Cyber Kill Chain - Defensive Approach

1. Reconnaissance

- Gathering information about the target

Passive: gathering info about the target passively about the company

- via whois, google, job listings, company website

Active: involves some level of interaction with the company

- TA will actively prob network system to look for open ports and service
    - Includes technical tools like Nmap, port scanning, banner grabbing, or vulnerability scanning (loud and obvious so they will limit scope or slow scan over time)

Defending against passive reconnaissance:

- Limiting the level of detail about the company online
    - Job postings, error messages, training, social media

Defending against active reconnaissance:

- 1. Disabling unused open ports/services
- Honeypots can divert attention & reveal what they're after and who they are
- Firewall for filtering, segmentation, monitoring
    - Most can block connections from tor networks and known proxy IP addresses > which are used during this phase to obfuscate the real IP

GOAL: exploit a weakness that can be exploited

2. Weaponization

- Finding or creating the attack to exploit vulnerability discovered in the reconnaissance phase
- Weapon of choice depends on the info they collected from the recon phase
    - ***common tools = metasploit or exploit DB***
        - Repositories for known exploits
    - **VEIL** framework: generates evasion code from malware
    - **social engineering toolkit**: if they decide to deliver malware via S.E.

Administrative Controls:

Patch Management: best defensive measures against the weaponization stage, because you can't exploit the vulnerability if there's no vulnerability to exploit

- unpatched servers are the leading cause of the vast majority of today's breaches

Macros, javascript, and browser plug-ins are all common avenues for TA to exploit.

- Disabling these alone will greatly decrease exposure

Technical Controls:

- **Anti-virus** on endpoint and perimeter to protect against known malware.
- **IPS** specifically tunned to look for exploit attempts, not just port scanning and banner grabbing
- **Email security** that includes anti-virus and anti-spyware

GOAL: select weapon(s) based on reconaissance.

3. Delivery

Varies by kind of attack:

- **Web-sites** - malicious or clean; TA can infect legit website user frequent
- **Social Media**
- **User input**: means TA has some level of interaction with a public server like a website or database
- **Email**: if TA finds a partner a company used during recon phase, they can embed malware into an order form employees are more likely to open if they phis the email to make it look like its coming from a partner
- **USB:** common, left in public areas and around employees cars

Best Security measure against delivery of the attack: user awareness

- Security training
- Phishing campaigns
    - Teaches personnel basics of good security practices

Email security - DKIM & SPF

- Email authentication methods to detect spoofed emails
- ***SPF*** ensures that emails are coming from an authorized IP of the domain
- ***DKIM*** uses digital signatures to verify authenticity

Both techniques ensure emails are coming from legitimate authorized channels.

Web Filtering: prevents a user from accessing questionable or known bad sites

DNS Filtering: blocks any DNS lookup attempt to prevent communication over any protocol

Disable USB and Limit admin privileges: prevents delivery mechanisms and malwares used

SSL accounts for majority of web and email traffic seen, so SSL inspection in all delivery channels is crucial. Otherwise we would be blind to whats passing through encrypted tunnel.

4. Exploitation

Weapon delivered and attack has been executed

Takes advantage of weakness in application or OS

Exploit could come in the form of a:

- buffer overflow
- SQL injection
- Malware undetected by anti-virus solution
- Client-side exploit executed on old version of JavaScript

Protective Measures: are limited once an attacker has been able to execute the exploit - but some do exist

Data Execution Prevention (DEP): software and hardware feature that attempts to prevent execution of code and memory where it doesnt belong

- Last line of defense against common exploit attempts

Anti-Exploit: a feature on some anti-virus solutions and monitors known applications for unusual calls to memory

- Last line of defense against common exploit attempts

When an attacker reaches this point, we rely on post-infection tools like a sandbox to detect exploits that have already been executed.

Sandbox: has some preventative capabilities

- For most network environments, there is ***patient zero***: first time an unknown file is seen on the network
    - First person to download the file would be infected because the malware analysis can take several minutes to complete
- Once sandbox determines that the file is malicious, it can clock the file and protect all other users
    - It will alert us that the patient zero is infection and we can move on towards the mediation and recovery step

GOAL: gain better access

5. Installation

Payload injected after the exploit to gain better access

GOAL: gain persistent access

- From attackers perspective, gaining better access allows them to control the victim at any point in the future, even after system has been patched and rebooted

Common payload and techniques:

- **DLL Hijacking**
- **Injecting Meterpreter**
- **Remote Access Tool** installation
- **Registry Changes** - to make program automatically startup or persistent
- **Powershell Commands** - execution, in fileless attacks

Limited protection tools exist:

Linux-based systems - can used CHROOT jail as a way to isolate processes from the rest of the system

- Limiting the amount of data the malicious file has access to

Windows-based systems - can disable PowerShell on systems that don't require it

Post-infection tools used at this stage, that monitor system files from the registry for unusual activities:

UBA/EDR solution: flags any new unauthorized program that has been installed

- also detects any changes to registry and system processes

Unauthorized changes to system processes and registries should cause a log and alert to go off. Before this stage, the team should have a SOP or plan for this type of event.

- Identifying if device is mission critical, removing device from network, changing credentials for users logged in, etc.

Once system is determined to be infected, we can begin process of restoring system to known good state.

6. Command and Control (C2)

Remote control of the system by the attacker.

System has been completely compromised and in the control of the attacker

- Their access is persistent even if they reboot or patch a vulnerability

Infected device can be used to carry out the mission or it can sit back and wait for further instructions from command and control center.

Defended tactics available to limit what they can control & detect unusual activity.

Limiting damage of a breach starts with **segmentation** - will make it harder for attacker to move laterally and easier to detect using audit logs.

- better if we can do **micro-segmentation** via **zero-trust security model**
    - This would leave infected user completely isolated on a port until they can verify the machine is clean and has been authenticated

Technical Controls:

- **NGFW** - have a database of known C2 servers. Enabling these features will help block remote access from known bad actors
    - Important to use ***layer 7 application controls*** to block commonly known remote access tools like telnet, SSH, netcat, powershell, RDP, etc > protocols that have no business leaving the network
        - If we there is a business case for using these tools, its important to lock it down to specific IP addresses
- Free/Paid DNS servers that offer botnet and C2 protection at the DNS level
    - Attackers often use ***evasion techniques*** like ***DBA*** or ***fast flux*** to generate a large number of domains that are used as rendezvous
    - blocking access to recently observed domains will stop connections to these common hubs
- attacker will almost always use encrypted connections to avoid being caught
    - If we dont do full SSL deep packert inspection, we are completely blind to any communication attemptes through that tunnel

Detections: IoCs are excellent post detective tools

- IoCs is an observed behavior by a user server that are indicative of a breach
- Observed and collected on the endpoint or could be collected by a SIEM device with an IoC feed

7. Action on Objective

Attacker executes desired actions.

Important to understand type of attacker that could be targeting the organization

Attackers moticators: money, politics, nation-state - espionage, malicious insider, lateral movement

If goal is ***data exfiltration*** - we can look into tools that prevent data from moving off of the endpoint or server

- Endpoint => DLP or UVA solutions have complementary features to detect and prevent specific files from moving off the network

Lateral Movement: common step for attacker once they gain access into system

- Begin reconnaissance stage all over again to gain information about the internal network
- This is way network segmentation between different clearance levels is important to a network design

<u>Zero Trust Security Model:</u>

- Built on the idea that eventually we're all going to fall victim ti this stage of the kill chain
- By removing the idea of trust on the inside network, we can treat all users as untrusted until proven otherwise
- Very effective at detecting infected machines and limiting damage that can be done by the attacker
- Once compromised machine is identified, we can begin incident response plan and reimage system before putting it back on the network

The kill chain is a blueprint for building a good cybersecurity program.

- By using multiple layers of security throughout each phase, we make it more and more challenging for the attacker to be successful

Dwell Time: length of time an attacker is active inside the network before being detected - average time is 191 days

MITRE ATT&CK

Attackers playbook.

Expands the last 3 steps in the kill chain > installation, C2 and action on objectives

- Where w focus detecting and responding to threats

TACTICS:

Initial Access: attacks first footprint into the network via exploited vulnerability

Execution and Persistance: involve running malicious code and trying to maintain their foothold so they can continue their access even if a system reboots.

Privilege Escalation: ivolves trying to get root or admin access on the box

Defensive Evasion: tricks attacker does to avoid getting caught

- Disabling logins or encrypting payload so they dont trigger IDS or antivirus programs

Credential Access: stealing account names and passwords

- not an end goal, but a very common step fro them to take once they've gotten this far into the network

Discovery: process of trying to understand the environment or the network

- Attacker will typically see what else they have access to via port scans or port sniffing

Lateral Movement: attacker is trying to bounce to another system from the compromised host

- More often than not, attacker will penetrate the network through the weakest link and often has to pivot or jump through multiple machines to get their end objective.

Collection: gathering any kind of data

- Screen captures, keystrokes, or data needed for other objectives

Command and Control: setting up the system to be controlled remotely

- Often disguised to look like normal HTTP traffic

Exfiltration: adversary is stealing data - in ways they won't get caught via encrypted tunnels or encrypting the file

Impact: results of the system based on what the attacker is trying to achieve

- EX: ransomware goal is to get money from the attack
    - Impact to the system is that the data is encrypted and possibly service has been stopped

Procedures = Behavior Profile: for a given attack is the **sum of the techniques** a given attack uses

- In its entiretly, it breaks down the steps the attacker took to accomplish their goal

Threat Modeling

We can work backward.

What could an attacker be after?

- Identify > Prioritize

EXAMPLE

WannaCry and NotPetya - the most devastating ransomware attacks

TA wants to install ransomware on network device = **data encrypted for impact**

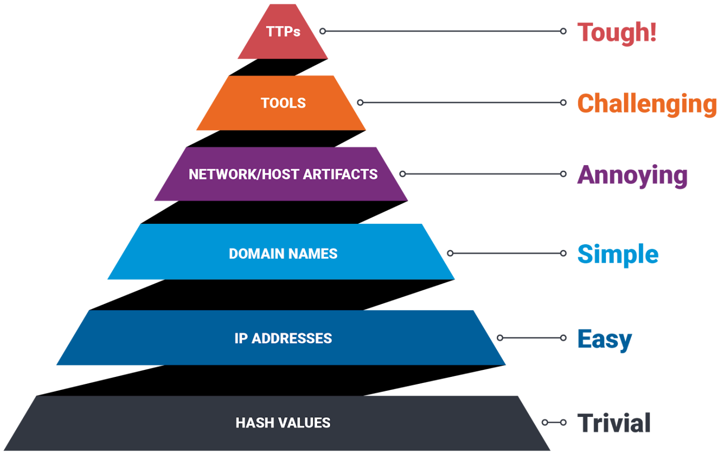
1. Select NotPetya > Layer controls > Color setup - low = 1, high =3 > Technique controls - scoring = 1
    a. Allows us to color code the matrix based on the attack

b. Allows us to see the steps taken but NotPetya
   i. We can see how they were able to get their initial access and step through various steps on their way to their end goal which was to encrypt data and hold it for ransome unless they were paid in bit coin

If we want to add multiple attacks and do advance correlation, we'll need to add another layer.

Pyramid of Pain

- Shows how easy it would be for an attack to go around the particular method that has been taken away from them.



- Blocking a file or an IP address is easy for the attacker to circumvent
- Blocking a domain name is slightly more of a burden because the attacker might have to change some line of code on their programs
- Network of host artifacts could be a safeguard on the target that is preventing them carrying out a mission
    - Blocking FTP outbound on the network
    - May prevent attacker from grabbing data, but they can circumvent that by tunneling the data out through another encrypted tunnel

- Taking away attackers tool is challenging because it would require them to come up with a new way of carrying out their end objective
    - EX: disabling powershell on work station
        - Taking away this tool from them and having other safeguards in place may prove to be challenging
- TTPs MITREATT&CK framework shows us - this the the behavior profile
    - When defending at this level, we're not going after tools which constantly change, but the attackers behavior which is more difficult to change

EX: Credential dumping technique

MIMIKATZ is popular tool used - end goal is to grab credentials

- Alerts go off if its every detected on end host or domain controller

- When attacker eventually gains access (**enumerates**) into the system, they might fail because they were blocked or endpoint alerted that someone was trying to use it

To defend against credential dumping, regardless of the tool, we can start to take measures to defend against behavior.

So instead we'll start to:

- disable settings that store credentials in memory
- lower admin debug levels
- disable password caching

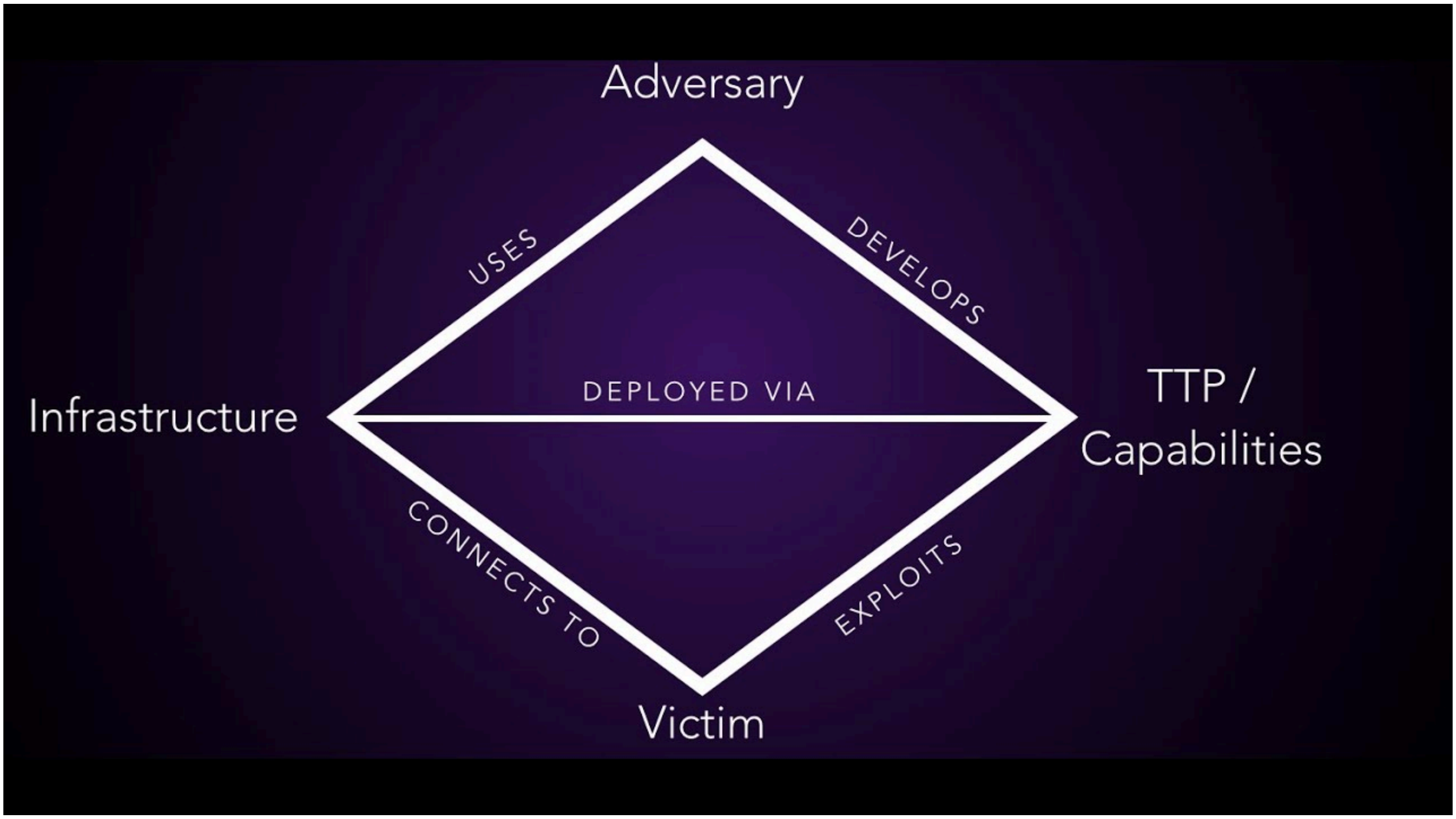
These are common techniques popular tools use to grab credentials.

This is an example of defending against behavior instead of the tool.

Implementing the ATT&CK Framework

1. Red Team Testing
2. Automated ATT&CK Simulator
3. Implement into a SIEM
4. Training

Diamond Model



Tools used for Threat Intel

VirusTotal - IP, hash, file, domain

ICANN LOOKUP - domain or IP addy

MAXMIND - list of IPs, identifies country, etc

Censys - IP, name, protocol