Forensic Analysis of Advanced Persistent Threat Attacks in Cloud Environments

Changwei Liu, Anoop Singhal, Duminda Wijesekera

Chapter 9

FORENSIC ANALYSIS OF ADVANCED PERSISTENT THREAT ATTACKS IN CLOUD ENVIRONMENTS

Changwei Liu, Anoop Singhal and Duminda Wijesekera

Abstract Cloud forensic investigations involve large volumes of diverse devices and data. Investigations involving advanced persistent threat attacks involve filtering noisy data and using expert knowledge to identify the missing steps in the attacks that typically have long time spans. Under such circumstances, obtaining timely and credible forensic results is a challenge.

This chapter engages a case study to demonstrate how MITRE's ATT&CK knowledge base and Lockheed Martin's Cyber Kill Chain methodology can be used in conjunction to perform forensic analyses of advanced persistent threat attacks in cloud environments. ATT&CK is a globally-accessible knowledge base of adversary tactics and techniques developed from real-world observations of attacks. The Cyber Kill Chain methodology describes a series of steps that trace a cyber attack from its early reconnaissance stage to the later data exfiltration stage. Because advanced persistent threat attacks on cloud systems involve the key Cyber Kill Chain phases of reconnaissance, command and control communications, privilege escalation, lateral movement through a network and exfiltration of confidential information, it is beneficial to combine the ATT&CK knowledge base and Cyber Kill Chain methodology to identify and aggregate evidence, and automate the construction of the attack steps.

Keywords: Cloud forensics, advanced persistent threat, ATT&CK, Cyber Kill Chain

1. Introduction

Digital forensics is the application of scientific theories and methodologies to the identification, collection, examination and analysis of evidentiary data while preserving its integrity and maintaining a strict

chain of custody [8]. Due to the volume and diversity of cyber activities and devices in a cloud environment, the scope of post-attack cloud forensic investigations has expanded in two dimensions. The first is the attack surfaces of cloud devices that may not have undergone rigorous security checks. The second is the analysis of diverse data. A key concern is that servers running on virtual machines (VMs) in the cloud are monitored by hypervisors that lack warnings, procedures and tools for forensic investigations. Current computer forensic techniques are not designed for cloud environments and it is challenging to use existing tools to perform forensic analyses of cloud environments. Moreover, in the case of advanced persistent threat (APT) attacks that stretch over long periods of time (e.g., one year or more), the timestamps of evidence from different sources may not be indicators of a single attack. Investigating cloud environment attacks involves filtering noisy data and using expert knowledge and experience to speculate about the attack steps. These tasks are challenging and make it difficult to obtain credible forensic results.

Several researchers have proposed methodologies for collecting evidence from multiple sources and correlating them during forensic analyses of cloud attacks. These include collecting data from hypervisors and virtual machines [9, 12], and leveraging graphical frameworks to reconstruct cloud attack scenarios [10, 13]. However, the research is based on strong assumptions that the forensic data can be manually aggregated and pre-processed to produce evidence representing pre-attack conditions and post-attack conditions, and the forensic investigator can construct the attack steps when the associated evidence is incomplete or compromised.

The Adversarial Tactics and Common Knowledge Base (ATT&CK) developed by MITRE [16] is a globally-accessible knowledge base of adversary tactics and techniques based on real-world observations that assists in emulating cyber attacks. It has been used in recent years to create a taxonomy of attacks on enterprise information technology environments that enable defenders to understand which attacks are being used in the wild and to apply methods for detecting the attacks, including certain APT attacks.

Lockheed Martin's Intrusion Kill Chain (also called Cyber Kill Chain) methodology considers seven distinct phases that include reconnaissance, weaponization, delivery, exploitation, installation, command and control, and actions on objectives. Because most APT attacks involve successful reconnaissance, command and control communications, privilege escalation, lateral movement in a network and exfiltration of sensitive information, the Cyber Kill Chain has been used to analyze security logs,

develop attack detection and defense systems, and aggregate evidence in analyses of APT attacks [1, 15].

Inspired by these works, the research described in this chapter leverages the ATT&CK knowledge base and Cyber Kill Chain methodology to identify evidence of cloud APT attacks from various sources, aggregate the evidence and subsequently correlate the evidence to construct the attack steps. The research advances previous work on cloud forensics [10, 13] that relies on digital forensic investigators' knowledge and experience to identify evidence and construct attack steps when the associated evidence is incomplete or compromised. Although researchers have used ATT&CK and the Cyber Kill Chain independently to detect cyber attacks and aggregate/correlate evidence [1, 3, 15, 18], no published work combines the two frameworks for attack evidence identification and correlation, which is the main contribution of this research. Sample advanced persistent threat attacks on an experimental cloud environment are employed to demonstrate how the combined frameworks can be used to identify forensic data in a cloud environment and convert it to pre-attack and post-attack conditions, which are processed by a Prolog-based forensic tool to automatically construct the attack steps.

2. Background and Related Work

This section describes MITRE's ATT&CK knowledge base, Lockheed Martin's Cyber Kill Chain methodology and related work.

2.1 ATT&CK Knowledge Base

MITRE's well-known Adversarial Tactics and Common Knowledge Base (ATT&CK) is a behavioral model that is based on real-world observations [16, 18]. Unlike other threat models that were constructed by analyzing available threat/vulnerability reports, ATT&CK describes the behaviors of real adversaries. All the attack techniques in ATT&CK correspond to real-world examples employed by malware and red teams. In addition, ATT&CK has public descriptions of attack techniques, how they are leveraged and why cyber defenders should pay attention to them. Therefore, it is useful for cyber defenders and forensic investigators to decide what should be monitored and investigated, respectively, in order to construct the attack steps and mitigate the risks.

2.2 Cyber Kill Chain Methodology

Figure 1 shows the seven attack phases in Lockheed Martin's Kill Chain methodology, which cover all the steps involved in a successful cyber attack. In the first "reconnaissance" phase, the adversary iden-

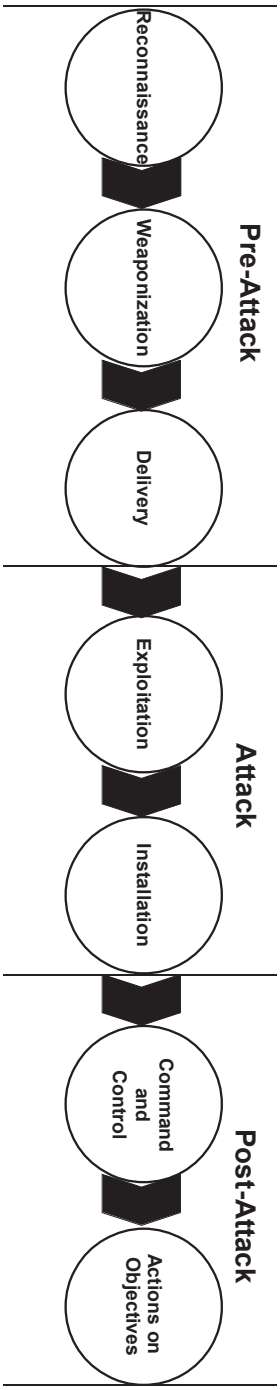

Figure 1. Cyber Kill Chain methodology.

tifies the targets by researching which targets can meet the attack objectives and collects information needed to launch the attack. In the second "weaponization" phase, the adversary prepares for the operation by coupling malware and exploits in a deliverable payload, and selecting backdoors/implants and an appropriate command and control infrastructure for the cyber operation. In the third "delivery" phase, the adversary conveys the malware to the target to launch the attack. In the fourth "exploitation" phase, the adversary triggers exploits to gain access to the target. In the fifth "installation" phase, the adversary installs a persistent backdoor or an implant in the target to maintain access for an extended period of time. In the sixth "command and control" phase, the adversary remotely controls a backdoor or implant to open a command channel so that the adversary can control the target. In the seventh and final "actions on objectives" phase, the adversary achieves the attack objectives, which include collecting user credentials, escalating privileges, destroying the system and overwriting, corrupting or modifying data [14]. According to Milajerdi at el. [15], most APT attacks are accomplished via steps that conform to the Cyber Kill Chain methodology and have the goal of obtaining and exfiltrating highly confidential information.

Cyber Reboot [2] has reexamined the seven phases and argued that there are three fundamental phases to most cyber attacks: (i) pre-attack; (ii) attack; and (iii) post-attack (Figure 1). During the pre-attack phase, the attacker is tasked with the attack objectives and performs reconnaissance of the target. During the attack phase, the attack is executed, enabling the attacker to break through the target's defense and set up communications with the target. During the post-attack phase, further exploitation and access of the target occur, which enable the attacker to escalate his/her privileges, destroy the victim system, steal confidential information, etc.

2.3 Related Work

Techniques such as remote data acquisition, management plane acquisition, live forensics and snapshot analysis have been proposed to collect evidence from cloud environments [17]. Dykstra and Sherman [5] have retrieved volatile and non-volatile data from an active Amazon EC2 cloud user instance platform using traditional forensic tools such as EnCase and FTK. In order to validate the integrity of the collected data, they subsequently developed the FROST toolkit that can be integrated in OpenStack to collect logs from an operating system that runs virtual machines [6]; however, this technique assumes that the cloud provider is

trustworthy. Zawoad and Hasan [19] recently eliminated this assumption by designing a forensics-enabled cloud.

Hay and Nance [7] have conducted live digital forensic analyses of cloud environments with virtual introspection, a process that enables the hypervisor or any other virtual machine to observe the state of a chosen virtual machine. Dolan-Gavitt et al. [4] have bridged the semantic gap between high-level state information and low-level sources such as physical memory and CPU registers, and have developed a suite of virtual introspection tools for Xen and KVM. Several hypervisors, including Xen, VMware, ESX and Hyper-V, support snapshot features that can be used to obtain information about the running states of virtual machines.

In order to reduce the time and effort involved in forensic investigations, researchers have automated evidence correlation and attack reconstruction by leveraging rule-based tools and business process diagrams [13]. However, these approaches rely on forensic experts when the evidence is missing, disjointed or compromised. To help investigate attacks in a methodical manner and detect real-time APT attacks, the Cyber Kill Chain methodology has been modified to facilitate data aggregation in a relational database [1, 15].

3. Experimental Cloud Environment Attacks

This section describes an experimental cloud environment that was targeted by conventional and cloud cyber attacks. The experimental environment and attacks are used to demonstrate how the ATT&CK knowledge base and Cyber Kill Chain methodology can be used together to advance cloud forensic investigations.

Based on the types of vulnerabilities and attacker capabilities, attacks on cloud environments can be categorized into two groups [10, 12]: (i) attacks from the Internet that exploit conventional cyber vulnerabilities to attack a virtual machine connected to the Internet; and (ii) attacks from a virtual machine that exploit vulnerabilities in shared cloud management resources to launch attacks on other virtual machines on the same hypervisor. The attacks include denial of service, information leakage, privilege escalation and arbitrary code execution, among others.

Figure 2 shows the experimental cloud environment and sample attacks. The environment comprised two Linux (Ubuntu 14.04) virtual machines, VM1 and VM2, configured on the same hypervisor (Xen 4.6). Additionally, a Windows machine was configured as a web server from which a web application could use SQL queries to retrieve database data stored in VM2, a file server that hosted a database and other files.

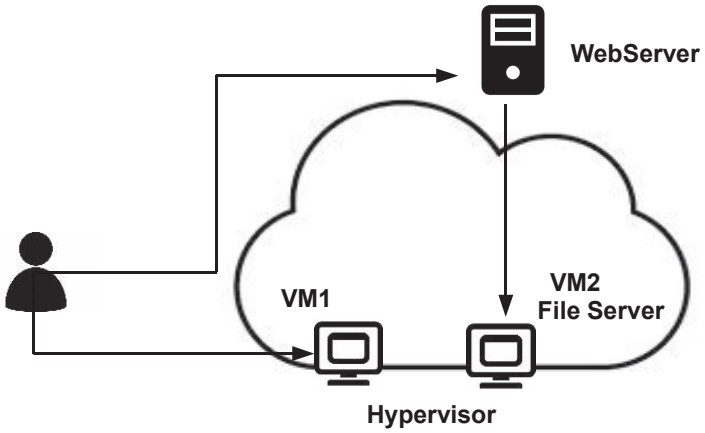

Figure 2. Experimental cloud environment and sample attacks.

Two attacks were launched at VM2. One was a conventional SQL injection attack that exploited the web application vulnerability that does not sanitize user inputs. The other one was a virtual machine escape attack, which could be a type of APT attack. The virtual machine escape attack exploited the CVE-2017-7228 vulnerability in VM1, which enabled VM1 to control Xen's privileged domain (domain 0) and then VM2, so that it could perform local operations such as deleting a file in VM2.

3.1 Forensic Data Obtained via Forensic Tools

Forensic data was collected by logging web server accesses, deploying the Snort intrusion detection system to monitor network traffic to the web server and file server, and installing the LibVMI virtual machine introspection tool on Xen Dom0 to capture events and running processes on the guest virtual machines VM1 and VM2. LibVMI is a C library that can be used to monitor the low-level details of a running Xen virtual machine by viewing its memory, trapping hardware events and accessing vCPU registers.

The IP addresses and forensic data captured using the methods/tools mentioned above are shown in Table 1 and Figures 3, 4 and 5, respectively. According to Table 1, the Snort alert in Figure 3 shows that the attacker at IP address `129.174.124.122` attempted to launch an SQL injection attack using the web application deployed on the web server at IP address `129.174.125.35` (port number 8080).

Table 1. IP addresses of machines and virtual machines in Figure 2.

Machine/Virtual Machine	IP Address
Attacker	129.174.124.122
Web Server	129.174.125.35
VM1	129.174.124.184
VM2 (File Server)	129.174.124.137

```
[**] SQL Injection Attempt --1=1 [**]
08/08-14:37:27.818279 129.174.124.122:1715 -> 129.174.124.35:8080
TCP TTL:128 TOS:0x0 ID:380 IpLen:20 DgmLen:48 DF
******S* Seq: 0xDEDBEABF  Ack: 0x0  Win: 0xFFFF  TcpLen: 28
TCP Options (4) => MSS: 1460 NOP NOP SackOK
...
```

Figure 3. Sample Snort alert.

```
129.174.124.122 - - [08/Aug/2019:14:35:34 -0400] "GET /lab/Test
                    HTTP/1.1" 200 368
129.174.124.122 - - [08/Aug/2019:14:35:39 -0400] "POST /lab/Test
                    HTTP/1.1" 200 981
...
```

Figure 4. Sample access log from the web server.

The web access history on the web server in Figure 4 shows that the attacker machine accessed the web application just before the Snort alert shown in Figure 3.

Figure 5 shows the SQL database log with the SQL injection query (`40 Query select * from profiles where name='Alice' AND password ='alice' or '1'='1'`) that resulted in the information leakage.

Figure 6 shows the forensic data obtained by running LibVMI on the attacker virtual machine. The data includes the running processes (Figure 6(a)), injected Linux modules (Figure 6(b)) and CPU (CR3) register values corresponding to the running processes (Figure 6(c)). Note that the process identifiers (PIDs) were used to find the process names.

```
130808 14:37:29
   40 Query SET NAMES latin1
   40 Query SET character_set_results = NULL
   40 Query SET autocommit=1
   40 Query SET GLOBAL general_log = 'ON'
   40 Query select * from profiles where name='Alice' AND
            password='alice' or '1'='1'
   40 Quit
```

Figure 5. Sample SQL database log.

```
[  630] agetty (struct addr:ffff880003c8e200)
[  669] systemd (struct addr:ffff880076060000)
[  674] (sd-pam) (struct addr:ffff880076104600)
[  677] bash (struct addr:ffff880003c8aa00)
[  703] sudo (struct addr:ffff880004341c00)
[  704] attack (struct addr:ffff880004343800)
```

(a) Running processes.

```
test
intel_rapl
x86_pkg_temp_thermal
coretemp
...
```

(b) Injected Linux modules.

```
Waiting for events...
PID 0 with CR3=77130000 executing on vcpu 1. Previous CR3=788d1000
Waiting for events...
PID 1246 with CR3=788d1000 executing on vcpu 1. Previous CR3=77130000
```

(c) CPU register values.

Figure 6. VM2 processes, injected Linux modules and CPU register values.

4. Forensic Investigation

This section shows how ATT&CK and the Cyber Kill Chain methodology are used to assist the forensic investigation.

4.1 Identifying Forensic Data

MITRE's ATT&CK includes a knowledge base of 11 tactics and hundreds of techniques that an attacker could leverage when compromising an enterprise environment. A tactic in ATT&CK is a high-level description of certain types of attack behavior whereas a technique provides a detailed description of every type of behavior within a tactic class. The tactics in ATT&CK are not followed in a linear order as in the case of Lockheed's Cyber Kill Chain methodology. Additionally, an attacker may bounce between tactics in order to achieve the final goal.

Forensic data is mapped to the ATT&CK matrix [16] in order to help identify the evidence in a cloud forensic investigation. The matrix model covers the phases of the attack lifecycle: "initial access," "execution," "persistence," "privilege escalation," "defense evasion," "credential access," "discovery," "lateral movement," "collection," "command and control," "exfiltration" and "impact" (first column). Each phase involves the application of various techniques listed in the matrix (second and third columns). The following are the general descriptions of the phases:

- **Initial Access:** This phase involves the application of techniques that use entry vectors to gain an initial foothold in a network, which may provide the attacker with continued access to external remote services.

- **Execution:** This phase involves the application of techniques that cause attacker-controlled code to execute on a local or remote system, which can achieve broader goals such as exploring a network or stealing data by pairing the techniques with other techniques. Note that this phase may not leave any evidence.

- **Persistence:** This phase involves the application of techniques that enable an attacker to maintain a foothold on a system, even after system interruptions cut off attacker access.

- **Privilege Escalation:** This phase involves the application of techniques that enable an attacker to gain higher-level privileges in a system or network. Common approaches involve exploiting system weaknesses, misconfigurations and vulnerabilities.

- **Defense Evasion:** This phase involves the application of techniques that uninstall or disable security software, or obfuscate or encrypt data and scripts used by an attacker to avoid detection over the entire attack lifecycle.

- **Credential Access:** This phase involves the application of techniques that enable an attacker to steal credentials to gain system or network access, providing the opportunity to create multiple accounts to achieve the attack goals.
- **Discovery:** This phase involves the application of techniques that enable an attacker to gain knowledge about the system and network. During this phase, the attacker explores what can be controlled and obtains knowledge that could advance the post-compromise information-gathering goals.
- **Lateral Movement:** This phase involves the application of techniques that enable an attacker to enter and control systems in a network. An attacker might install custom remote access tools to accomplish lateral movement or use legitimate credentials with the help of native network and operating system tools.
- **Collection:** This phase involves the application of techniques that enable an attacker to gather sensitive information. Having obtained the information, the attacker may proceed to exfiltrate (steal) the information.
- **Command and Control:** This phase involves the application of techniques that enable an attacker to communicate with and control systems in the targeted network.
- **Exfiltration:** This phase involves the application of techniques that enable an attacker to steal sensitive information from the targeted network. The attacker often compresses or encrypts the information to avoid detection. The channels used for exfiltration typically include the attacker's command and control channel or an alternate channel with limited bandwidth.
- **Impact:** This phase involves the application of techniques that enable an attacker to disrupt availability or compromise integrity by manipulating business and operational processes, including destroying or tampering with data.

In the experimental cloud environment, evidence of the SQL injection attack was provided by the Snort alert (Figure 3) along with the SQL query (Figure 5), which clearly identified it as an SQL injection attack.

In the case of the virtual machine escape attack that exploited the CVE-2017-7228 vulnerability, although the attack was observed (deletion of a file in VM2), it was difficult to construct the attack from the data obtained using LibVMI (Figure 6). This is because there was no

obvious logged data that could help identify the attack. Clearly, this step did not leave any evidence.

In such a situation, ATT&CK could be used to narrow the scope of the search and help find evidence. According to ATT&CK, the initial access techniques include "drive-by compromise, exploit public facing application, external remote services, hardware additions, replication through removable media, spear phishing attachment and trusted relationship."

In the experimental cloud environment, except for the facts that the database in VM2 could be queried by the web application on the web server and that VM2 shared the same hypervisor (and thus hardware) with VM1, it did not have any other connected media, remote services or running applications. Thus, the initial accesses could be narrowed to the tactics: "exploit public-facing application" from the web server and "hardware additions" from the hypervisor.

Because the observed attack activities on VM2 included the SQL injection alert and file deletion, according to ATT&CK, the attack execution techniques fall into two categories "exploitation for client execution" (corresponding to the web application on the web server) and "command-line interface" (corresponding to the hardware addition). Additionally, the techniques for "privilege escalation" could be narrowed down to "exploitation for privilege escalation" because the attacker obviously escalated his/her privileges over the Internet or from the other virtual machine remotely. Other techniques such as "access token manipulation" and "accessibility features" would not be applicable given the configuration of the cloud environment.

The SQL injection attack left obvious evidence as shown in Figures 3, 4 and 5. However, in the case of the virtual machine escape attack that resulted in the file deletion, the data in Figure 6 only show the running processes (including the normal Linux processes and a suspicious user process named `attack`) and injected modules (including normal Linux modules and a suspicious injected user module named `test`). No information was available about the attack process that exploited the shared hardware vulnerability.

Using the potential attack tactics from ATT&CK, a forensic practitioner could continue to investigate more forensic data related to successful exploitations of "hardware additions" and "command-line interface" that enabled the attacker to escalate privileges to the hypervisor level and proceed to delete the file in VM2. Previous papers by the authors of this chapter [10, 12] have revealed that system calls constitute good forensic evidence, so a snapshot of VM2 captured during the attack was used to retrieve the system calls and kernel messages of the suspicious process `attack` and suspicious module `test`.

```
 1. execve("./attack", ["./attack", "rm victim ~/samplefile.txt"],
    [/* 30 vars */]) = 0
 2. brk(NULL) = 0x8cd000
 3. mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS,
    -1, 0) = 0x7fa3a3022000
 4. access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or
    directory)
 5. open("/etc/ld.so.cache", O_RDONLY|O_CLOEXEC) = 3
... ...
25. open("test.ko", O_RDONLY) = 3
26. finit_module(3, "user_shellcmd_addr=1407334317317"..., 0) = 0
27. fstat(1, {st_mode=S_IFCHR|0620, st_rdev=makedev(136, 0), ...}) = 0
28. mmap(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS,
    -1, 0) = 0x7fa3a3021000
29. mmap(0x600000000000, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|
    MAP_FIXED|MAP_ANONYMOUS|MAP_LOCKED, -1, 0) = 0x600000000000
30. delete_module("test", O_NONBLOCK) = 0
31. exit_group(0) = ?
```

Figure 7. System calls obtained by tracing the `attack` process.

Figure 7 shows the system calls obtained by tracing the `attack` process. In fact, the arguments following the `execve` command in Line 1 clearly reveal that the attacker on VM1 used a command line to execute a program named `attack` and attempted to delete the file `samplefile.txt` located in the home folder of VM2 (named `victim` in the experimental network). Also, Line 25 clearly shows that the Linux module `test.ko` was injected into the Linux kernel of VM1 for some reason.

Figure 8 shows the VM2 kernel activities. The kernel messages between Lines 1 and 6 reveal that the attacker on VM1 wrote some bytes to memory after the address `ffff88007c723008`. The messages between Lines 8 and 19 show that the attacker controlled the page table in Xen to execute his/her shellcode by linking the physical memory address where the shellcode was held to the virtual memory address in the page table. This clearly shows the attacker used the shared memory to launch the attack.

Identifying an attack component is not a trivial task due to the nature of APTs. It requires detailed analysis such as looking at all the processes and process threads that could have altered the state of an object, even under enhanced super-user privileges. As shown in this example, identifying some of these missing steps may have to consider the system call logs. ATT&CK maintains tactics and techniques that

```
 1. [  127.408066] write_byte_hyper(ffff88007c723008, 0x7)
 2. [  127.436071] write_byte_hyper(ffff88007c723009, 0x90)
 3. [  127.460074] write_byte_hyper(ffff88007c72300a, 0xba)
 4. [  127.484055] write_byte_hyper(ffff88007c72300b, 0x26)
 5. [  127.512054] write_byte_hyper(ffff88007c72300c, 0x1)
 6. [  127.548083] write_byte_hyper(ffff88007c72300d, 0x0)
 7. [  127.628071] write_byte_hyper(ffff88007c723010, 0x0)
 8. [  127.660074] going to link PMD into target PUD
 9. [  127.668058] linked PMD into target PUD
10. [  127.676046] going to unlink mapping via userspace PUD
11. [  127.684077] mapping unlink done
12. [  127.692076] copying HV and user shellcode...
13. [  127.700077] copied HV and user shellcode
14. [  127.708066] int 0x85 returned 0x7331
15. [  127.716077] remapping paddr 0x21e8dd000 to vaddr
                   0xffff880079846800
16. [  127.724076] IDT entry for 0x80 should be at 0xffff83021e8dd800
17. [  127.732080] remapped IDT entry for 0x80 to 0xffff804000100800
18. [  127.740077] IDT entry for 0x80: addr=0xffff82d080229ef0,
                   selector=0xe008, ist=0x0, p=1, dpl=3, s=0, type=15
19. [  127.748085] int 0x85 returned 0x1337
```

Figure 8. Kernel message from the injected module.

are based on real-world observations, which makes them very helpful for identifying processes and system calls related to a sub-attack phase of an APT attack.

4.2 Mapping Log Entries to Attack Steps

After the evidence has been identified by leveraging ATT&CK, the Cyber Kill Chain model can be used to map the evidence to various attack phases in order to construct the attack steps.

The evidence shown in Figures 3, 4 and 5 pertain to the SQL injection attack because it is consistent with the timestamps and alerts. The data in Figures 3 and 4 reveal that the attacker at IP address `129.174.124.122` accessed the web server at IP address `129.174.124.35` using the SQL injection attempt (`'1'='1'`), which is considered to be "initial access" in ATT&CK. This is easily mapped to "weaponization" in pre-attack phase. The data in Figure 5 shows that, at the same time, the database was queried using `select * from profiles where name='Alice' AND password='alice' or '1'='1'`, which is clearly an SQL injection attack on the database. Since the database did not have

any security mechanisms, the implication is that the attack was successful. Therefore, the data in Figure 5 can be mapped to the "attack" phase in the Cyber Kill Chain.

The forensic data in Figure 6 was linked to the same attack by matching the process name `attack` and injected module name `test.ko`. Because the data in Figures 6(a) and 6(b) only show that the attacker from VM1 ran the process `attack` and module `test` to do some work (without any details), the data could be mapped to the "weaponization" phase that belongs to the pre-attack stage.

In addition, as described in Section 4.1, the data in Figures 7 and Figure 8 show that the attacker manipulated the shared memory in the same hypervisor to execute shellcode on the victim virtual machine, which can be mapped to the "exploitation" phase in the attack stage. Because the `samplefile.txt` file in the victim virtual machine was deleted, the attack succeeded and can be mapped to "actions on objectives" in the Cyber Kill Chain of the post-attack stage.

4.3 Correlating Attack Steps to APTs

In previous work by the authors [11], a Prolog-based tool was employed to generate attack steps using evidence (expressed as Prolog predicates) to instantiate rules with the predicates that represented attack pre-conditions and post-conditions. The rules, which simulated generic attack techniques, were written in the form: p :- p_1, p_2, $\cdots$, p_n, where the predicate p represents the post-conditions of an attack and predicates p_1, p_2, $\cdots$, p_n represent the pre-conditions of the attack. The post-conditions refer to the privileges that the attacker obtained after the attack and the pre-conditions include the attacker's initial privileges, location, system configuration and vulnerability exploited by the attack.

While the Prolog-based tool can be used to generate attack steps, it requires users to categorize evidence pertaining to the post-attack conditions, attack techniques and pre-attack conditions. The tool does not map predicates to the seven Cyber Kill Chain phases. Also, it does not have corresponding rules that correlate the evidence associated with the seven phases of the Cyber Kill Chain to pre-attack conditions and post-attack conditions.

The deficiencies are addressed by making the following changes:

1. Predicates Pr_r, Pr_w, Pr_d, A_e, A_i, Po_c and Po_a are used to represent the pre-attack "reconnaissance," "weaponization" and "delivery" phases, the attack "exploitation" and "installation" phases, and the post-attack "command and control" and "actions on objectives" phases, respectively.

Table 2. Descriptions of the nodes in Figure 9.

Node	Description	Node	Description
1	ExecCode(VM2, read)	6	networkServiceInfo(database, httpd, tcp, 3660, user)
2	ExecCode(VM2, modify)	7	vulExists(webServer, 'CWE89,' httpd)
3	Through 3 (Remote exploit of server)	8	hasAccount(attacker, VM1, root)
4	Through 8 (Compromise of host via shared hardware)	9	vulExists(VM2, 'CVE-2017-7228,' sharedmemory)
5	attackerAccess(publicWebApp)	10	vulProperty('CVE-2017-7228,' localExploit, privEscalation)

2. Techniques in the ATT&CK matrix are converted to the corresponding predicates and mapped to the Cyber Kill Chain phases as follows: (i) predicates of "initial access" are mapped to Pr_r; (ii) predicates of "execution," "persistence," "privilege escalation," "defense evasion" and "credential access" are mapped to Pr_w; (iii) predicates of "discovery" are mapped to A_i; (iv) predicates of "lateral movement" are mapped to A_i; (v) predicates of "command and control" are mapped to Po_c; and (vi) predicates of "collection," "exfiltration" and "impact" are mapped to Po_a.

Note that symbols Pr_r, Pr_w, Pr_d, A_e, A_i, Po_c and Po_a are used to categorize predicates to pre-attack conditions, attack techniques and post-attack conditions, which are removed when the predicates are presented to show the constructed attack steps, as illustrated in Table 2. In the table, Nodes 5, 6, 7, 8, 9, 10 correspond to pre-attack conditions; Nodes 3, 4 correspond to attack techniques; and Nodes 1, 2 correspond to post-attack conditions.

The predicates have names and variables that depict facts such as system configuration, attacker privileges, network topology, operating system permissions and software vulnerability. The "exploit public-facing application" technique in "initial access" of the ATT&CK matrix is written to "Pr_r(attackerAccess(_host, _program))" and the "account manipulation" technique in "credential access" of the ATT&CK matrix is written to "Pr_w(hasAccount(_principal, _host, _account))," where the variables (e.g., _host, _program, _account) following the predicate names (e.g., "attackerAc-

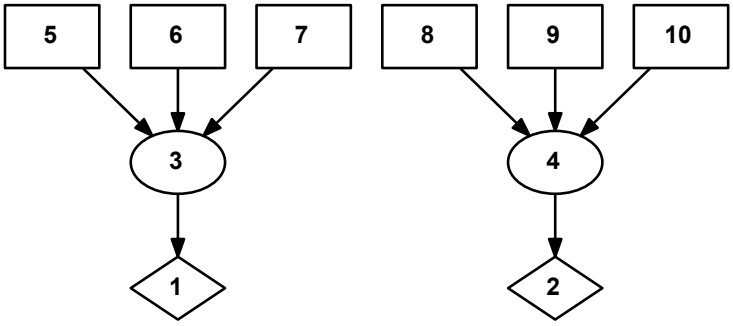

Figure 9. Constructed attack steps in the experimental cloud environment.

cess" and "hasAccount") are instantiated using concrete information during the execution of the Prolog tool.

3. Rules are added to use the Cyber Kill Chain to correlate the predicates corresponding to different phases to an attack step. The rules are of the form:

$$Po_c : -(Pr_r; Pr_w; Pr_d), (A_e; A_i).$$

and

$$Po_a : -(Pr_r; Pr_w; Pr_d), (A_e; A_i).$$

where ";" denotes logical OR and "," denotes logical AND.

These rules mean that, if there is evidence found in all the pre-attack, attack and post-attack phases, then an attack step is constructed.

After incorporating these changes in the Prolog-based tool, intuitive graphical attack steps were constructed for the two attacks as shown in Figure 9 (Table 2 provides the node descriptions). The left path shows that the attacker used a publicly-available web application to launch the SQL injection attack on the database in VM2. The right path shows the attacker exploited the vulnerability in the shared hardware to attack VM2 and then deleted a file in VM2.

Note that the miniature example provides an initial example of the ATT&CK rule model. Although some rules – e.g., lateral movement by the attacker and passing the hash attacks – are missing, the missing rules could be generated by machine learning algorithms and incorporated into the steps of the ATT&CK process.

5. Conclusions

Justifying the pre-attack, attack and post-attack phases requires evidence of activities related to the phases. When performing an APT attack analysis, difficulties are encountered in constructing attack steps because: (i) APT attacks do not lend themselves to using time as indicators for identifying forensic evidence; and (ii) recognizing the pre-attack and post-attack phases may require the application of statistical correlation techniques on evidence from multiple sources. As a result, creating valid arguments for APT attacks becomes more challenging, in particular, assigning timestamps to APT attacks in a cloud environment.

The ATT&CK knowledge base is readily leveraged to identify the evidence and build the attack steps by mapping the available evidence to various phases in the Cyber Kill Chain methodology. The experimental cloud environment case study validates the benefits of combining the ATT&CK knowledge base and Cyber Kill Chain methodology to identify and aggregate evidence, and feed it to a Prolog-based tool that can automate the construction of the attack steps.

Future research will attempt to extend the relationships between the Cyber Kill Chain and the evidence gathering and attack-attribution tasks.

References

[1] B. Bryant and H. Saiedian, A novel kill-chain framework for remote security log analysis with SIEM software, *Computers and Security*, vol. 67, pp. 198–210, 2017.

[2] Cyber Reboot, The Cyber Attack Chain, In-Q-Tel, Menlo Park, California (`www.cyberreboot.org`), 2020.

[3] A. D'Amico and K. Whitley, The real work of computer network defense analysts, *Proceedings of the Workshop on Visualization for Computer Security*, pp. 19–37, 2007.

[4] B. Dolan-Gavitt, B. Payne and W. Lee, Leveraging Forensic Tools for Virtual Machine Introspection, Technical Report GT-CS-11-05, School of Computer Science, Georgia Institute of Technology, Atlanta, Georgia, 2011.

[5] J. Dykstra and A. Sherman, Acquiring forensic evidence from infrastructure-as-a-service cloud computing: Exploring and evaluating tools, trust and techniques, *Digital Investigation*, vol. 9(S), pp. S90–S98, 2012.

[6] J. Dykstra and A. Sherman, Design and implementation of FROST: Digital forensic tools for the OpenStack cloud computing platform, *Digital Investigation*, vol. 10(S), pp. S87–S95, 2013.

[7] B. Hay and K. Nance, Forensic examination of volatile system data using virtual introspection, *ACM SIGOPS Operating Systems Review*, vol. 42(3), pp. 74–82, 2008.

[8] K. Kent, S. Chevalier and T. Grance, Guide to Integrating Forensic Techniques into Incident Response, NIST Special Publication 800-86, National Institute of Standards and Technology, Gaithersburg, Maryland, 2006.

[9] LibVMI Community, LibVMI: LibVMI Virtual Machine Introspection, LibVMI (`libvmi.com`), 2020.

[10] C. Liu, A. Singhal, R. Chandramouli and D. Wijesekera, Determining forensic data requirements for detecting hypervisor attacks, in *Advances in Digital Forensics XV*, G. Peterson and S. Shenoi (Eds.), Springer, Cham, Switzerland, pp. 253–272, 2019.

[11] C. Liu, A. Singhal and D. Wijesekera, A probabilistic network forensic model for evidence analysis, in *Advances in Digital Forensics XII*, G. Peterson and S. Shenoi (Eds.), Springer, Cham, Switzerland, pp. 189–210, 2016.

[12] C. Liu, A. Singhal and D. Wijesekera, Identifying evidence for cloud forensic analysis, in *Advances in Digital Forensics XIII*, G. Peterson and S. Shenoi (Eds.), Springer, Cham, Switzerland, pp. 111–130, 2017.

[13] C. Liu, A. Singhal and D. Wijesekera, A layered graphical model for cloud forensic mission attack impact analysis, in *Advances in Digital Forensics XIV*, G. Peterson and S. Shenoi (Eds.), Springer, Cham, Switzerland, pp. 263–289, 2018.

[14] Lockheed Martin Corporation, Gaining the Advantage – Applying Cyber Kill Chain Methodology to Network Defense, Bethesda, Maryland, 2015.

[15] S. Milajerdi, R. Gjomemo, B. Eshete, R. Sekar and V. Venkatakrishnan, HOLMES: Real-time APT detection through correlation of suspicious information flows, *Proceedings of the IEEE Symposium on Security and Privacy*, pp. 1137–1152, 2018.

[16] MITRE Corporation, ATT&CK Matrix for Enterprise, Bedford, Massachusetts (`attack.mitre.org`), 2020.

[17] A. Pichan, M. Lazarescu and S. Soh, Cloud forensics: Technical challenges, solutions and comparative analysis, *Digital Investigation*, vol. 13, pp. 38–57, 2015.

[18] B. Strom, J. Battaglia, M. Kemmerer, W. Kupersanin, D. Miller, C. Wampler, S. Whitley and R. Wolf, Finding Cyber Threats with ATT&CK-Based Analytics, MITRE Technical Report MTR170202, MITRE Corporation, Annapolis Junction, Maryland, 2017.

[19] S. Zawoad and R. Hasan, A trustworthy cloud forensics environment, in *Advances in Digital Forensics XI*, G. Peterson and S. Shenoi (Eds.), Springer, Cham, Switzerland, pp. 271–285, 2015.