Pyramid of Pain

Basically it consists of 7 Stages including

- Hash values(trivial)
- IP address(Easy)
- Domain names(simple)
- Host artifacts(annoying)
- Network artifacts(annoying)
- Tools(Challenging)
- TTPs(Tough)

Hash Values (Trivial)

As per Microsoft, the hash value is a numeric value of a fixed length that uniquely identifies data. A hash value is the result of a hashing algorithm. The following are some of the most common hashing algorithms:

- MD5
- SHA1
- SHA2

A hash is not considered to be cryptographically secure if two files have the same hash value or digest. Security professionals use hash values to get insight into a specific malware sample, a malicious file. Various online tools can be used to do hash lookups like VirusTotal and Metadefender Cloud - OPSWAT. However, as an attacker, modifying a file by even a single bit is trivial, which would produce a different hash value. With so many variations and instances of known malware or ransomware, threat hunting using file hashes as the IOC (Indicators of Compromise) can become difficult.

IP Address (Easy)

An IP address is used to identify any device connected to a network. These devices range from desktops, to servers and even CCTV cameras! We rely on IP addresses to send and receive the information over the network. But we are not going to get into the structure and functionality of the IP address. As a part of the Pyramid of Pain, we'll evaluate how IP addresses are used as an indicator. From a defense standpoint, knowledge of the IP addresses an adversary uses can be valuable. A common defense tactic is to block, drop, or deny inbound requests from IP addresses on your parameter or external firewall. This tactic is often not bulletproof as it's trivial for an experienced adversary to recover simply by using a new public IP address.

One of the ways an adversary can make it challenging to successfully carry out IP blocking is by using **Fast Flux**.

According to Akamai, Fast Flux is a DNS technique used by botnets to hide phishing, web proxying, malware delivery, and malware communication activities behind compromised hosts acting as proxies. The purpose of using the Fast Flux network is to make the communication between malware and its command and control server (C&C) challenging to be discovered by security professionals.

So, the primary concept of a Fast Flux network is having multiple IP addresses associated with a domain name, which is constantly changing. Palo Alto created a great fictional scenario to explain Fast Flux: "Fast Flux 101: How Cybercriminals Improve the Resilience of Their Infrastructure to Evade Detection and Law Enforcement Takedowns"

Domain name (Simple)

Domain name is basically mapping the ip address to a string of text. A domain name can contain a domain and a top-level domain (evilcorp.com) or a sub-domain followed by a domain and top-level domain (tryhackme.evilcorp.com). But we will not go into the details of how the Domain Name System (DNS) works. Domain Names can be a little more of a pain for the attacker to change as they would most likely need to purchase the domain, register it and modify DNS records. Unfortunately

for defenders, many DNS providers have loose standards and provide APIs to make it even easier for the attacker to change the domain.

Imagine there are two addresses ad1das.de and adidas.de you will see there is something wrong in the URL but this is called punny code attacks. What is Punycode? As per Wandera, "Punycode is a way of converting words that cannot be written in ASCII, into a Unicode ASCII encoding." Internet Explorer, Google Chrome, Microsoft Edge, and Apple Safari are now pretty good at translating obfuscated characters into the full Punycode domain name.To detect malicious domains, proxy logs or web server logs can be used.

Attackers usually hide the malicious domains under **URL Shorteners.** A URL Shortener is a tool that creates a short and unique URL that will redirect to the specific website specified during the initial step of setting up the URL Shortener link. According to Cofense, attackers use the following URL Shortening services to generate malicious links:

- bit.ly
- goo.gl
- ow.ly
- s.id
- smarturl.it
- tiny.pl
- tinyurl.com
- x.co

You can see the actual website the shortened link is redirecting you to by appending "+" to it (see the examples below). Type the shortened URL in the address bar of the web browser and add the above characters to see the redirect URL.

To practise this and see how things work actually you can just go to the any.run it is a sandboxing environment but make sure you don’t follow any links as these are malwares and doing something wrong.

Host Artifacts(annoying)

On this level, the attacker will feel a little more annoyed and frustrated if you can detect the attack. The attacker would need to circle back at this detection level and change his attack tools and methodologies. This is very time-consuming for the attacker, and probably, he will need to spend more resources on his adversary tools.

Host artifacts are the traces or observables that attackers leave on the system, such as registry values, suspicious process execution, attack patterns or IOCs (Indicators of Compromise), files dropped by malicious applications, or anything exclusive to the current threat. See the below example for diving more into this.

Suspicious events followed by opening a malicious application:

Time o...	Process Name	PID	Operation	Path	Result
3:24:26....	POwersheLL.exe	3540	WriteFile	C:\Users\RussianPanda\Jehhzda\Ben14fr\G_jugk.exe	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	WriteFile	C:\Users\RussianPanda\Jehhzda\Ben14fr\G_jugk.exe	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	WriteFile	C:\Users\RussianPanda\Jehhzda\Ben14fr\G_jugk.exe	SUCCESS
3:24:26....	POwersheLL.exe	3540	WriteFile	C:\Users\RussianPanda\Jehhzda\Ben14fr\G_jugk.exe	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	WriteFile	C:\Users\RussianPanda\Jehhzda\Ben14fr\G_jugk.exe	SUCCESS
3:24:26....	POwersheLL.exe	3540	WriteFile	C:\Users\RussianPanda\Jehhzda\Ben14fr\G_jugk.exe	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	WriteFile	C:\Users\RussianPanda\Jehhzda\Ben14fr\G_jugk.exe	SUCCESS
3:24:26....	POwersheLL.exe	3540	WriteFile	C:\Users\RussianPanda\Jehhzda\Ben14fr\G_jugk.exe	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	WriteFile	C:\Users\RussianPanda\Jehhzda\Ben14fr\G_jugk.exe	SUCCESS
3:24:26....	POwersheLL.exe	3540	WriteFile	C:\Users\RussianPanda\Jehhzda\Ben14fr\G_jugk.exe	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	WriteFile	C:\Users\RussianPanda\Jehhzda\Ben14fr\G_jugk.exe	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	WriteFile	C:\Users\RussianPanda\Jehhzda\Ben14fr\G_jugk.exe	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS
3:24:26....	POwersheLL.exe	3540	WriteFile	C:\Users\RussianPanda\Jehhzda\Ben14fr\G_jugk.exe	SUCCESS
3:24:26....	POwersheLL.exe	3540	TCP Receive	192.168.75.222:1047 -> 35.214.215.33:80	SUCCESS

Network artifacts (annoying)

Network Artifacts also belong to the yellow zone in the Pyramid of Pain. This means if you can detect and respond to the threat, the attacker would need more time to go back and change his tactics or modify the tools, which gives you more time to respond and detect the upcoming threats or remediate the existing ones.

A network artifact can be a user-agent string, C2 information, or URI patterns followed by the HTTP POST requests.An attacker might use a User-Agent string that hasn’t been observed in your environment before or seems out of the ordinary. The User-Agent is defined by RFC2616 as the request-header field that contains the information about the user agent originating the request.Network artifacts can be detected in Wireshark PCAPs (file that contains the packet data of a network) by using a network protocol analyzer such as TShark or exploring IDS (Intrusion Detection System) logging from a source such as Snort.

These are the most common User-Agent strings found for the Emotet Downloader Trojan . If you can detect the custom User-Agent strings that the attacker is using, you might be able to block them, creating more obstacles and making their attempt to compromise the network more annoying.

Tools(Challenging)

At this stage, we have levelled up our detection capabilities against the artifacts. The attacker would most likely give up trying to break into your network or go back and try to create a new tool that serves the same purpose. It will be a game over for the attackers as they would need to invest some money into building a new tool (if they are capable of doing so), find the tool that has the same potential, or even gets some training to learn how to be proficient in a certain tool.

Attackers would use the utilities to create malicious macro documents (maldocs) for spearphishing attempts, a backdoor that can be used to establish C2 (Command and Control Infrastructure), any custom .EXE, and .DLL files, payloads, or password crackers.

Antivirus signatures, detection rules, and YARA rules can be great weapons for you to use against attackers at this stage.

MalwareBazaar and Malshare are good resources to provide you with access to the samples, malicious feeds, and YARA results - these all can be very helpful when it comes to threat hunting and incident response.

For detection rules, SOC Prime Threat Detection Marketplace is a great platform, where security professionals share their detection rules for different kinds of threats including the latest CVE's that are being exploited in the wild by adversaries.

Fuzzy hashing is also a strong weapon against the attacker's tools. Fuzzy hashing helps you to perform similarity analysis - match two files with minor differences based on the fuzzy hash values. One of the examples of fuzzy hashing is the usage of SSDeep; on the SSDeep official website, you can also find the complete explanation for fuzzy hashing.

TTPS(Tough)

It is not over yet. But good news, we made it to the final stage or the apex of the Pyramid of Pain!
TTPs stands for Tactics, Techniques & Procedures. This includes the whole MITRE

ATT&CK Matrix, which means all the steps taken by an adversary to achieve his goal, starting from phishing attempts to persistence and data exfiltration.

If you can detect and respond to the TTPs quickly, you leave the adversaries almost no chance to fight back. For, example if you could detect a Pass-the-Hash attack using Windows Event Log Monitoring and remediate it, you would be able to find the compromised host very quickly and stop the lateral movement inside your network. At this point, the attacker would have two options:

1. Go back, do more research and training, reconfigure their custom tools
2. Give up and find another target

Option 2 definitely sounds less time and resource-consuming.

END OF CHAPTER