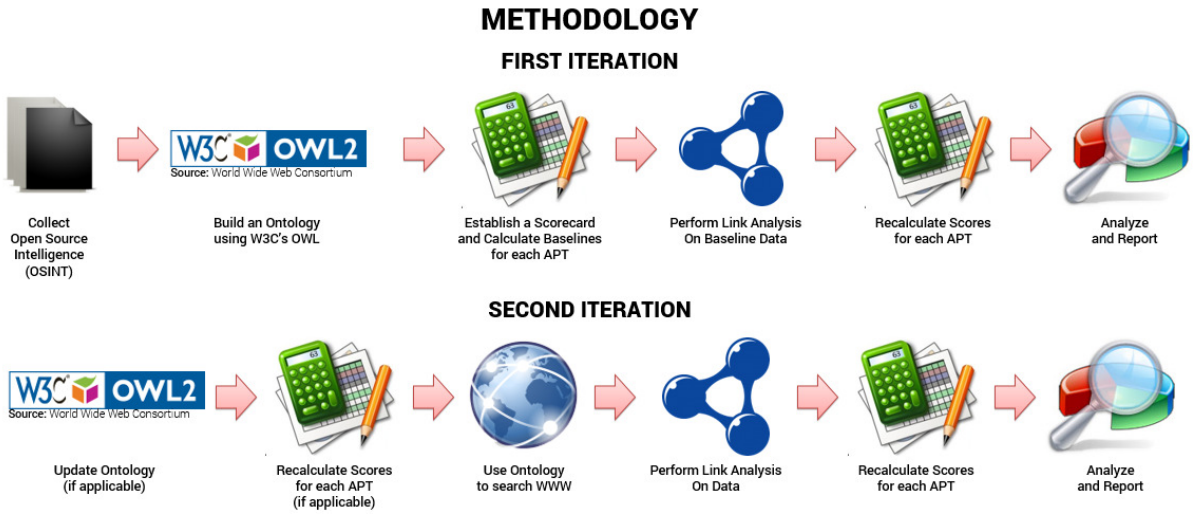

Figure 1: Sketch of the Planned Research Methodology

First Iteration

The first iteration will be completed in six steps starting with a defined group of OSINT to be used for the development of the initial ontology. The second step of this phase involved the development of an initial ontology. For this process, we intend to incorporate W3C's Ontology Language 2.0 (OWL2) and a yet to be determined software interface. W3C is an internationally organization for establishing open source standards including OWL2, HTML, CSS, and others.

The next step involves the development of a scorecard. The purpose of this scorecard to establish an objective means for quantifying how much is known about each particular APT attack. For example, if one of the scoring criteria is 'Known Command and Control (C2) Servers' and the OSINT reveals 3 different C2 servers a given APT communicates with the scorecard would reflect that. The scorecard will enable researchers to mathematically determine the success or failure of the link analysis process.

The fourth step in this phase involves information extraction and link analysis upon the initial OSINT used to create the ontology earlier in the research. The plan is to incorporate lessons learned from the Ben-Dov, et.al. research. Although, at the present time we have not settled on an application to perform the link analysis. We are still exploring open source and commercial link-analysis software packages before selecting the application we will employ.

The fifth step in the phase will be to recalculate scores for each APT attack accounting for any new information derived from the link analysis completed in step four. The final step of

the first phase is to compare the updated scores with the baseline scores and to report on the results.

Second Iteration

The first step of the second phase involve a review of the ontology and a determination if modifications or additions needs to be made to it. If the researchers determine modifications are needed, then scores for each of the APTs will be recalculated. If not, then the second step will be skipped.

The third step of this phase will consist of several rounds of internet searches for new intelligence based upon the ontology and key information already known. The newly collected information will undergo information extraction and link analysis during the fourth step of this iteration. Then, as in first phase, scores for the APTs will be recalculated based upon the latest knowledge and compared to the scores as they were before this phase's link analysis was performed.

Conclusion

Take away the hype and the buzzwords and APTs are still a real threat to governments and corporations around the world. The sophisticated attacks executed by organized, knowledgeable, skilled, and patient adversaries. What we presented in this paper is our plan for improving threat intelligence about individual APTs by employing text-mining and link analysis upon OSINT sources pertaining to a wide range of historical APT attacks.

References

[1] R. Bejtlich, TaoSecurity: What Is APT and What Does It Want?, (n.d.). http://taosecurity.blogspot.se/2010/01/what-is-apt-and-what-does-it-want.html (accessed April 27, 2016).

[2] M. Ask, P. Bloemerus, P. Bondarenko, A. Nordbø, J.E. Rekdal, J.E. Rekdal, D. Piatkivskyi, G. Piatkivskyi, J.E. Rekdal, Advanced Persistent Threat (APT) Beyond the Hype, Gjøvik, 2013.

[3] L. Arsene, The Anatomy of Advanced Persistent Threats - Dark Reading, (2015). http://www.darkreading.com/partner-perspectives/bitdefender/the-anatomy-of-advanced-persistent-threats/a/d-id/1319525 (accessed April 28, 2016).

[4] E. Cole, Advanced Persistent Threat: Understanding the Danger and How to Protect Your Organization, Syngress, Waltham, 2012.

[5] D. Bodeau, R. Graubart, W. Heinbockel, E. Laderman, Cyber Resiliency Engineering Aid – Cyber Resiliency Techniques: Potential Interactions and Effects Deborah Bodeau Approved By, Bedford, 2014.

[6] P.S. Radzikowski, CyberSecurity: Expanded Look at the APT Life Cycle and Mitigation, (2016). http://drshem.com/2016/02/11/cybersecurity-expanded-look-apt-

life-cycle-mitigation/ (accessed April 27, 2016).
[7] S. Chandran, H. P, P. Poornachandran, An Efficient Classification Model for Detecting Advanced Persistent Threat, in: 2015 International Conference on Advances in Computing, Communications and Informatics, IEEE, 2015: pp. 2001–2009.
[8] E.M. Hutchins, M.J. Cloppert, R.M. Amin, Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains, 2011. http://papers.rohanamin.com/wp-content/uploads/papers.rohanamin.com/2011/08/iciw2011.pdf$\$nhttp://www.lockheedmartin.com/content/dam/lockheed/data/corporate/documents/LM-White-Paper-Intel-Driven-Defense.pdf.
[9] V.S. Raj, R.M. Chezhian, M. Mrithulashri, Advanced Persistent Threats & Recent High Profile Cyber Threat Encounters, International Journal of Innovative Research in Computer and Communication Engineering. 2 (2014) 2412–2417.
[10] T. Armerding, SANS Digital Forensics and Incident Response Blog | Advanced Persistent Threats Can Be Beaten | SANS Institute, (2012). https://digital-forensics.sans.org/blog/2012/08/10/advanced-persistent-threats-can-be-beaten.
[11] M. Auty, Anatomy of an Advanced Persistent Threat, Network Security. 2015 (2015) 13–16. doi:10.1016/S1353-4858(15)30028-3.
[12] Mandiant, APT1 Exposing One of China's Cyber Espionage Units, (2013) 1–76. http://intelreport.mandiant.com/Mandiant_APT1_Report.pdf.
[13] R. Bejtlich, China's "Advanced Persistent Threat" to American Computer Networks, Hampton Roads International Security Quarterly. (2013) 16–19. doi:http://dx.doi.org/10.1108/17506200710779521.
[14] TrendLabs Security Intelligence BlogConnecting the APT Dots - TrendLabs Security Intelligence Blog, (n.d.). http://blog.trendmicro.com/trendlabs-security-intelligence/connecting-the-apt-dots-infographic/ (accessed April 30, 2016).
[15] Ponemon Institute, 2015 Cost of Data Breach Study: Global Analysis, (2015) 1–19.
[16] D. Bisson, The OPM Breach: Timeline of a Hack, Tripwire. (2015). http://www.tripwire.com/state-of-security/security-data-protection/cyber-security/the-opm-breach-timeline-of-a-hack/ (accessed April 30, 2016).
[17] Damballa, Advanced Persistent Threats (APT), (2010). https://www.damballa.com/downloads/r{_}pubs/advanced-persistent-threat.pdf.
[18] D. Kushner, The Real Story of Stuxnet, IEEE Spectrum. (2013) 48–53.
[19] United States. The White House, Executive Order 13636: Improving Critical Infrastructure Cybersecurity, Federal Register. 78 (2013) 1–8.
[20] K.L. Card, M.S. Rogers, Navy Cyber Power 2020, 2012.
[21] Websense, Advanced Persistent Threats and Other Advanced Attacks :, (2011) 1–14. http://www.websense.com/assets/white-papers/whitepaper-websense-advanced-persistent-threats-and-other-advanced-attacks-en.pdf (accessed March 13, 2016).
[22] M. Sikorski, A. Honig, Practical Malware Analysis: The Hands-on Guide to Dissecting Malicious Software, 2012.
[23] M.O. Saarinen, Developing a Grey Hat C2 and RAT for APT Security Training and Assessment, in: GreHack 2013, GreHack, Grenoble, 2013: pp. 12–24.
[24] D. Alperovitch, Revealed: Operation Shady RAT, White Paper. (2011) 1–14.
[25] McAfee, Global Energy Cyberattacks: "Night Dragon," (2011) 19. www.mcafee.com.
[26] RSA FraudAction Research Labs, Anatomy of an Attack - Speaking of Security - The

RSA Blog and Podcast, (2011). https://blogs.rsa.com/anatomy-of-an-attack/ (accessed April 30, 2016).
[27] W. Ashford, RSA hit by advanced persistent threat attacks, ComputerWeekly.com. (2011). http://www.computerweekly.com/news/1280095471/RSA-hit-by-advanced-persistent-threat-attacks (accessed April 30, 2016).
[28] A. Coviello, Open Letter to RSA Customers, (n.d.). https://www.sec.gov/Archives/edgar/data/790070/000119312511070159/dex991.htm (accessed April 30, 2016).
[29] Kaspersky Lab Global Research and Analysis Team (GREAT), The Icefog APT: A Tale of Cloak and Three Daggers, (2013) 1–68. http://kasperskycontenthub.com/wp-content/uploads/sites/43/vlpdfs/icefog.pdf.
[30] Kaspersky Lab Global Research and Analysis Team (GREAT), Securelist | The Icefog APT: A Tale of Cloak and Three Daggers - Securelist, Securelist. (2013). https://securelist.com/blog/research/57331/the-icefog-apt-a-tale-of-cloak-and-three-daggers/ (accessed April 29, 2016).
[31] N. Perlroth, Chinese Hackers Infiltrate New York Times Computers - The New York Times, New York Times. (2013). http://www.nytimes.com/2013/01/31/technology/chinese-hackers-infiltrate-new-york-times-computers.html?_r=0 (accessed April 29, 2016).
[32] Mandiant, M-Trends: Beyond the Breach, (2014) 24.
[33] S. Naval, V. Laxmi, M. Rajarajan, M.S. Gaur, M. Conti, Employing Program Semantics for Malware Detection, 10 (2014) 2591–2604. doi:10.1109/TIFS.2015.2469253.
[34] Mandiant, M-Trends: The Advanced Persistent Threat, (2010) 32. doi:10.1049/etr.2014.0025.
[35] N. Villeneuve, J. Bennett, Detecting APT Activity with Network Traffic Analysis, Cupertino, 2012. http://www.trendmicro.pl/cloud-content/us/pdfs/security-intelligence/white-papers/wp-detecting-apt-activity-with-network-traffic-analysis.pdf.
[36] M. Egele, T. Scholte, E. Kirda, C. Kruegel, A survey on automated dynamic malware-analysis techniques and tools, ACM Computing Surveys. 44 (2012) 1–42. doi:10.1145/2089125.2089126.
[37] M. Brand, C. Valli, A. Woodward, Malware Forensics: Discovery of the Intent of Deception, Journal of Digital Forensics, Security and Law. 5 (2010) 31–42. http://ojs.jdfsl.org/index.php/jdfsl/article/view/142.
[38] A.F. Shosha, J.I. James, A. Hannaway, C.-C. Liu, P. Gladyshev, Digital Forensics and Cyber Crime, in: M. Rogers, K.C. Seigfried-Spellar (Eds.), Digital Forensics and Cyber Crime, Springer, Lafayette, 2012: pp. 66–80.
[39] D.A. Mundie, D.M. McIntire, An Ontology for Malware Analysis, Proceedings - 2013 International Conference on Availability, Reliability and Security, ARES 2013. (2013) 556–558. doi:10.1109/ARES.2013.73.
[40] H. De Huang, G. Acampora, V. Loia, C.S. Lee, H.Y. Kao, Applying FML and Fuzzy Ontologies to malware behavioural analysis, IEEE International Conference on Fuzzy Systems. (2011) 2018–2025. doi:10.1109/FUZZY.2011.6007716.
[41] S. Meckl, G. Tecuci, M. Boicu, D. Marcu, Towards an Operational Semantic Theory of Cyber Defense Against Advanced Persistent Threats, in: STIDS 2015 Proceedings, 2015: pp. 58–65.

[42] C. Fleisher, Using open source data in developing competitive and marketing intelligence, European Journal of Marketing. 42 (2008) 852–866. doi:10.1108/03090560810877196.
[43] R.D. Steele, Open source intelligence, in: L.K. Johnson (Ed.), Handbook of Intelligence Studies, Routledge, New York, 2007: pp. 129–147.
[44] H.P. Burwell, Online Competitive Intelligence: Increase Your Profits Using Cyber-Intelligence, Facts on Demand Press, 1999.
[45] C. Best, Challenges in open source intelligence, Proceedings - 2011 European Intelligence and Security Informatics Conference, EISIC 2011. (2011) 58–62. doi:10.1109/EISIC.2011.41.
[46] M. Ben-Dov, W. Wu, P.A. Cairns, A. Place, Improving Knowledge Discovery By Combining Text-Mining And Link-Analysis Techniques Semantics links, Foundations. (2004) 1–7.

Biographies

COREY HOLZER is currently a PhD Candidate of Computer and Information Technology at Purdue University. He earned his B.A. degree from St. John's University, NY; an M.A. degree in Government and Politics from St. John's University, NY; an M.S. in Networking Communications Management from Keller Graduate School of Management, IL; and an M.B.A. from Keller Graduate School of Management, IL. He also holds industry certifications including CISSP, Security+, and CEH. Capt. Holzer is an Officer in the United States Army. His research interests include Information Security, Cyber Security, Forensics, Risk Analysis, Cyber Resiliency, and Information Assurance Ethics. Capt. Holzer may be reached at cholzer@purdue.edu.

J. ERIC DIETZ, PhD, PE earned Chemical Engineering BS and MS degrees from Rose-Hulman Institute of Technology and PhD from Purdue. Now director of the Purdue Homeland Security Institute and professor of Computer and Information Technology, Eric formerly served on the Cabinet of Indiana's Governor as the founding executive director for the Indiana Department of Homeland Security. Dr. Dietz can be reached at jedietz@purdue.edu.

BAIJIAN YANG is currently an Associate Professor of Computer and Information Technology at Purdue University. He earned his B.S. and M.S. degrees in Automation from Tsinghua University, Beijing, China; and Ph.D. in Computer Science from Michigan State University, East Lansing, MI. Dr. Yang is a member of IEEE Cybersecurity Steering Committee and a board member of ATMAE. He holds a few industry certifications, such as MCSE, CISSP and Six Sigma Black Belt. His research interests include data-driven security analyses, forensics, cloud computing, big data, and cybersecurity education. Dr. Yang may be reached at byang@purdue.edu.