NUST Journal of Engineering Sciences, Vol 9, No 2, 2016, pp 68-75

Ontology Generation of Advanced Persistent Threats and their Automated Analysis

Zafar Iqbal[1], Zahid Anwar[1,2]*
[1]School of Electrical Engineering and Computer Science (SEECS), National University of Sciences and Technology (NUST), Islamabad, Pakistan
[2]Fontbonne University, St. Louis, Missouri, USA
*zahid.anwar@seecs.nust.edu.pk, zanwar@fontbonne.edu
Submitted: 20 September 2016 Accepted: 15 November 2016

Abstract

Advance Persistent Threats (APTs) are a continuous hacking process during which the perpetrator changes signatures and uses different malware to launch an attack. For these reasons most of the time APTs remain undetected by the conventional IDSs. Ironically a large amount of data is available regarding APTs in literature and online repositories. However, due to high adaptivity and large volume of data, analyzing information about APT incidents is challenging for security analysts. Several security models have been proposed for analysis and understanding of the APTs. In this regard, two recent approaches: Cyber Kill Chain (CKC) and Pyramid of Pain (POP) are noteworthy. CKC is an attacker model while POP is a defender model. If these approaches are combined into a suitable defense framework, then these can be used as an early warning system against APTs. The contributions of this paper are two-fold. The first is development of CKC and POP's standalone ontologies, identifying relationships between these and developing a common ontology of APTs. Secondly, we propose a novel framework "**A**PTs **A**nalysis and **C**lassification **S**ystem – **A2CS**" which uses semantic rules for automatic analysis of APTs such as identification of their missing artifacts and inferencing of the Tactics, Techniques and Procedures being employed.

Keywords: Advanced Persistent Threat, Cyber Kill Chain, Pyramid of Pain, Point of Sale, Tactics Techniques and Training and Procedures.

Introduction

Primarily, Intrusion Detection Systems (IDS) are signature based. These systems consider atomic and computed indicators of previously known attacks for detection of imminent attack. Statistical anomaly based IDSs (SIDS) are designed to analyze the behavior of the network traffic against a baseline profile. The baseline profile is a detailed description of a normal network behavior, usually enumerated by the administrator. SIDSs classify all normal and abnormal behavior on the network with reference to the baseline behavior. A poorly defined baseline profile reduces the detection ability of an IDS system. In rule based IDSs, intrusion is detected by perceiving events on the network. Rules are applied to decide whether an activity is an intrusion or not. The malware detection capability of such systems greatly depend on the rules. In these systems, defining the correlation rules is the biggest challenge. Furthermore, analysts need to consider numerous logs because they don't have an idea, which log will be relevant. To keep track all of this requires considerable expertise. Customized protocols used by the perpetrator makes writing rules a difficult job. With all of these challenges, manual writing of rules is not practically feasible.

Security Information and Event Management System (SIEM) performs real-time analysis and correlation of events generated by the network applications and hardware. These tools provide fast search based on big-data indexing techniques. Such systems are only useful, if the security analyst knows what to search.

According to a recent security survey [1], security incidents have raised to 42.8M around the world and these incidents are risen 66% each year since 2009. The average reported loss was up to 34% in 2014 as compared to 2013 and 86% of the cyber-attacks involved by these losses were launched by nation states. Some governments have made cyber-attacks campaign part of their military strategy and have built their own cyber armies. According to [2], cybercriminals are trying their best to attack individuals, organizations and different states. A majority of these attacks are targeting government, financial, healthcare and marketing industries. APTs have diverse goals: some APTs are interested in financial gains e.g. Zeus and Carbanak, some in political gains and sabotage e.g. Naikon and Stuxnet APTs and other requires personnel information e.g. PoSeidon and BlackPOS.

A massive volume of data about APTs is available on different security webs and blogs but it is mostly unstructured. A few efforts (Open IOC, STIX) are made towards the standardization of the cyber-threat data by the government but are slow in adaption. Regardless whether the data is structured or unstructured, the major part of the available data is regarding atomic and computed indicators (*IPs, Domain Names and Hash Values)* while the data related to higher level artifacts (*File name, Registry* entries, *Protocols used*, Obfuscation methods, and *TTPs*) which is more related to decisions is generally missing. A perpetrator can change the atomic indicators with little effort but the higher level artifacts are hard to change because perpetrator invested great time & money during development of these artifacts.

Most of the time, the APTs related data is distributed; available on different webs and blogs in bits and pieces but there is no standard way to access this data. For these reasons, it has become a great challenge for security analysts, to collect such distributed data, manually process it, identify and extract the relevant information and then analyzed

different APTs. Ontology based systems demonstrate shared understanding of the information about the concepts within a domain and provide the reasoning capability for automatic analysis of the information. In the recent past, two models related to cyber-attacks are proposed such as the CKC [3] and the POP [4]. The CKC guides an analyst regarding how a perpetrator uses different phases to launch an APT and guides the security analyst regarding how signatures and artifacts available at different attack levels can be used to defend their network from APTs.

Heretofore, the CKC and POP are theoretical models and are not used in real IDSs. These models are complementary to each other and cyber-attack picture can't be seen holistically without any of these models. Due to these reasons, we developed a combined ontology of both the models. We have selected real examples of Point of Sale (POS) APTs and scanned different security vendor's webs and blogs and found significant amount of CKC and POP information regarding the POS APTs. Our proposed framework A2CS stores this information in the form of an ontology, which helps the A2CS for identification of the missing artifacts and inferencing of the high level TTPs with help of the low level artifacts.

The paper is organized as follows. The technical background of the paper is sketched in section 2. Related work is presented in Section 3. In section 4, we presented a combined ontology of the CKC and POP. The proposed methodology is presented in section 5 while conclusion and future work is presented in section 6.

Background

We don't assume that users have prior knowledge of *Ontology, Pyramid of Pain and Cyber Kill Chain*. For ease of their reading and better understanding, we are briefly discussing these concepts in this section. References are provided for further reading.

Ontology

Ontology is a graph model which represents domain knowledge, by which developers and machines can exchange domain information with each other and with others experts. Since last few years, researchers have focused on how an ontology and linked knowledgebase could be constructed from structured and unstructured data sources and how to infer an attack using knowledgebase.

Cyber Kill Chain (CKC)

The Kill Chain is a military concept [5] used for structuring an attack. It is a stage based model used to describe different phases of an attack. Recently, the authors in [3] and (An American Global Aerospace, Defense, Security and Advanced Tech Company) have used this concept in Information Security (IS) domain to combat against advanced threats. According to authors, a malware campaign may be divided into seven different phases, as shown in Fig 1. In *Reconnaissance* phase, the perpetrator collects information regarding the target through web, social media and using other publically available information.

Fig 1: Cyber Kill Chain [3]

Then in *Weaponization* phase the perpetrator analyzes the collected data of the *Reconnaissance phase* and decides: what attack method should be used; who should be targeted in an organization and which OS and technologies should be targeted. In the *Delivery* phase of the CKC, the perpetrator sends the malware payload to the target. Once delivered, malware *exploits* the vulnerabilities at the target machine to execute the perpetrator code. Then the malware is *installed* on the target machine and it establishes a communication channel with adversary *Command and Control (C2)*. Finally the perpetrator collects the desired data during *Exfiltration* phase, encrypt it and then send it to the *C2*.

Pyramid of Pain (POP)

The author presented the concept of POP in [6]. The POP is not the replacement of the CKC but it is the counterpart of it. It is a framework for hunting the cyber threats. The POP model can be seen in Fig. 2. This model describes different indicators which can be used for the detection of the advanced threats. *Hash Values* are the base part of the pyramid and these are the unique reference to a specific malware files used in intrusion. *IP Addresses* are the fundamental indicators and are the widest part of the POP. These are used by the C2 for monitoring of the

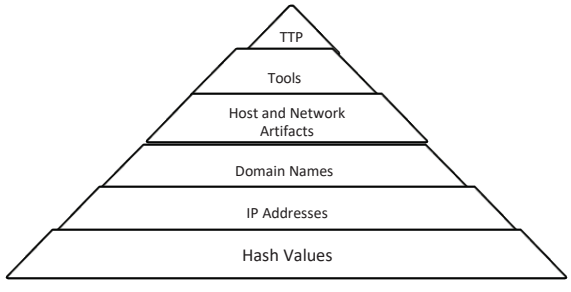

Fig. 2: Pyramid of Pain [6]

target machine and to acquire the desired information from it. *Domain Names* are used as a rendezvous points between the targeted machine and the perpetrator's C2 servers. In the middle of the pyramid, there are *Host and Network Artifacts*. *Host artifacts* are the entries made by the perpetrators on the host machine. These entries may include: *Registry entries, Files or Folders creation* and *the malicious process* on the infected machine. The *Network Artifacts* are the noticeable activities of perpetrator on the network. These includes: *URL patterns and C2 information*. *Tools* are the software used by the perpetrator during malware campaign for accomplishment of an attack. These tools includes *utilities*: to deliver malware payload; to create backdoors; password cracker and other host based adversary software. In this level of pyramid, analysts try to detect the artifacts of the adversary tools. Once analysts get information about these tools then they become able to protect themselves from adversary attacks. At the peak of the POP there is *Tactics, Techniques and Procedures (TTP)* phase. *TTPs* are very

important indicators. These guide the analyst how the perpetrator will accomplishes mission.

Related Work

Due to novelty of our work we could find relatively less amount of associated literature in this domain. However, our work benefits from: surveys; advance attacks; techniques and study of different analysis models of APTs. Our literature review is organized as follows: First, we outline various efforts that have been carried out for expressing unstructured cyber data into structured and machine understandable format. Subsequently, we present an abstract of the earlier models which have been made for study, analysis and correlations of APTs. Afterward; we present an overview, how perpetrator uses social engineering techniques for reconnaissance and delivery of APTs. Then we outline different contributions made for automatic detection of different attacking techniques. Finally, we give an overview of the earlier attempts to develop ontologies for cyber-attacks detection and resilience.

Structuring and Integration of Security Concepts

We have seen three categories of malware reports on the web: incident reports, blog reports and deep web reports. These reports are an important information sources about vulnerabilities and cyber-attacks. Significant amount of information is available as unstructured text at different security webs (Kaspersky, IBM), Blogs (Metasploit and Krebs) and chat rooms. The governments are pushing for sharing cyber data and multiple efforts are carrying on for expressing this non structured information into structured and machine understandable format. IBM X-Force and National Vulnerability Database (NVD) provide an XML feed that gives information regarding cyber-attacks and vulnerabilities with diverse degree of details. There exists multiple standards of threat information exchange like: CIF, IODF by IBM, CRITS by Community, OPEN IOC, STIX, taxi and Cybox. Although different firms and governments are trying to bring this textual information into structured form but malware's information regarding CKC phases and POP's level is very limited.

Analysis Models for APTs

With the ever increasing number of data breaches due to cyber-attacks, timely diagnosis of attack vectors is of paramount importance. With the onslaught of new APTs that share behavioral signatures or leverage the same exploit kits, attack detection process can become nearly impossible. Different efforts are made for study and analysis of cyber threat domain such as: CKC by Lockheed Martin in [3] and [7] and an ontology by MITRE [8]. The authors in [3] present the concept of cyber-attack Kill Chain. In [6], the author presents the concept of POP. Furthermore in [9] author gives an idea of "Integration of CKC and POP". In [10], MITRE presents ATT&K model which classify APT's TTPs into nine different classes. In [11], the authors present a framework to handle APTs attack by using Intrusion Kill Chain (IKC) which is similar to Lockheed Martin KC. The researchers in [12] classify APTs attack into five different phases from malware delivery to data exfiltration. They do not discuss the *Reconnaissance* and *Weaponization* phases of APTs. In [13], the authors present the analysis of different attacks and on basis of these they describe an attack process model. The model has eight different steps and some of these are similar to CKC. The authors in [14] present a five dimension (*Target, Carrier, Vulnerability, Privilege Escalation and Firing Source*) computer attack taxonomy. All of these works became motivation for us to develop combine ontology of CKC and POP and automatic extraction of APTs related data from *Web text, Security webs and Blogs*.

How APTs Exploits Humans

Today perpetrators are widely using social engineering techniques: *Emails, Facebook, LinkedIn, Blogs* and other sources for *Reconnaissance* and *Delivery* phases of cyber-attack. In [15], the authors present the taxonomy of social engineering. The social media is widely used for collection of target information and delivery of the malware. In [16], the author presents different techniques, which can be used to send malicious codes to victim machines. Spear phishing and web-based click hijacking are mostly used for malware delivery. The authors in [17] describe that *Reconnaissance and Delivery* phases of APTs are successful because of human manipulation. They highlight some of the famous examples of APTs which uses human manipulation for delivery of the APTs such as: Stuxnet uses USBs; Dugu uses infected MS Word files via email; Red October uses infected MS Word and Excel documents via spear phishing; Operation Aurora uses infected web sites; Operation Shady Rat uses infected MS Word, Excel and Pdf documents via spear phishing and RSA attacks uses MS Excel documents attachment with in spear phishing emails.

Automatic Analysis of Common Attack Techniques

Research shows that tactics and techniques in multiple APTs remains same or may be used with small changes and if analysts know the general technique of the APTs then they can catch multiple APTs easily. The McAfee in [12] outlines that during analysis of single C2 (used by Operation Shady Rat) their researchers have found single organization which hacked almost 71 companies of 31 diverse industries of different countries. In [18], the researchers developed a technique to identify the patterns in DNS to infer whether an attack is generated by an algorithm or by some human beings. The authors in [19] describe the idea that APTs can obfuscate more intelligently. The manual process of sifting through tons of log data to pinpoint these tactics and techniques is a challenging job. Efforts are required for automatic detection of these techniques.

Ontology Design and Resilience

Few work has been made on ontologies and is generally based on representation of the cyber-attacks attributes in a taxonomical structure. In [20], the authors suggest countermeasures based on the cost of the metrics. The researchers in [21] describe nine different metrics like: *Input Validation, Authentication, Authorization, Configuration*

and Installation, Sensitive Data, Session Management, Cryptography, Exception Management and Auditing and Logging. They suggested a metric based model for malware classification. The paper [22] is an extension of author's previous work [21].The research is mostly focused on extracted metrics, attacks against these and counter measures to prevent these attacks. In [23], the researchers present the idea of extracting security concepts from text, compare these with monitoring sensors logs and then generate security alerts with the help of reasoner. To the best of our knowledge, the idea of using heterogeneous sources (txt and IDS logs) is a worthy solution, although ontology (taken from [24]) is a very basic and does not give holistic view of an attack. The authors in [25] present a framework for extraction of vulnerability and cyber-attack related information from web text and then compare these with Wikitology. A model is proposed in [26], which takes unstructured text as an input, automatically extract the entities and concepts from it and then passes these to DBpedia spotlight. At DBpedia these concepts matched and assigned corresponding class value. The authors in [27] present a Maximum Entropy Model for automatic labeling of text. All [25], [26] and [27] works are worthy contribution for point of data retrieval and these efforts are complementary for our work. A cyber-security ontology named "CRATELO" is introduce in [28]. It is a logical ontology but it is very basic. The research moves around the spatial and temporal part of an attack. It does not discuss the host and network artifacts of the APTs. The authors introduce concept of trust measurement in [29], according to author any value of delay which is out of the acceptable network delay range, will be unreliable. In [30], the researchers develop an ontology which semantically analyzes HTTP traffic and detect cyber-attacks. The work seems worthy contribution but it is limited to the application layer only. The authors in [31] present a unified ontology which integrates different cybersecurity systems and standards. They developed a prototype system which extract entities from NVD and map them to their parallel entities from DBpedia. This work seems praiseworthy contribution but it is limited to integration of ontologies and does not discuss behaviors of the malware.

Combined Ontological Model Of CKC and POP

In this paper, we propose a combined ontological model of CKC and POP which can be seen in Fig. 3. The combined ontology is quite rich and due to space limitations, we have shown important classes only. In our proposed ontology, we have developed 45 classes, 44 objects and 10 data properties. In the combined ontology blue circles depicts entities of CKC, orange circles are associated with POP and green entities are common.

Proposed Methodology – A2CS

The A2CS architecture can be seen in Figure 4 and in the remaining part of this section we will provide details of its various modules by using two APTs namely *JackPOS* and *BackOff*. These two are selected from large family of POS APTs [32] which comprises of *Reedum, Fsyna,*

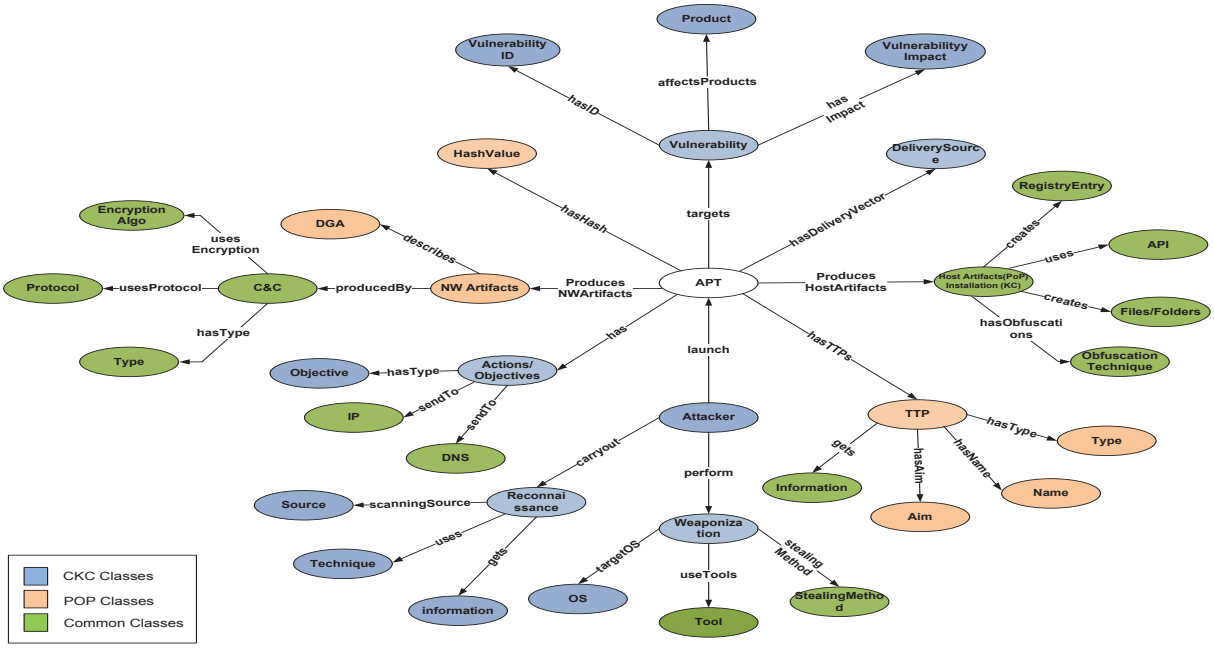

Fig. 3: Combined Ontology of CKC and POP

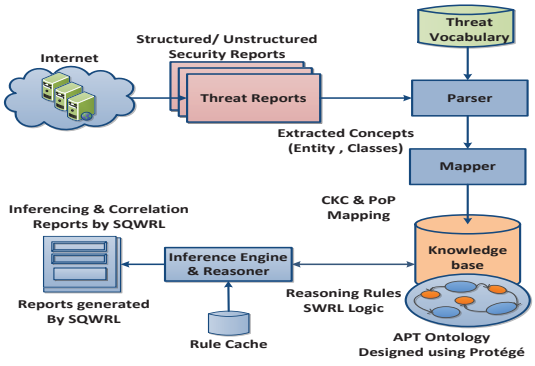

Fig. 4: A2CS Architecture

Dexter, Treasure hunt, Posfind, Alina, Poseidon, JackPOS, and BackOff. Our proposed system fetches web reports from the Internet and forwards these to the entity and concept *Parser* module. The *Parser* parses the data and extracts the entities and concepts. Next, the *Mapper* module correlates these extracted concepts with different phases of CKC and POP. As the example shown in Fig. 5.

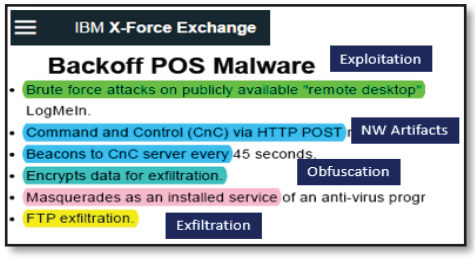

Fig. 5: Concept Extraction and Mapping

The outputs of the *Mapper* module are as follows:

- **Install/ Host Artifacts:** These artifacts are: *Registry entries and Files/ Folder* name e.g. during installation phase the JackPOS creates *"%Temp%\svchost.exe, java.exe, javaw.exe, javcpl.exe"* and the BackOff creates "*javaw.txt, Log.txt, Local.dat, winserv.exe"*.
- **Network Artifacts:** These are *C2 / Domain Name*. In this phase, both the malware are using *HTTP protocol and hard coded domain names* to communicate with C2.
- **TTPs/ Host Artifacts**: The BackOff malware uses both *Memory Scraping and Keystroke logging* techniques for data stealing while JackPOS uses *Memory Scraping* technique only.

Then *Mapper* feeds this extracted information into the knowledgebase. Next the reasoner module executes the rules over the knowledgebase. The next section will give details of the reasoning module.

Analysis via Reasoning

In our research we have employed different methods for analysis of APTs such as Risk analysis, Dependency analysis, Complexity analysis, Common Artifacts analysis, TTPs analysis and Time analysis. In this paper we are presenting two of these.

Identification of Missing Artifacts

As a result of our studies, we have observed that higher level artifacts of APTs are generally missing. We have introduced two types of techniques for identification of these missing artifacts.

Using the first technique, A2CS fetches information regarding different aspects of the APT from heterogeneous sources and combines them together in the ontology knowledgebase. For example in our case of information retrieval regarding the Backoff APT, concerning Host artifacts was retrieved from the Symantec portal whereas Network artifacts were extracted from IBM X-force, as shown in Fig. 6. This is important because threat sources usually specialize in particular aspects of threat reporting.

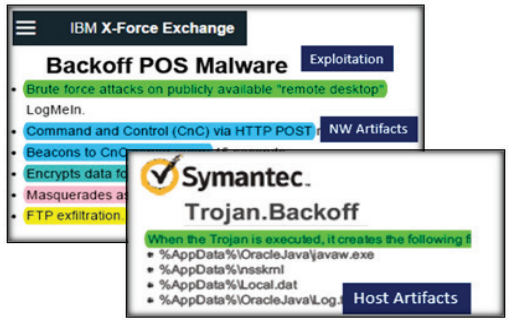

Fig. 6: Identification of Missing Artifacts

The second technique concerns the augmentation and enrichment of information about an incomplete APT from information about known or previously studied APTs of the same family. For example, JackPOS is a recent successor of BackOff and is therefore not as well studied as the latter. Our knowledgebase already consisted of information regarding BackOff APT's stealing methods and affected device. When the reasoning module correlated the artifacts of both, it concluded that since both are attacking same domain i.e. the Retailer Industry and directly affecting the terminal therefore JackPOS may be employing a similar Stealing_Method as used by the BackOff.

We have developed a number of queries for identification of missing artifacts in Semantic Query-Enhanced Web Rule Language (SQWRL), sample of these queries are given in Fig. 7.

```
CSecQry_1:  Attacker(?AT)  ^  APT(?AP)  ^
  launch(?AT,            ?AP)            ^
  producesHostArtifacts(?AP,      ?HA)     ^
  createFile(?HA,  ?CF)  ->  sqwrl:select(?AP,
  ?CF) ^ sqwrl:orderBy(?CF)

CSecQry_2:    Attacker(?AT)  ^  APT(?AP)  ^
  Weaponization(?WP)  ^  launch(?AT,  ?AP)  ^
  Perform(?AT, ?WP) ^ usesStealingMethod(?WP,
  ?SM)   ->   sqwrl:select(?AP,   ?SM)   ^
  sqwrl:orderBy(?AP)
```

Fig. 7: SQWRL Queries for APTs Correlation

The CSecQry_1 correlates files and folders and identify the common. Similarly, the CSecQry_2 is designed for finding

the information regarding stealing methods used by the APTs.

The correlation of the JackPOS and BackOff APTs generated by our proposed system is shown in Fig. 8. Doted lines indicate partially matched artifacts while fully matched artifacts are presented by solid lines. The correlation results are summarized in Fig. 9. These results indicate that most of the phases such as: *Weaponization, Host Artifacts, Network Artifacts and TTPs* are common in *JackPOS* and *BackOff*. The results demonstrate that both the APTs have 53% artifacts in common. On the bases of these results, the A2CS declares that both the APTs are belong to same family. The main difference between the APTs is in their *Delivery* phase i.e. the *JackPOS* focused more on Delivery phase than BackOff. If an analyst wants to block these APTs then he should focus on deploying controls to mitigate their Delivery phase.

Inferencing of the Possible TTPs

In cyber-attack analysis, the role of the TTPs is to identify individual patterns of behaviors. Identifying the behaviors allows identification and characterization of general behavior of an attacker. If an organization can block the general APT behavior, then he can cause much more pain to the attacker. If data about low level indicator is available in knowledgebase then A2CS on the basis of ontological design and inferencing rules can predict the TTPs. We have developed a number of SWRL rules for inferencing of the TTPs and some of these rules can be seen in Fig. 10. ***Rule-1*** infers that if target exploits "*Remote_Desktop_Login* vulnerability then the *Delivery* method of the malware will be "M*anual planting"*.

CKC and POP Phases & Levels	Infostealer JackPOS	Mapping	Infostealer Backoff
	Properties		Properties
Reconnaissance	--		--
Weaponization Stealing Method Targeted OS Tool Obfuscation	• Memory Scrapping • Window (All Versions) • Loader • Compiled Autolt Script • Code Obfuscated in java Bin		• Memory Scrapping, KeyLogging • Window (All Versions) • Proxy Server • Masquerades as an installed service
Delivery	• Distributed by Download • Phishing/ Spear Phishing Email		• Manual Planting
Exploitation	• Application Vulnerability • Fake Java Update		• Application Vulnerability • Remote Desktop and Brute force to login
Installation / Host Artifacts Folder Creation Files Creation Registry Entries API/DLL	• %APPDAT% Folder • %Temp%svchost.exe java.exe, javaw.exe, javcpl • Registry Entries • HttpOpenRequestW / HttpSendRequestW		• %APPDAT% Folder • javaw.txt, Log.txt, nsskrnl, Local.dat, winserv.exe • Registry Entries • CopyFileA / WinExec
CnC / Network Artifacts DN Algo / Hardocded Protocol Encryption	• DN - Hardcoded • HTTP Post • Encoding		• DN - Hardcoded • HTTP Post • Encoding • RC4 Encryption
Exfiltration	• Data Exfiltration • URL/DN • IPs		• Data Exfiltration • URL/DN • IPs
Hash	• Multiple Hashes		• Multiple Hashes
TTPs Targeted OS System Advantages	• Windows • Computer & Embedded • Credit Card & PII Info • Economic, Persecution and Theft of PII		• Windows • Computer & Embedded • Credit Card & PII Info • Economic, Persecution and Theft of PII

Fig. 8: APTs Comparison Generated by SQWRL

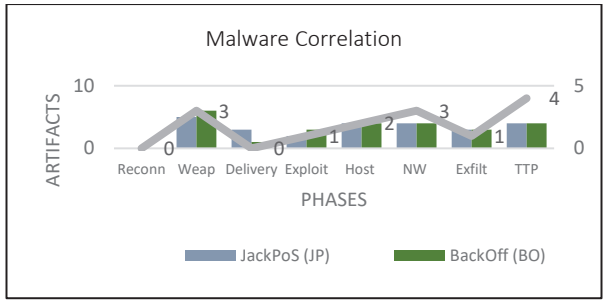

Fig. 9 : Correlation Results

Similarly, **Rule-2** describes that "RAM Scrapping" technique is used and APT belongs to POS family then the aim of the perpetrator will be to steal "Credit card and PII". **Rule-3** describes that if in an attack "RAM Scrapping" technique and "Browser" is used then the perpetrator will be interested in stealing "Banking_Credentials and *PII"*. The inferencing results are very meaningful, which indicate that if someone belongs from an organization which deals with credit card or online accounts then he must be careful about these APTs and try to safeguard the system from these information stealing techniques.

```
Rule_1: Attacker(?AT) ^ APT(?AP) ^ TTP(?TT) ^
  Vulnerability(?VUL)   ^   Delivery(?DV)   ^
  launch(?AT,  ?AP)  ^  hasTTP(?AP,  ?TT)  ^
  hasDeliveryVector(?AP,        ?DV)        ^
  targetVulnerability(?AP,      ?VUL)       ^
  hasVulType(?VUL, Remote_Desktop_Login)    -
  > hasSource(?DV, Manual_Planting)

Rule_2: launch(?AT, ?AP) ^ hasTTP(?AP, ?TT) ^
  Perform(?AT,             ?WP)             ^
  belongsTo(?AP,                POS_Family)^
  usesStealingMethod(?WP,      Ram_Scrapping)
  ->  hasAim(?TT, Credit_Card_and_PII)

Rule_3: launch(?AT, ?AP) ^ hasTTP(?AP, ?TT) ^
  Perform(?AT,  ?WP)  ^  uses(?AP,  Browser)
  ^usesStealingMethod(?WP, Ram_Scrapping)  ->
  hasAim(?TT,Banking_Crendentialand_PII)
```

Fig. 10: SWRL Rules

Conclusion and Future Work

The Pyramid of Pain and Cyber Kill Chain are emerging and promising models for network defense. These models are complementary for each other and cyber-attack picture can't be seen exclusively without any of these model. To best of our knowledge both of these models are theoretical and previously no one has developed a combine ontology of these. Due to these reasons, we developed a combined ontology of both the models for identification of missing artifacts and inferencing of TTPs. We tested our proposed system "A2CS" using data from real world APTs and found that a large percentage of APTs have several behaviors in common. Future work will involve developing an IDS of our proposed "A2CS" system.

REFERNCES

1. PriceWaterhouseCoopers, "The Global State of Information Security Survey," 2015.
2. ISACA, "Advanced Persistent Threat Awareness," 2015.
3. E.M. Hutchins, et al., "Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains," *Leading Issues in Information Warfare & Security Research*, vol. 1, 2011, pp. 80.
4. D. Bianco, "The Pyramid of pain," 2013.
5. A.J. Hebert, "Compressing the kill chain," *Air Force Magazine*, vol. 86, no. 3, 2003, pp. 50-55.
6. SANS, "Killing Advanced Threats in Their Tasks - An Intellignece Approach to Attack Prevention," 2014.
7. L. Obrst, et al., "Developing an Ontology of the Cyber Security Domain," *Proc. STIDS*, 2012, pp. 49-56.
8. D.Bionic, "What Do You Get When You Cross a Pyramid With A Chain," 2013.
9. MITRE, "Adversarial Tactics, Techniques, and Common Knowledge," 2015.
10. P. Bhatt, et al., "Towards a framework to detect multi-stage advanced persistent threats attacks," *Proc. Service Oriented System Engineering (SOSE), 2014 IEEE 8th International Symposium on*, IEEE, 2014, pp. 390-395.
11. McAfee, "Combating Advanced Persistent Threats," 2011.
12. M.S. Gadelrab, et al., "Defining categories to select representative attack test-cases," *Proc. Proceedings of the 2007 ACM workshop on Quality of protection*, ACM, 2007, pp. 40-42.
13. M.S. Geramiparvar and N. Modiri, "An Approach to Counteracting the Common Cyber-attacks According to the Metric-Based Model," *International Journal of Computer Science and Network Security (IJCSNS)*, vol. 16, no. 1, 2016, pp. 81.
14. K. Krombholz, et al., "Advanced social engineering attacks," *Journal of Information Security and applications*, vol. 22, 2015, pp. 113-122.
15. P. Institute, "State of Endpoint Risk," 2013.
16. M. Bere, et al., "How Advanced Persistent Threats Exploit Humans," *International Journal of Computer Science Issues (IJCSI)*, vol. 12, no. 6, 2015, pp. 170.
17. S. Yadav, et al., "Detecting algorithmically generated malicious domain names," *Proc. Proceedings of the 10th ACM SIGCOMM conference on Internet measurement*, ACM, 2010, pp. 48-61.
18. I. You and K. Yim, "Malware obfuscation techniques: A brief survey," *Proc. Broadband, Wireless Computing, Communication and Applications (BWCCA), 2010 International Conference on*, IEEE, 2010, pp. 297-300.
19. I. Kotenko, et al., "The ontology of metrics for security evaluation and decision support in SIEM systems," *Proc. Availability, Reliability and Security (ARES), 2013 Eighth International Conference on*, IEEE, 2013, pp. 638-645.

20. M.S. Geramiparvar and N. Modiri, "Presenting a Metric-Based Model for Malware Detection and Classification," *International Journal of Computer and Information Technologies*, vol. 2, no. 4, 2014, pp. 536-539.
21. S. More, et al., "A knowledge-based approach to intrusion detection modeling," *Proc. Security and Privacy Workshops (SPW), 2012 IEEE Symposium on*, IEEE, 2012, pp. 75-81.
22. J. Undercoffer, et al., "Modeling computer attacks: An ontology for intrusion detection," *Proc. International Workshop on Recent Advances in Intrusion Detection*, Springer, 2003, pp. 113-135.
23. V. Mulwad, et al., "Extracting information about security vulnerabilities from web text," *Proc. Web Intelligence and Intelligent Agent Technology (WI-IAT), 2011 IEEE/WIC/ACM International Conference on*, IEEE, 2011, pp. 257-260.
24. A. Joshi, et al., "Extracting cybersecurity related linked data from text," *Proc. Semantic Computing (ICSC), 2013 IEEE Seventh International Conference on*, IEEE, 2013, pp. 252-259.
25. A. Oltramari, et al., "Building an Ontology of Cyber Security," *Proc. STIDS*, Citeseer, 2014, pp. 54-61.
26. A. Oltramari, et al., "Computational ontology of network operations," *Proc. Military Communications Conference, MILCOM 2015-2015 IEEE*, IEEE, 2015, pp. 318-323.
27. A. Razzaq, et al., "Semantic security against web application attacks," *Information Sciences*, vol. 254, 2014, pp. 19-38.
28. Z. Syed, et al., "UCO: A unified cybersecurity ontology," *Proc. Proceedings of the AAAI Workshop on Artificial Intelligence for Cyber Security*, AAAI Press, 2016.