A Threat-based Security Monitoring Approach Using Mitre ATT&CK Matrix

Patrick Bareiß, Senior Security Research Engineer, Splunk

27th November 2019

splunk> turn data into doing™

Patrick Bareiß

Current: Senior Security Research Engineer, Splunk

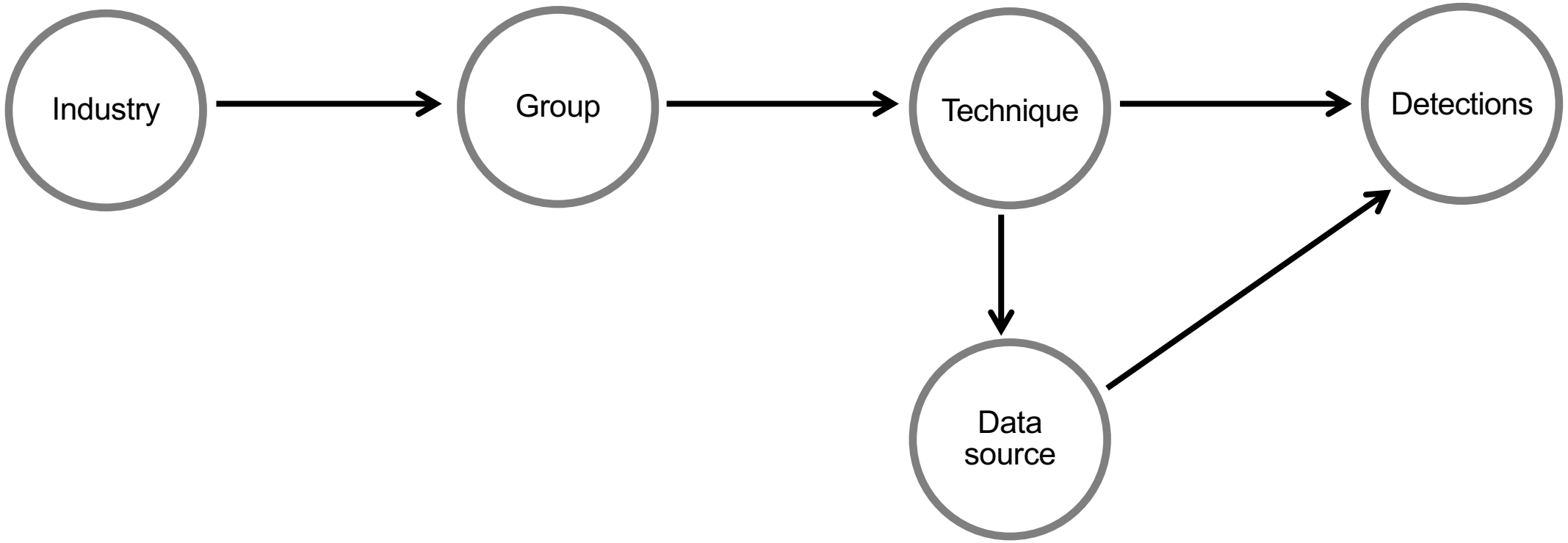
Former: Cyber Security Engineer, Airbus Defence & Space

Open Source Projects: Sigma2SplunkAlert, Sigma Hunting App for Splunk, …

Twitter: @bareiss_patrick

splunk> turn data into doing™

Goal



EVIL CORP

Financial Sector

Evil Corp

from the tv show Mr. Robot

Evil Corp is one of the largest multi-national conglomerates in the world. The company owns 70% of the global consumer credit industry. Evil Corp is targeted by many threat actors. One of them is the hacking crew fsociety.

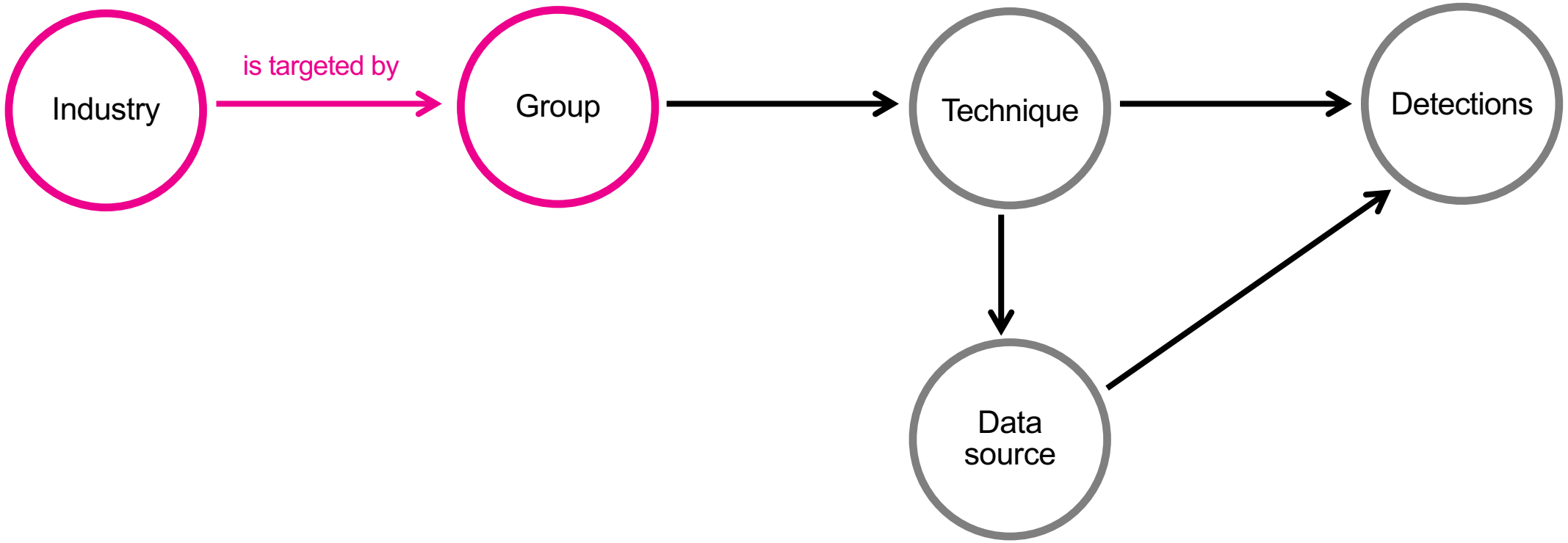
EVIL CORP

Financial Sector

Evil Corp

from the tv show Mr. Robot

Goal



Fireeye

APT10

Also known as: Menupass Team

Suspected attribution: China

Target sectors: Construction and engineering, aerospace, and telecom firms, and governments in the United States, Europe, and Japan

Overview: APT10 is a Chinese cyber espionage group that FireEye has tracked since 2009. They have historically targeted construction and engineering, aerospace, and telecom firms, and governments in the United States, Europe, and Japan. We believe that the targeting of these industries has been in support of Chinese national security goals, including acquiring valuable military and intelligence information as well as the theft of confidential business data to support Chinese corporations.

Associated malware: HAYMAKER, SNUGRIDE, BUGJUICE, QUASARRAT

Attack vectors: This recent APT10 activity has included both traditional spear phishing and access to victim's networks through managed service providers. (For more information on infection via service providers see M-Trends 2016). APT10 spear phishes have been relatively unsophisticated, leveraging .lnk files within archives, files with double extensions (e.g. [Redacted]_Group_Meeting_Document_20170222_doc_.exe) and in some cases simply identically named decoy documents and malicious launchers within the same archive. In addition to the spear phishes, FireEye ISIGHT Intelligence has observed APT10 accessing victims through global service providers.

Back to top ▵

Additional resources

Blog – APT10 Targeting Japanese Corporations Using Updated TTPs

Blog – APT10 (MenuPass Group): New Tools, Global Campaign Latest Manifestation of Longstanding Threat

Source: https://www.fireeye.com/current-threats/apt-groups.html

Crowdstrike

Meet CrowdStrike's Adversary of the Month for August: GOBLIN PANDA

August 29, 2018 Adam Meyers Research & Threat Intel

CrowdStrike® first observed GOBLIN PANDA activity in September 2013 when indicators of its activity were discovered on the network of a technology company operating in multiple sectors.

Malware variants primarily used by this actor include PlugX and HttpTunnel. This actor focuses a **significant amount of its targeting activity on entities in Southeast Asia, particularly Vietnam**. Heavy activity was observed in the late spring and early summer of 2014 when tensions between China and other Southeast Asian nations were high, due to conflict over territory in the South China Sea. **GOBLIN PANDA targets have been primarily observed in the defense, energy, and government sectors.**

CATEGORIES

- ENDPOINT PROTECTION (182)
- ENGINEERING & TECH (8)
- EXECUTIVE VIEWPOINT (100)
- FROM THE FRONT LINES (88)
- RESEARCH & THREAT INTEL (140)
- TECH CENTER (59)

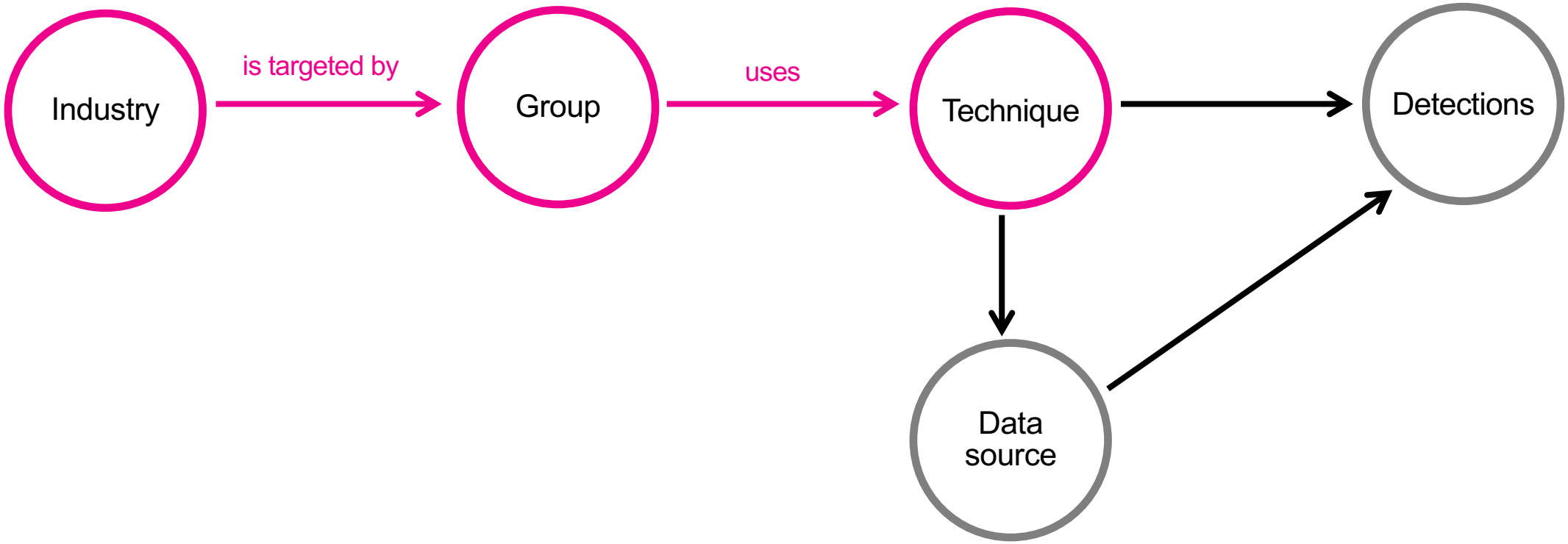
CONNECT WITH US

BREACHES

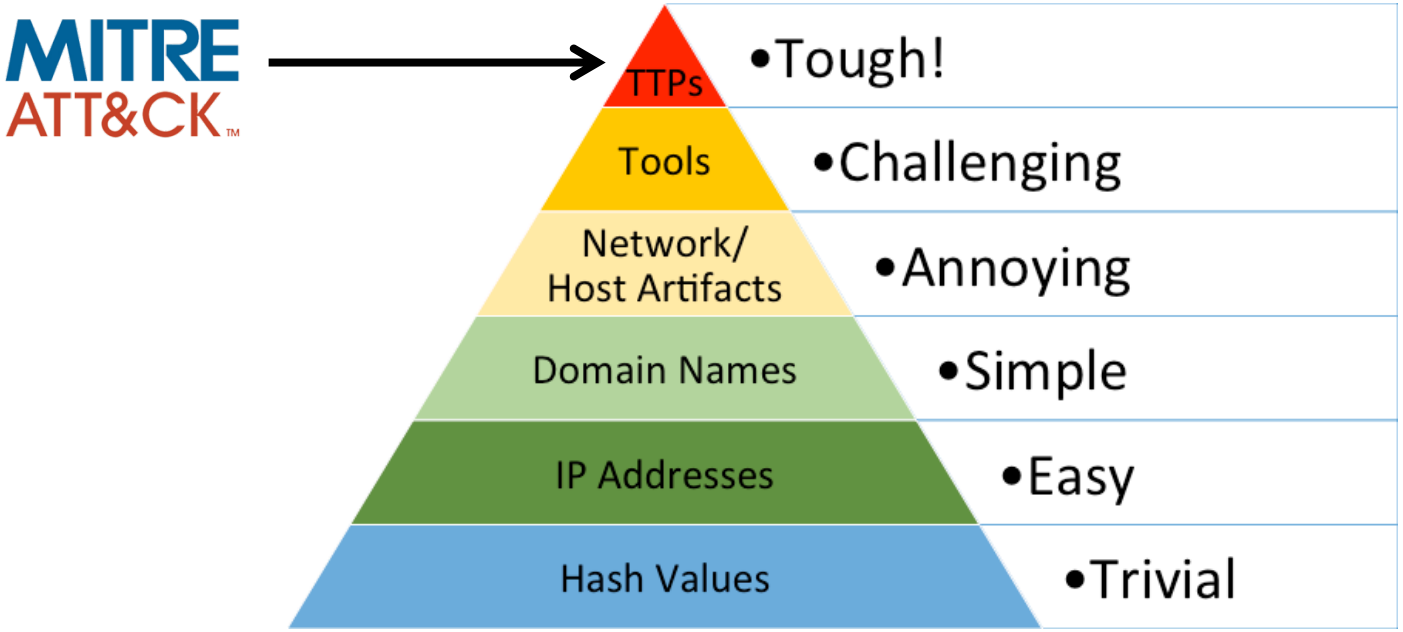
Source:

https://www.crowdstrike.com/blog/meet-crowdstrikes-adversary-of-the-month-for-august-goblin-panda/

Goal

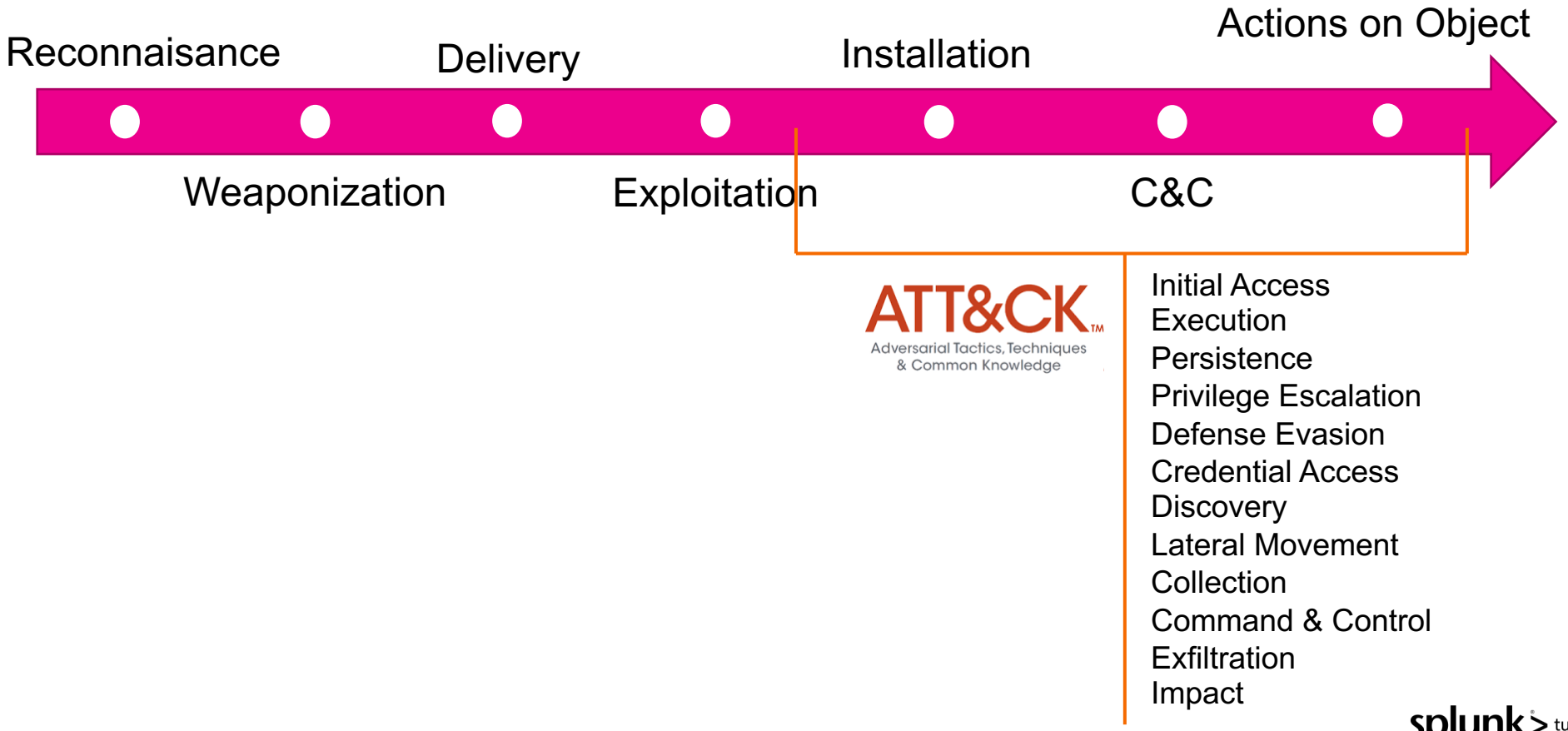


Biancos Pyramid of Pain



Source: http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html

Cyber Kill Chain VS. Mitre ATT&CK



Mitre ATT&CK Matrix

- Tactics: the adversary technical goals
- Techniques: how the goals are achieved

Initial Access	Execution	Persisten					Collection	Command and Control	Exfiltration	Impact
Drive-by Compromise	AppleScript	.bash_profile an					udio Capture	Commonly Used Port	Automated Exfiltration	Data Destruction
Exploit Public-Facing Application	CMSTP	Accessibility F					nated Collection	Communication Through Removable Media	Data Compressed	Data Encrypted for Impact
External Remote Services	Command-Line Interface	Account Manip					ipboard Data	Connection Proxy	Data Encrypted	Defacement
Hardware Additions	Compiled HTML File	AppCert D					Data Staged	Custom Command and Control Protocol	Data Transfer Size Limits	Disk Content Wipe
Replication Through Removable Media	Control Panel Items	AppInit DL					from Information Repositories	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol	Disk Structure Wipe
Spearphishing Attachment	Dynamic Data Exchange	Application Sh					ta from Local System	Data Encoding	Exfiltration Over Command and Control Channel	Endpoint Denial of Service
Spearphishing Link	Execution through API	Authentication					a from Network Shared Drive	Data Obfuscation	Exfiltration Over Other Network Medium	Firmware Corruption
Spearphishing via Service	Execution through Module Load	BITS Job					from Removable Media	Domain Fronting	Exfiltration Over Physical Medium	Inhibit System Recovery
Supply Chain Compromise	Exploitation for Client Execution	Bootkit					ail Collection	Domain Generation Algorithms	Scheduled Transfer	Network Denial of Service
Trusted Relationship	Graphical User Interface	Browser Exte					put Capture	Fallback Channels		Resource Hijacking
Valid Accounts	InstallUtil	Change Default File					in the Browser	Multi-Stage Channels		Runtime Data Manipulation
	LSASS Driver	Component Fi					reen Capture	Multi-hop Proxy		Service Stop
	Launchctl	Component Object M					deo Capture	Multiband Communication		Stored Data Manipulation
	Local Job Scheduling	Create Acc						Multilayer Encryption		Transmitted Data Manipulation
	Mshta	DLL Search Order						Port Knocking		
	PowerShell	Dylib Hijacking	Path Interception	Decobfuscate/Decode Files or Information	Password Filter DLL	System Information Discovery	Windows Admin Shares	Remote Access Tools		
	Regsvcs/Regasm	External Remote Services	Plist Modification	Disabling Security Tools	Private Keys	System Network Configuration Discovery	Windows Remote Management	Remote File Copy		
						System Network		Standard Application Layer		

Spearphishing Link

Spearphishing with a link is a specific variant of spearphishing. It is different from other forms of spearphishing in that it employs the use of links to download malware contained in email, instead of attaching malicious files to the email itself, to avoid defenses that may inspect email attachments.

All forms of spearphishing are electronically delivered social engineering targeted at a specific individual, company, or industry. In this case, the malicious emails contain links. Generally, the links will be accompanied by social engineering text and require the user to actively click or copy and paste a URL into a browser, leveraging User Execution. The visited website may compromise the web browser using an exploit, or the user will be prompted to download applications, documents, zip files, or even executables depending on the pretext for the email in the first place. Adversaries may also include links that are intended to interact directly with an email reader, including embedded images intended to exploit the end system directly or verify the receipt of an email (i.e. web bugs/web beacons).

ID: T1192

Tactic: Initial Access

Platform: Windows, macOS, Linux

Data Sources: Packet capture, Web proxy, Email gateway, Detonation chamber, SSL/TLS inspection, DNS records, Mail server

CAPEC ID: CAPEC-163

Version: 1.0

Examples

Name	Description
APT28	APT28 sent spearphishing emails which used a URL-shortener service to masquerade as a legitimate service and to redirect targets to credential harvesting sites.[1]
APT29	APT29 has used spearphishing with a link to trick victims into clicking on a link to a zip file containing malicious files.[2]
APT32	APT32 has sent spearphishing emails containing malicious links.[3][4]

Example Group: Turla

MITRE ATT&CK

Matrices | Tactics | Techniques | Mitigations | Groups | Software | Resources | Blog | Contribute | Search site

GROUPS

- Overview
- admin@338
- APT1
- APT12
- APT16
- APT17
- APT18
- APT19
- APT28
- APT29
- APT3
- APT30
- APT32
- APT33
- APT37
- APT38
- APT39
- APT41

Home > Groups > Turla

Turla

Turla is a Russian-based threat group that has infected victims in over 45 countries, spanning a range of industries including government, embassies, military, education, research and pharmaceutical companies since 2004. Heightened activity was seen in mid-2015. Turla is known for conducting watering hole and spearphishing campaigns and leveraging in-house tools and malware. Turla's espionage platform is mainly used against Windows machines, but has also been seen used against macOS and Linux machines. [1] [2] [3] [4]

ID: G0010

Associated Groups: Waterbug, WhiteBear, VENOMOUS BEAR, Snake, Krypton

Contributors: Edward Millington

Version: 1.2

Associated Group Descriptions

Name	Description
Waterbug	Based similarity in TTPs and malware used, Turla and Waterbug appear to be the same group.[12]
WhiteBear	WhiteBear is a designation used by Securelist to describe a cluster of activity that has overlaps with activity described by others as Turla, but appears to have a separate focus. [14]
VENOMOUS BEAR	[3]
Snake	[3][5]
Krypton	[3]

APT: Turla

- Tactics: the adversary technical goals
- Techniques: how the goals are achieved

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command and Control	Exfiltration	Impact
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Data Destruction
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	BITS Jobs	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Communication Through Removable Media	Data Compressed	Data Encrypted for Impact
External Remote Services	Command-Line Interface	Account Manipulation	AppCert DLLs	Binary Padding	Brute Force	Browser Bookmark Discovery	Distributed Component Object Model	Clipboard Data	Connection Proxy	Data Encrypted	Defacement
Hardware Additions	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	Domain Trust Discovery	Exploitation of Remote Services	Data Staged	Custom Command and Control Protocol	Data Transfer Size Limits	Disk Content Wipe
Replication Through Removable Media	Control Panel Items	AppInit DLLs	Application Shimming	CMSTP	Credentials in Files	File and Directory Discovery	Logon Scripts	Data from Information Repositories	Custom Cryptographic Protocol	Exfiltration Over Alternative Protocol	Disk Structure Wipe
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Bypass User Account Control	Clear Command History	Credentials in Registry	Network Service Scanning	Pass the Hash	Data from Local System	Data Encoding	Exfiltration Over Command and Control Channel	Endpoint Denial of Service
Spearphishing Link	Execution through API	Authentication Package	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Network Share Discovery	Pass the Ticket	Data from Network Shared Drive	Data Obfuscation	Exfiltration Over Other Network Medium	Firmware Corruption
Spearphishing via Service	Execution through Module Load	BITS Jobs	Dylib Hijacking	Compile After Delivery	Forced Authentication	Network Sniffing	Remote Desktop Protocol	Data from Removable Media	Domain Fronting	Exfiltration Over Physical Medium	Inhibit System Recovery
Supply Chain Compromise	Exploitation for Client Execution	Bootkit	Exploitation for Privilege Escalation	Compiled HTML File	Hooking	Password Policy Discovery	Remote File Copy	Email Collection	Domain Generation Algorithms	Scheduled Transfer	Network Denial of Service
Trusted Relationship	Graphical User Interface	Browser Extensions	Extra Window Memory Injection	Component Firmware	Input Capture	Peripheral Device Discovery	Remote Services	Input Capture	Fallback Channels		Resource Hijacking
Valid Accounts	InstallUtil	Change Default File Association	File System Permissions Weakness	Component Object Model Hijacking	Input Prompt	Permission Groups Discovery	Replication Through Removable Media	Man in the Browser	Multi-Stage Channels		Runtime Data Manipulation
	LSASS Driver	Component Firmware	Hooking	Control Panel Items	Kerberoasting	Process Discovery	SSH Hijacking	Screen Capture	Multi-hop Proxy		Service Stop
	Launchctl	Component Object Model Hijacking	Image File Execution Options Injection	DCShadow	Keychain	Query Registry	Shared Webroot	Video Capture	Multiband Communication		Stored Data Manipulation
	Local Job Scheduling	Create Account	Launch Daemon	DLL Search Order Hijacking	LLMNR/NBT-NS Poisoning and Relay	Remote System Discovery	Taint Shared Content		Multilayer Encryption		Transmitted Data Manipulation
	Mshta	DLL Search Order Hijacking	New Service	DLL Side-Loading	Network Sniffing	Security Software Discovery	Third-party Software		Port Knocking		
	PowerShell	Dylib Hijacking	Path Interception	Deobfuscate/Decode Files or Information	Password Filter DLL	System Information Discovery	Windows Admin Shares		Remote Access Tools		
	Regsvcs/Regasm	External Remote Services	Plist Modification	Disabling Security Tools	Private Keys	System Network Configuration Discovery	Windows Remote Management		Remote File Copy		
						System Network			Standard Application Layer		

Techniques used by APT Groups in ATT&CK

MITRE ATT&CK™ Navigator

Detections x

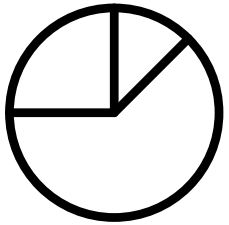
selection controls | layer controls | technique controls

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Command And Control	Exfiltration	Impact
11 items	34 items	62 items	32 items	69 items	21 items	23 items	18 items	13 items	22 items	9 items	16 items
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Commonly Used Port	Automated Exfiltration	Account Access Removal
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Communication Through Removable Media	Data Compressed	Data Destruction
External Remote Services	Command-Line Interface	Account Manipulation	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Component Object Model and Distributed COM	Clipboard Data	Connection Proxy	Data Encrypted	Data Encrypted for Impact
Hardware Additions	Compiled HTML File	AppCert DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	Domain Trust Discovery	Exploitation of Remote Services	Data from Information Repositories	Custom Command and Control Protocol	Data Transfer Size Limits	Defacement
Replication Through Removable Media	Component Object Model and Distributed COM	AppInit DLLs	Application Shimming	Clear Command History	Credentials from Web Browsers	File and Directory Discovery		Data from Local System	Custom Cryptographic	Exfiltration Over Alternative Protocol	Disk Content Wipe
Spearphishing Attachment	Control Panel Items	Application Shimming	Bypass User Account Control	CMSTP	Credentials in Files	Network Service Scanning					Disk Structure Wipe
Spearphishing Link	Dynamic Data Exchange	Authentication Package	DLL Search Order Hijacking	Code Signing	Credentials in Registry						Endpoint Denial of Service
Spearphishing via Service	Execution through API	BITS Jobs	Dylib Hijacking	Compile After Delivery	Exploitation f Credential Ac						Firmware Corruption
Supply Chain Compromise	Execution through Module Load	Bootkit	Elevated Execution with Prompt	Compiled HTML File	Forced Authenticatio						Inhibit System Recove
Trusted Relationship	Exploitation for Client Execution	Browser Extensions	Emond	Component Firmware	Hooking						Network Denial of Service
Valid Accounts	Graphical User Interface	Change Default File Association	Exploitation for Privilege Escalation	Component Object Model Hijacking	Input Capture						Resource Hijacking
	InstallUtil	Component Firmware	Extra Window Memory Injection	Connection Proxy	Input Prompt						Data Manipulation
	Launchctl	Component Object Model Hijacking	File System Permissions Weakness	Control Panel Items	Kerberoasting						Service Stop
	Local Job Scheduling	Create Account	Hooking	DCShadow	Keychain						Stored Data Manipulation
	LSASS Driver	DLL Search Order Hijacking	Image File Execution Options Injection	Deobfuscate/Decode Files or Information	LLMNR/NBT-NS Poisoning and Relay						System
	Mshta	Dylib Hijacking	Launch Daemon	Disabling Security Tools	Network Sniffing	Software Discovery	Shared Webroot	Video Capture	Multilayer Encryption		
	PowerShell	Emond	New Service	DLL Search Order Hijacking	Password Filter DLL	System Information Discovery	SSH Hijacking		Port Knocking		
	Regsvcs/Regasm	External Remote Services	Parent PID Spoofing	DLL Side-Loading	Private Keys	System Network Configuration Discovery	Taint Shared Content		Remote Access Tools		
	Regsvr32	File System Permissions Weakness	Path Interception	Execution Guardrails	Securityd Memory	System Network Connections Discovery	Third-party Software		Remote File Copy		
	Rundll32	Hidden Files and Directories	Plist Modification	Exploitation for Defense Evasion	Steal Web Session Cookie	System Owner/User Discovery	Windows Admin Shares		Standard Application Layer Protocol		
	Scheduled Task	Hooking		Extra Window Memory Injection	Two-Factor Authentication Interception	System Service Discovery	Windows Remote Management		Standard Cryptographic Protocol		
	Scripting	Hypervisor		File and Directory Permissions Modification		System Time Discovery			Standard Non-Application Layer Protocol		
	Service Execution	Image File Execution		File Deletion		Virtualization/Sandbox Evasion			Uncommonly Used Port		
	Signed Binary Proxy			File System Logical Offsets							

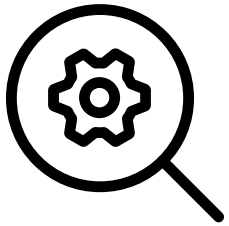
I can't spend 3 million Euros on writing detections for all these different attacks.

What techniques should we focus on?

Probability



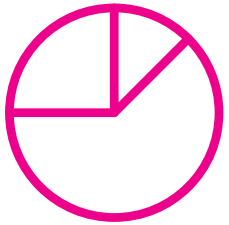
Detections



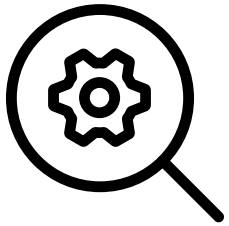
Log Sources

What techniques should we focus on?

Probability

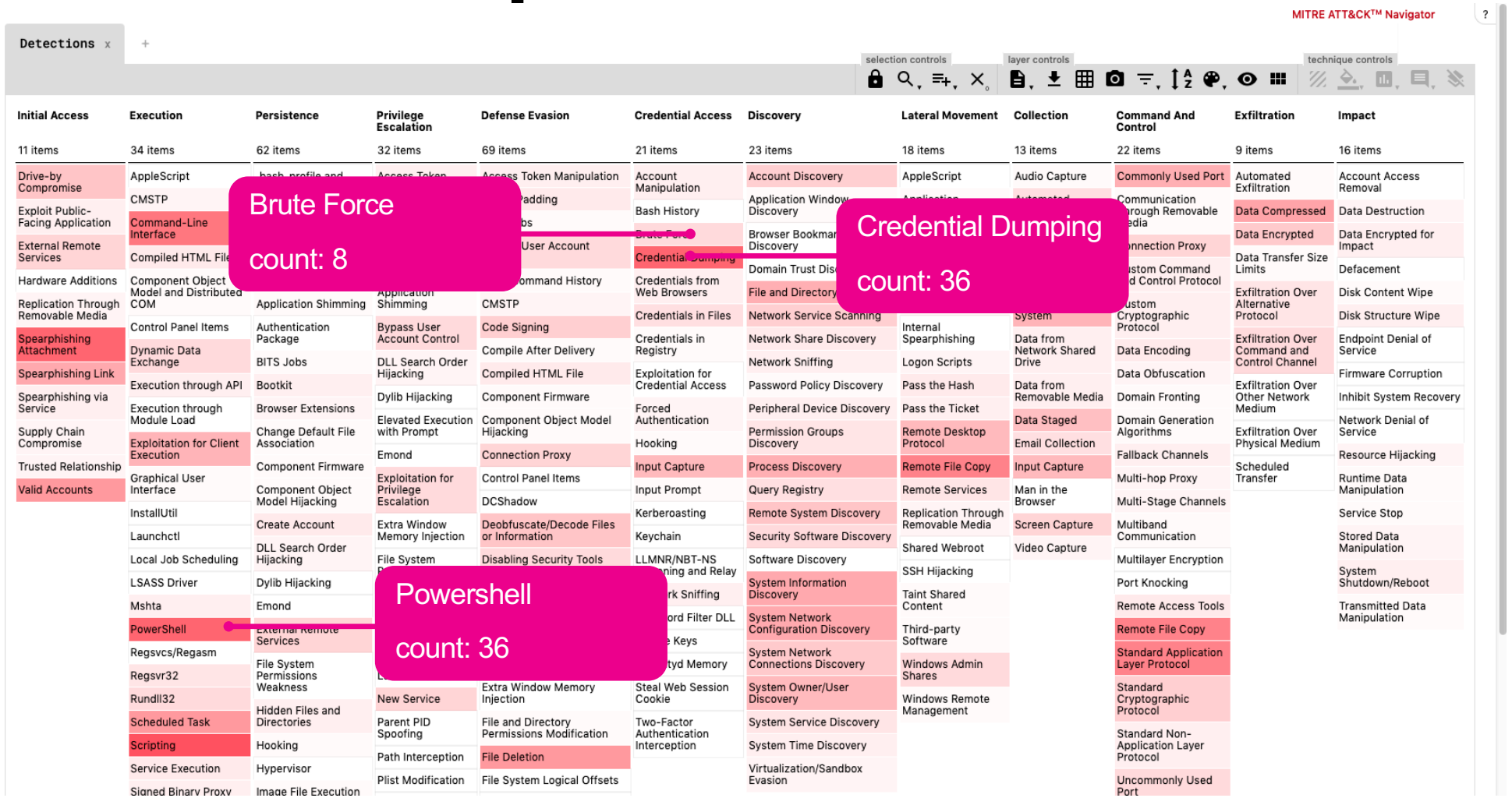


Detections



Log Sources

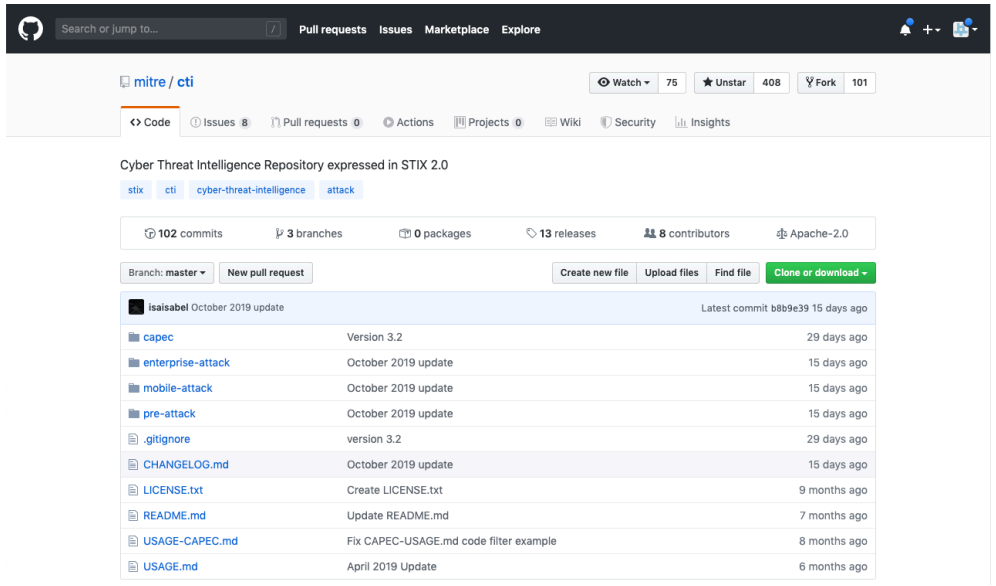
All APT Groups in ATT&CK



ATT&CK Navigator:

https://github.com/mitre-attack/attack-navigator

Mitre Cyber Threat Intelligence (CTI)



Mitre CTI:

https://github.com/mitre/cti

Mitre Object	STIX 2 Object
ATT&CK Technique	
ATT&CK Tactic	  
ATT&CK Group	
ATT&CK Software	 
ATT&CK Mitigation	

Mitre CTI

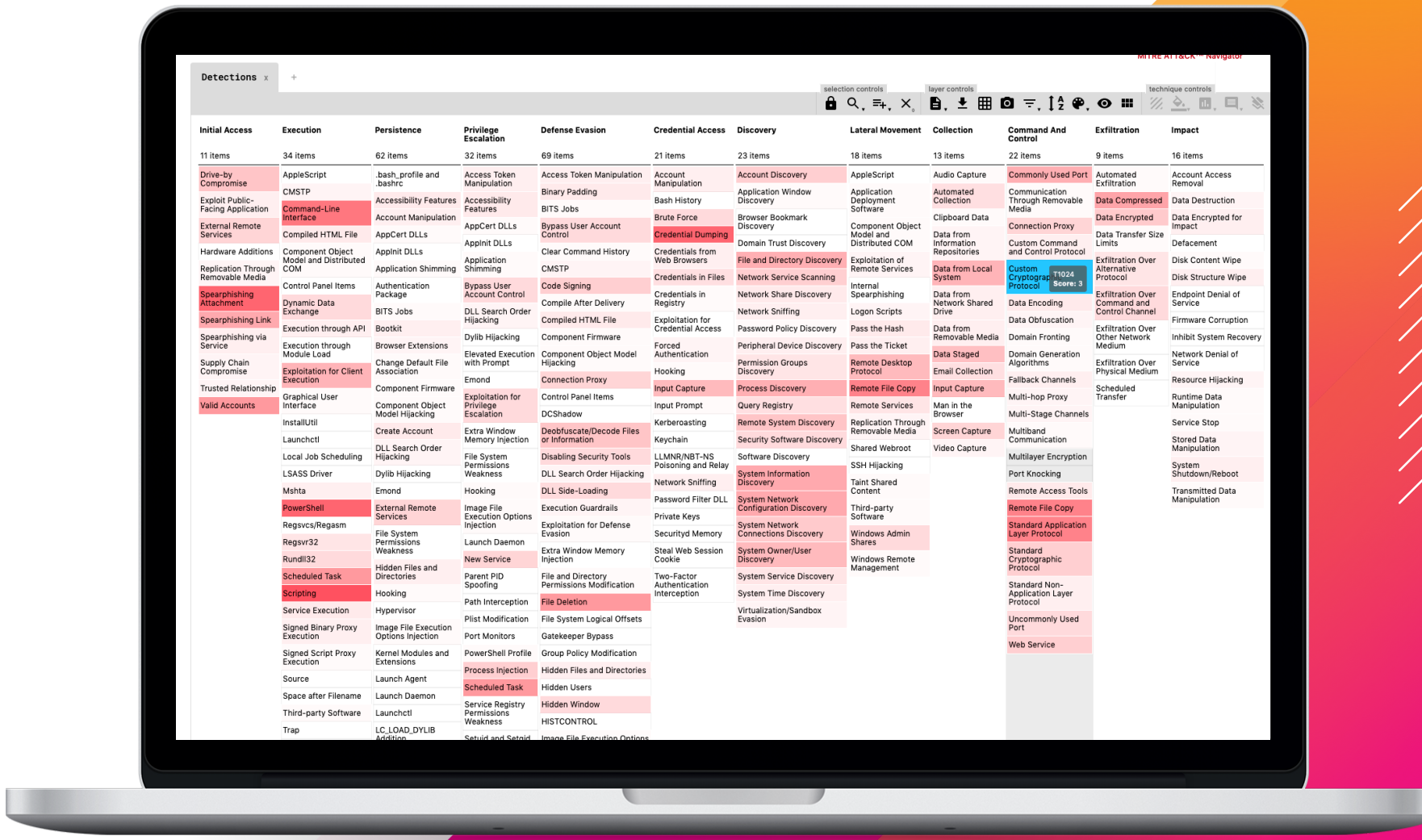
STIX 2 format consists of a machine readable JSON files, which makes it easy to use it in your own scripts.

```python
def get_all_techniques(src):
    filt = [Filter('type', '=', 'attack-pattern')]
    return src.query(filt)


def get_group_by_alias(src, alias):
    return src.query([
        Filter('type', '=', 'intrusion-set'),
        Filter('aliases', '=', alias)
    ])


def get_all_groups(src):
    filt = [Filter('type', '=', 'intrusion-set')]
    return src.query(filt)


def get_technique_by_group(src, stix_id):
    relations = src.relationships(stix_id, 'uses', source_only=True)
    return src.query([
        Filter('type', '=', 'attack-pattern'),
        Filter('id', 'in', [r.target_ref for r in relations])
    ])
```

splunk> turn data into doing

Mitre CTI

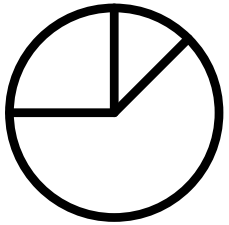
in order to create your own ATT&CK Navigator overlays.



splunk > turn data into doing

What techniques should we focus on?

Probability



Detections

Log Sources

Splunk Security Content

- open source repository containing 200+ Splunk detections
- detections, investigations, responses, baselines are combined to analytics stories.
- Available as GitHub repository and API
- Mapped to Mitre ATT&CK Matrix
- Mapped to CIM data model / log sources

security-content

branch	build status
develop	PASSED
master	PASSED

Security Content:

https://github.com/splunk/security-content

Splunk Security Content - Example

```
search: '| tstats summariesonly=true count min(_time) as firstTime max(_time) as lastTime
  from datamodel=Endpoint.Processes where Processes.process_name=reg.exe Processes.process=*save*
  (Processes.process=*HKEY_LOCAL_MACHINE\\Security* OR Processes.process=*HKEY_LOCAL_MACHINE\\SAM* OR
  Processes.process=*HKEY_LOCAL_MACHINE\\System* OR Processes.process=*HKLM\\Security* OR
  Processes.process=*HKLM\\System* OR Processes.process=*HKLM\\SAM*)
  by Processes.user Processes.process_name Processes.process Processes.dest
  | `drop_dm_object_name(Processes)` | `ctime(firstTime)`| `ctime(lastTime)`'
```

Atomic Test #4 - Registry dump of SAM, creds, and secrets

Local SAM (SAM & System), cached credentials (System & Security) and LSA secrets (System & Security) can be enumerated via three registry keys. Then processed locally using https://github.com/Neohapsis/creddump7

Supported Platforms: Windows

Run it with `command_prompt`! Elevation Required (e.g. root or admin)

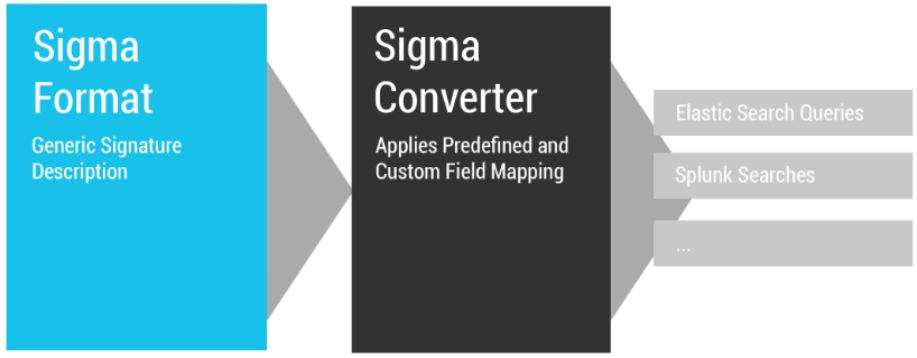
```
reg save HKLM\sam sam
reg save HKLM\system system
reg save HKLM\security security
```

Atomic Red Team:

https://github.com/redcanaryco/atomic-red-team

Sigma

- Generic and open signature format with 200+ open source detections
- Supporting different SIEM systems
- Focus on simplicity
- Mapped to Mitre ATT&CK Matrix
- Mapped to log sources



Sigma:

https://github.com/Neo23x0/sigma

Sigma - Example

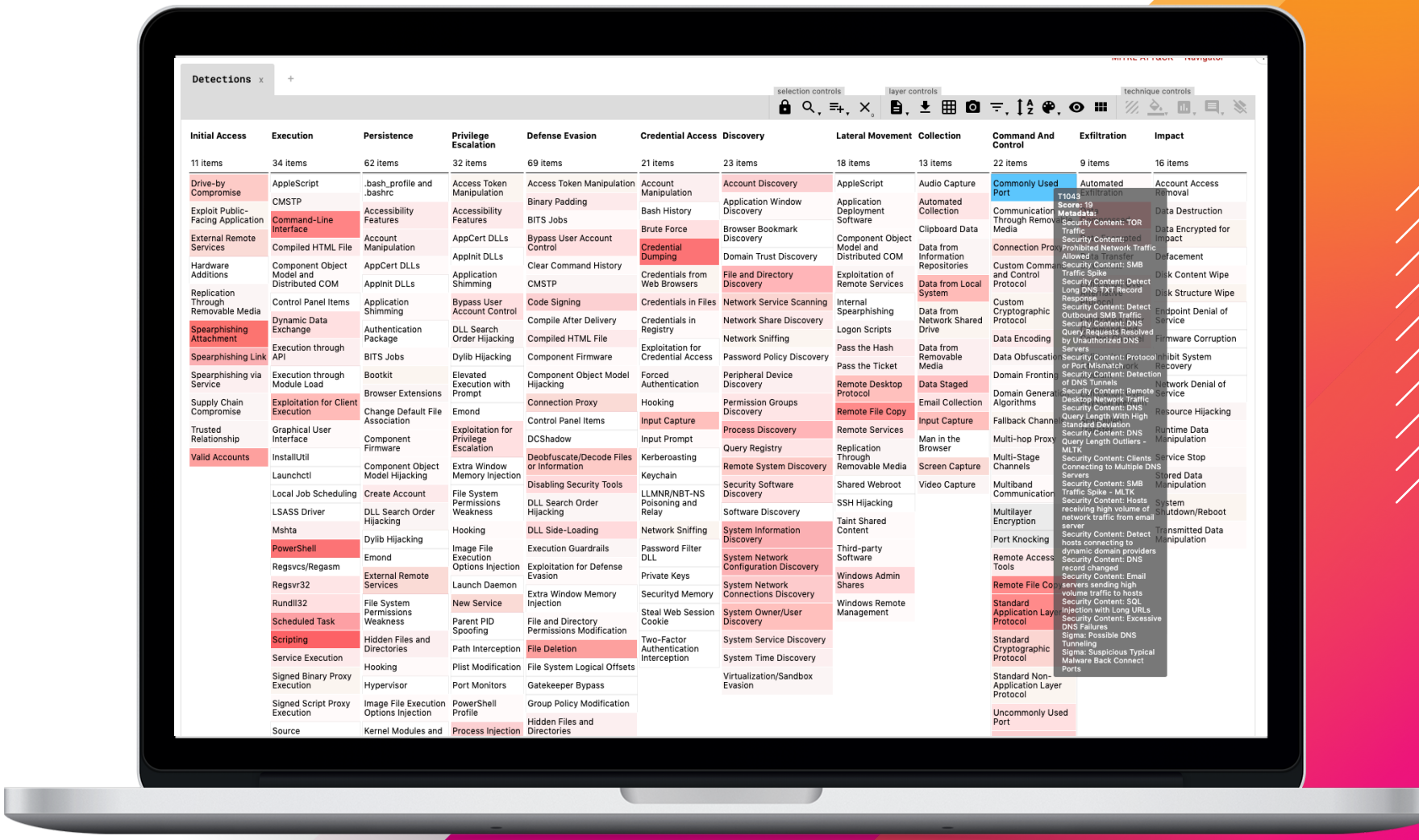
```yaml
title: Security Eventlog Cleared
description: Some threat groups tend to delete the local 'Security' Eventlog using certain utitlities
tags:
    - attack.defense_evasion
    - attack.t1070
author: Florian Roth
logsource:
    product: windows
    service: security
detection:
    selection:
        EventID:
            - 517
            - 1102
    condition: selection
falsepositives:
    - Rollout of log collection agents (the setup routine often includes a reset of the local Eventlog)
    - System provisioning (system
level: high
```

sigmac win_susp_security_eventlog_cleared.yml --target splunk

source = “WinEventLog:Security” (EventID = 517 OR EventID = 1102)

Detections

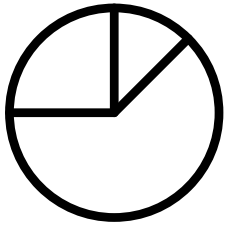
a little bit more python coding.



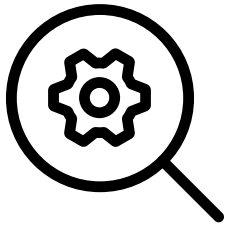
splunk > turn data into doing

What techniques should we focus on?

Probability



Detections



Log Sources

Evil Corp Log Sources

40%	Firewall Logs	> 5% Detections
18%	Windows Server Logs	50% Detecti
12%	Proxy Logs	15% Detecti
10%	Router & Switch Logs	> 1% Detections
10%	Linux Server Logs	10% Detections
10%	Other	

40%	Windows Server Logs
	Proxy Lo
	DNS Logs
10%	Linux S
10%	
10%	

Log Sources based on Detection Strategy. Makes Sense.

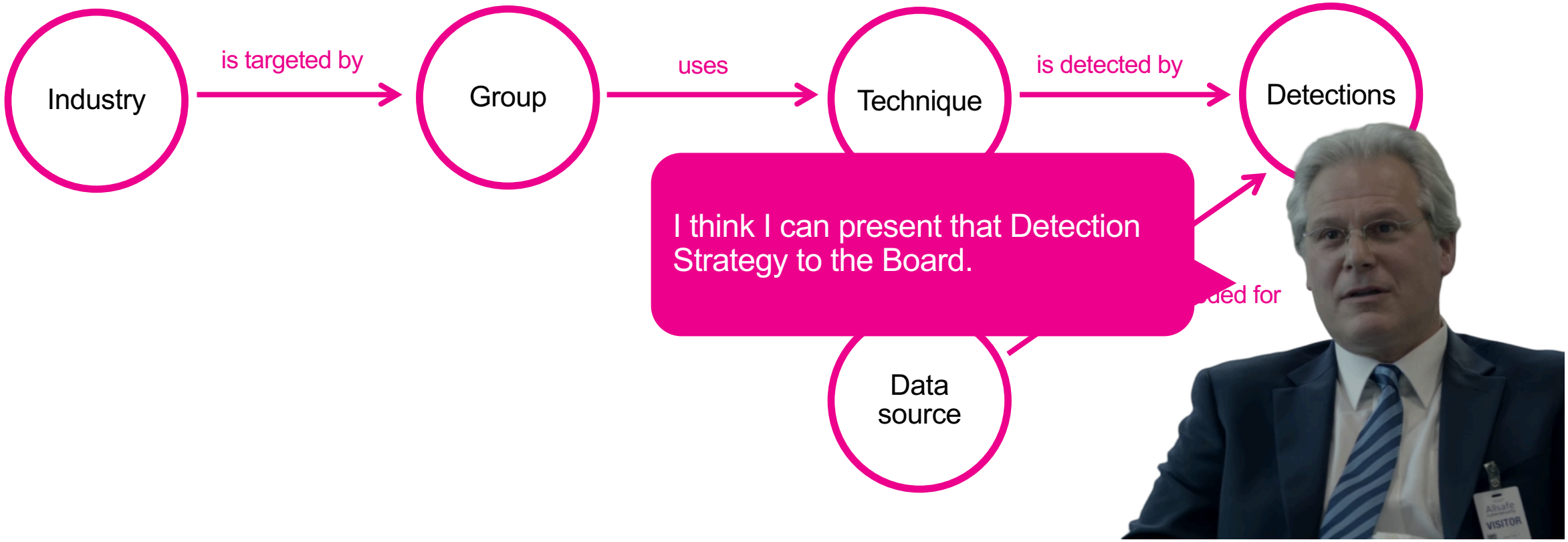
Log Sources

Log Source	Benefit	Volume
Windows Logs	**System:** service installs **Application:** App Crashes & AV Events **Security:** Logins, Group Management, … **Sysmon:** Parent-Child relation, Registry Keys, Network Connections, WMI, Named Pipes, Image/Driver Loads, … **Powershell:** Executed PS scripts, Obfuscation, …	Low to High
Proxy Logs	C2 addresses malicious UserAgents malicious URL patterns contain stage2 payload downloads	Medium

Log Sources

Log Source	Benefit	Volume
DNS Logs	C2 domains DNS tunnels exfiltration over DNS	Medium to High
Linux Server Logs	history: executed shell commands, privilege escalation, information gathering, … auditd: alter bash profile, webshell rce, … osquery: process information, socket auditing, authentication events, …	Low to Medium
AV Logs	Indicators of Threat Group activities Security Alerts	Low to Medium

Goal

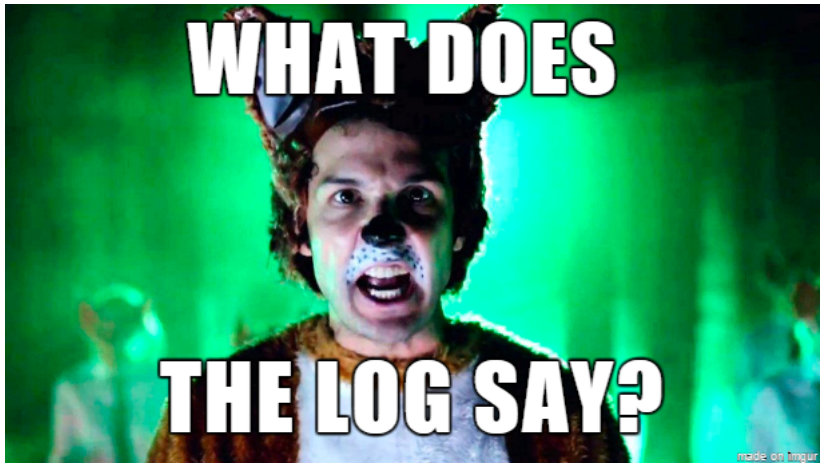


Key Takeaways

Analyze Threat Landscape to define Detection Strategy

Use Open Source Detections (It's for free)

Focus on Log Sources based on your detections (and not the other way around)



Thank You

splunk> turn data into doing™

Questions?