D

Detection and Threat-Hunting Approaches for Advanced Persistent Threats

Sutanu Kumar Ghosh and Rigel Gjomemo
University of Illinois Chicago, Chicago, IL, USA

Synonyms

AI: Artificial Intelligence; APT: Advanced Persistent Threat; C2/CnC: Command and Control; CTI: Cyber Threat Intelligence; IDS: Intrusion Detection System; IOC: Indicators of Compromise; IP: Internet Protocol; MITRE ATT&CK: MITRE Adversarial Tactics, Techniques, and Common Knowledge; POI: Point of Interest; SIEM: Security Information and Event Management; TTP: Tactics, Techniques, and Procedures

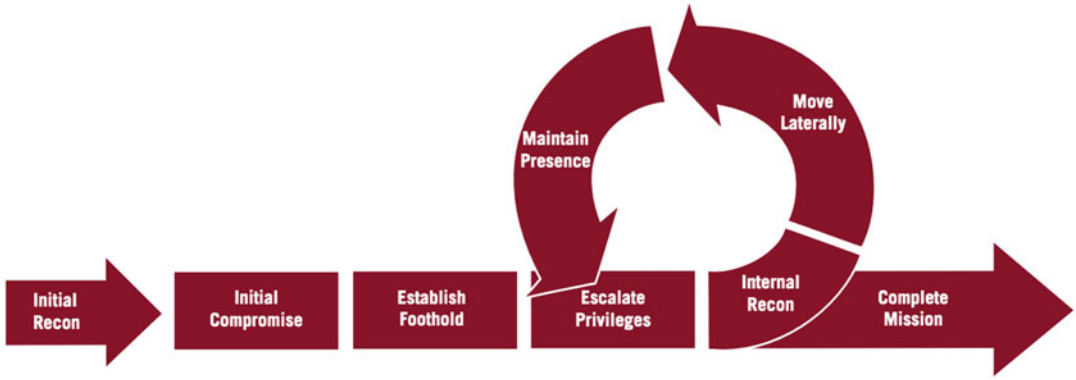
Definition

APTs are sophisticated, stealthy, and continuous cyberattack campaigns typically orchestrated by nation-states or organized groups. These attacks aim to gain prolonged unauthorized access to a network, often focusing on espionage or data theft, rather than causing immediate harm. APTs are characterized by their high level of resources, advanced techniques, and persistence in achieving their long-term objectives.

Background

APTs are highly sophisticated and stealthy attacks that target enterprise networks with the goal of exfiltrating valuable data or disrupting operations. Attackers or groups who employ APTs are often well-funded nation-state highly skilled actors. As a result, due to their sophistication, APT activities inside enterprise networks are often missed by intrusion detection systems. As an example, one of the first APTs to be reported compromised more than a hundred of organizations and stole hundreds of terabytes of data from the same. A corresponding report, published by Mandiant, revealed that on an average malwares persisted undetected in these organizations for around a year. The same report included a description of the APT life cycle, or *kill chain* (Fig. 1), which is still relevant today as almost all of the APT groups' high-level attack steps follow this life cycle. Figure 1 shows the kill chain of APTs, and on a very high level, it generally works in the following way: Threat actors conduct a thorough recon (enumeration) of the target network to identify the network landscape of the organization and potential vulnerabilities. Through initial compromise, the attackers drop malwares that exploit an existing vulnerability or gain entry

© Springer Science+Business Media LLC 2024
S. Jajodia et al. (eds.), *Encyclopedia of Cryptography, Security and Privacy*,
https://doi.org/10.1007/978-3-642-27739-9_1821-1



Detection and Threat-Hunting Approaches for Advanced Persistent Threats, Fig. 1 APT life cycle

to the network by spear-phishing emails or by any other social engineering means. Once the network (more specifically, a host in the network) is compromised, the attackers establish foothold by communicating to a command and control (C2 or CnC) server to receive instructions (and payloads) in order to carry out the remaining stages of the attack. Generally, the attacker compromises several hosts in the network by moving laterally across the network and cleans up their tracks once their foal is achieved by removing the malwares and other footprint in order to evade forensic threat hunting.

Application

Enterprise networks often consist of several hundreds of hosts, each of which produces millions of logs from the daily user activities per day. A very small percentage (about 0.01%) of those relate to actual attacker activities if an attack is unfolding in the network. Thus, detection of such APT activities is often viewed as finding *needle in a haystack*. Earlier, the detection of APTs was focused on *anomaly-based* systems (Berlin et al. 2015) which learned a benign behavior model of the hosts and detected deviations from this benign behavior. Some works (Uppuluri and Sekar 2001) also developed *specification-based* systems which rely on specification or detection policies defined by experts. Although it reduces false positives than compared to anomaly-based systems, they require application-specific policies that rely on expert knowledge. In the recent years, researchers have published several papers which tackle the APT detection problem with the help of host-audit logs and *provenance graph*-based solutions. Usage of machine learning models is also quite common nowadays (Alsaheel et al. 2021). In recent times, majority of the host-based detection systems leverages the host-audit logs to build a graphical representation of these audit logs and its related activities, which is in general termed as *provenance graphs*. Some studies (Hassan et al. 2019) use readily available graph databases such as Neo4J, PostgreSQL, and Titan, while others (Hossain et al. 2017; Milajerdi et al. 2019) use main-memory provenance graph representations. Almost all of these studies produce a high- or low-level graph as an output that represents the attacker activities inside a host. However, many of these studies suffer from *dependency explosion* problem as many of them rely on coarse-grained provenance for forensic analyses. *This often happens when a process reads from an IP or any other network source, and the subsequent write events from the process are treated to be dependent on the IP*. This leads to an overwhelming explosion in the dependency as every output event of that process becomes dependent on previous input and is evident in long-running processes such as web browsers. Another key issue which hinders cyber-analysts is *threat alert fatigue*. This happens primarily due to an overload of alarms generated by threat detec-

tion systems most of which tend to be benign or false alarms.

APT Detection Mechanisms

Hossain et al. (2017) developed a platform neutral main-memory provenance graph representation, SLEUTH, that leverages taint propagation and custom detection policies for real-time reconstruction of attack scenarios. SLEUTH addresses the key problem of an efficient event storage and event analysis (from the audit logs) based on the main memory by the development of a compact main-memory dependence graph representation, which performs far better than popular graph databases. Milajerdi et al. (2019) developed HOLMES which is a real-time APT detection system that uses the correlation between suspicious or malicious information flows to detect APTs. HOLMES maps the low-level attacker activities to the kill chain which enables it to raise critical alarms that are semantically similar to MITRE ATT&CK framework's Tactics, Techniques, and Procedures (TTP). As a result, HOLMES produces *high-level scenario graphs* in which the nodes correspond to different TTPs and the edges represent the information flow between them in real time. It tackles the dependence explosion problem by using the concept of minimum ancestral cover and developing an efficient algorithm to reduce spurious dependencies among different system entities. Hassan et al. (2019) developed an alert correlation system NODOZE which mitigates threat alert fatigue issue that burdens cyber analysts. It leverages the contextual and historical information from a generated alert and builds a causal dependency graph which then assigns an anomaly score to each edge and propagates it to the neighboring edges using a novel network diffusion algorithm. The anomaly score is assigned based on the frequencies of all the events in an enterprise which are stored in an event frequency database. Finally, a true alert dependency graph is generated based on the behavioral execution partitioning which contains the most anomalous dependency paths generated from the candidate event. NODOZE is used as an extension to the existing IDS in order to complement them for threat alert fatigue mitigation purposes.

Forensic Analysis and Threat Hunting of APTs

Analysts also leverage causality or forensic analysis as a basis for investigation and detection of APT attacks. Hossain et al. (2020) utilized two novel techniques, *tag attenuation* and *tag decay*, in a system MORSE which mitigates dependency explosion and detects single or multistage APTs. In particular, these techniques take advantage of the benign behavior of processes in order to reduce dependence explosion and leverage a strong set of tag propagation policies in order to reduce false positives and detect attacks. Inam et al. introduced DOSSIER, a system (Inam et al. 2022) that records application configuration changes, adding vital details to OS-level audit logs. It uses a kernel module to monitor file-based changes and combines program annotations with static analysis for tracking memory-based modifications in configuration variables. Fang et al.'s DEPIMPACT (Fang et al. 2022) framework assists in cyberattack investigations by analyzing dependency graphs, identifying critical points using linear discriminant analysis, and comparing backward and forward graph analyses to pinpoint key attack components.

Through threat hunting, analysts try to use CTI to determine the presence of attacker artifacts in an enterprise network. These artifacts are published in detail in CTI reports that include different IOC of an APT (or any other attack) and the behavior of those entities in a compromised system published by security companies and researchers in order to aid faster detection of new and ongoing attacks. Milajerdi et al. (2019) developed POIROT which uses the information in these publicly available CTI reports in order to detect traces of APTs in different hosts. It extracts a query graph from a CTI report which highlights the behavior of an attacker and its related entities and tries to find the presence of that query graph in the system provenance graph developed from the host audit logs. POIROT models the threat-hunting problem as an inexact graph pattern matching problem based on a novel

similarity metric which produces an alignment score between a query graph and a provenance graph. Hasan et al.'s SWIFT (Hassan et al. 2020) enhances audit log storage using an in-memory graph database, prioritizing key data for efficient memory use and applying historical data for alert management in intrusion detection. Ostinato (Ghosh et al. 2022) leveraged a unique intuition about the tools' usage of APT groups and successfully correlated attacks across different host in a network. Extractor (Satvat et al. 2021) performed automatic extraction of concise attack behaviors from CTI reports generating compact attack graphs that could be used by detection mechanisms for efficient threat hunting.

Open Problems and Future Directions

In the evolving landscape of APTs, the increasing sophistication and stealth of these groups call for adaptive approaches in detection methodologies. With the growing prevalence of insider attacks and the emergence of new zero-day exploits, future APT detection solutions must pivot their focus to address these evolving threats. Even though several academic and industrial solutions tackle the issue of false positives or threat alert fatigue in several innovative ways, however, in a recent study, Alahmadi et al. (2022) pointed out that majority of the alarms fired by IDS or SIEM in the enterprise networks are false positives. As a result, analysts spend a significant amount of time manually sifting through those alarms and further engineering those detection mechanisms according to their needs. The study also reveals the shortcomings of machine learning solutions and the need for those to be further innovated. Thus, innovating approaches to further reduce false positives in IDS and SIEM systems, improving the efficiency of threat analysis would be an immediate future research direction. Moreover, several studies (Kaspersky 2022) predict that there would be an influx of supply chain attacks in the near future. These attacks are highly rewarding and valuable to APT groups as they give access to a large number of potential targets. Hence, developing detection systems that can adapt to the evolving tactics of APTs, particularly against insider attacks, zero-day exploits, and prevention strategies for supply chain attacks should be prioritized. Another goal should be integrating AI to improve the machine learning approaches in order to predict and identify APT activities proactively, especially in dynamic network environments. These directions aim to address the multifaceted challenges posed by APTs, combining technological advancements with strategic and human-centric approaches. The CASTLE (Cyber Agents for Security Testing and Learning Environments) initiative of DARPA is designed to enhance cyber testing and evaluation methodologies. It focuses on creating a comprehensive toolkit that generates authentic network environments and educates AI agents to counter APTs. The program will leverage reinforcement learning, a subset of machine learning, to streamline and automate the process of identifying and mitigating network vulnerabilities.

References

Alahmadi BA, Axon L, Martinovic I (2022) 99% false positives: a qualitative study of soc analysts' perspectives on security alarms. In: Proceedings of the 31st USENIX Security Symposium (USENIX Security), Boston, MA, USA, pp 10–12

Alsaheel A, Nan Y, Ma S, Yu L, Walkup G, Celik ZB, Zhang X, Xu D (2021) {ATLAS}: a sequence-based learning approach for attack investigation. In: 30th USENIX Security Symposium (USENIX Security 21), pp 3005–3022

Berlin K, Slater D, Saxe J (2015) Malicious behavior detection using windows audit logs. In: Proceedings of the 8th ACM Workshop on Artificial Intelligence and Security, pp 35–44

Fang P, Gao P, Liu C, Ayday E, Jee K, Wang T, Ye YF, Liu Z, Xiao X (2022) Back-Propagating system dependency impact for attack investigation. In: 31st USENIX Security Symposium (USENIX Security 22), Boston, MA. USENIX Association

Ghosh SK, Satvat K, Gjomemo R, Venkatakrishnan V (2022) Ostinato: cross-host attack correlation through attack activity similarity detection. In: International Conference on Information Systems Security. Springer, pp 1–22

Hassan WU, Guo S, Li D, Chen Z, Jee K, Li Z, Bates A (2019) NoDoze: combatting threat alert fatigue with automated provenance triage. In: Network and Distributed Systems Security Symposium

Hassan WU, Li D, Jee K, Yu X, Zou K, Wang D, Chen Z, Li Z, Rhee J, Gui J et al (2020) This is why we can't cache nice things: lightning-fast threat hunting using suspicion-based hierarchical storage. In: Annual Computer Security Applications Conference, pp 165–178

Hossain MN, Milajerdi SM, Wang J, Eshete B, Gjomemo R, Sekar R, Stoller S, Venkatakrishnan V (2017) SLEUTH: real-time attack scenario reconstruction from COTS audit data. In: 26th USENIX Security Symposium (USENIX Security 17), Vancouver, BC. USENIX Association, pp 487–504

Hossain MN, Sheikhi S, Sekar R (2020) Combating dependence explosion in forensic analysis using alternative tag propagation semantics. In: 2020 IEEE Symposium on Security and Privacy (SP). IEEE, pp 1139–1155

Inam MA, Hassan WU, Ahad A, Bates A, Tahir R, Xu T, Zaffar F (2022) Forensic analysis of configuration-based attacks. In: 29th Annual Network and Distributed System Security Symposium, NDSS 2022

Kaspersky predicts advanced persistent threat trends in 2022 (2022). https://www.globenewswire.com/news-release/2021/11/17/2336472/0/en/Kaspersky-Predicts-Advanced-Persistent-Threat-Trends-in-2022.html

Milajerdi SM, Eshete B, Gjomemo R, Venkatakrishnan V (2019) Poirot: aligning attack behavior with kernel audit records for cyber threat hunting. In: Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security

Milajerdi SM, Gjomemo R, Eshete B, Sekar R, Venkatakrishnan V (2019) Holmes: real-time apt detection through correlation of suspicious information flows. In: 2019 IEEE Symposium on Security and Privacy (SP). IEEE, pp 1137–1152

Satvat K, Gjomemo R, Venkatakrishnan V (2021) Extractor: extracting attack behavior from threat reports. In: 2021 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, pp 598–615

Uppuluri P, Sekar R (2001) Experiences with specification-based intrusion detection. In: International Workshop on Recent Advances in Intrusion Detection. Springer, pp 172–189