Twitter: @aboutsecurity

Presentation based on SEC530: Defensible Security Architecture and Engineering

SANS

# Prioritizing your Threat Hunting & Blue Teaming Strategy with MITRE ATT&CK Analytics

# About Me

**Ismael Valenzuela**

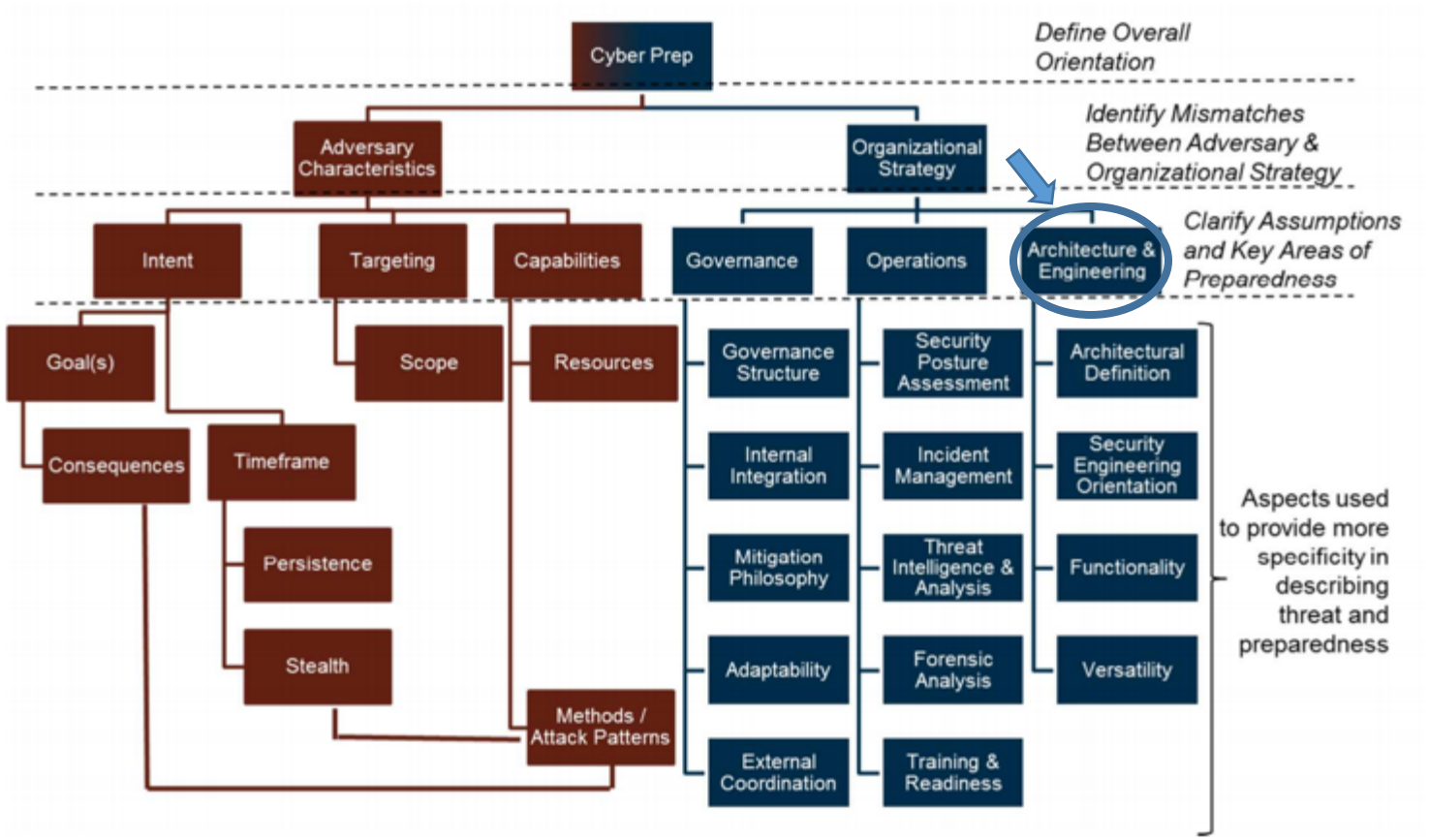
- SEC530 Co-author
- 10 yrs teaching for SANS internationally DFIR, CyberDefense & BlueTeam curriculum
- GSE # 132, plus 11 other GIAC certifications
- Sr. Principal Engineer at McAfee
- +20 yrs 'Defending All The Things!'
- **Twitter**: @aboutsecurity


MITRE WHAT?!

# Think Red, Act Blue

# Practical Threat Modeling: Purple Teaming

- We need to determine the characteristics of the adversaries which could be expected to target our systems so we can define and implement effective architectures and controls
- Purple teaming (**red** + **blue**) facilitate this by **working together** through simulation of specific threat scenarios




*MITRE Cyber Prep 2.0, May 2017*

# Where do we start?

Use case: **HACME CATS**

Biz: Online retailer – Organic food for cats

Threat: FIN-Dogs (super-scary-APT)

Tools:

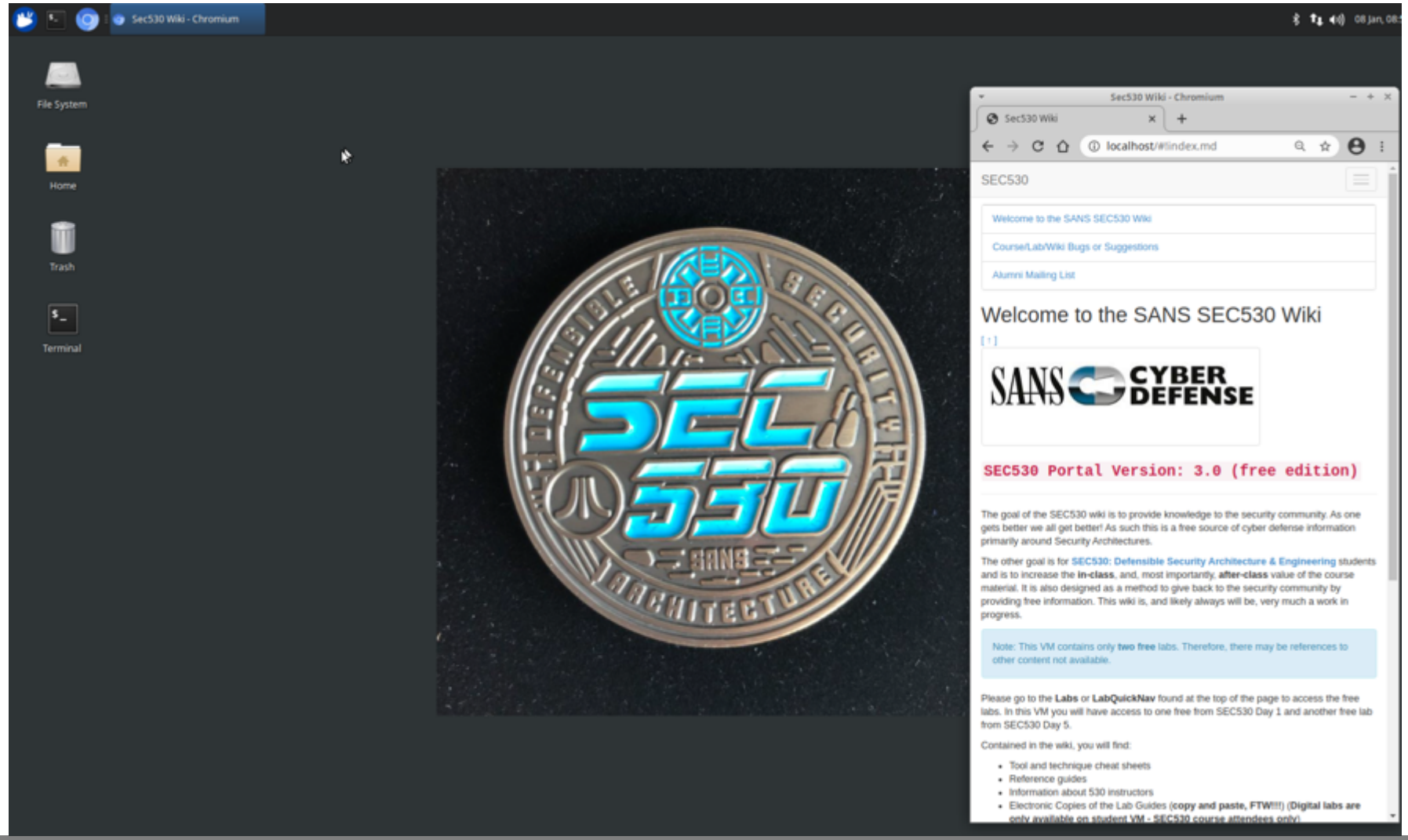
MITRE ATT&CK

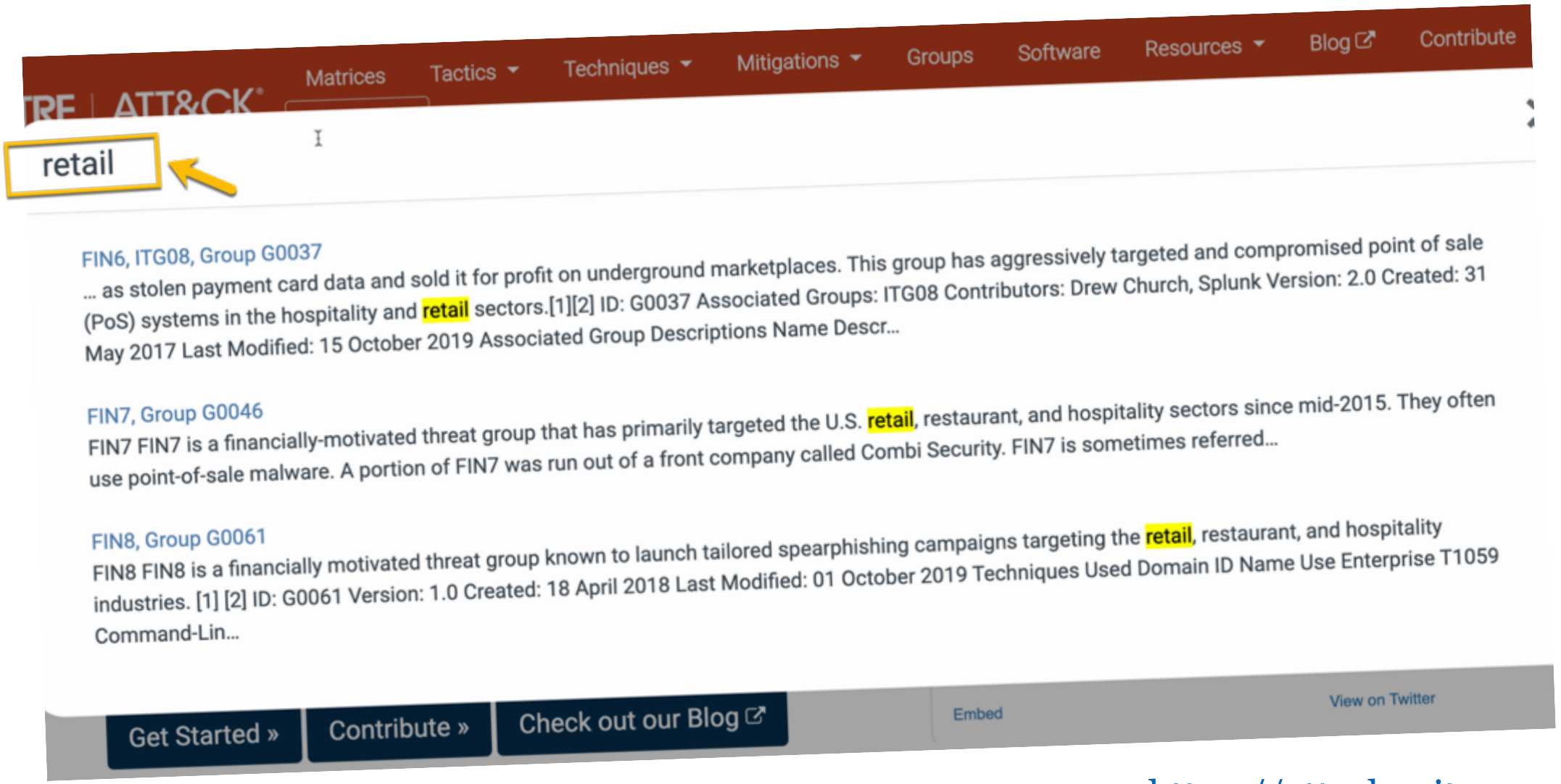
- https://attack.mitre.org/

ATT&CK Navigator

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

# Use the FREE Security 530 VM and included Wiki with sample Labs






https://attack.mitre.org/

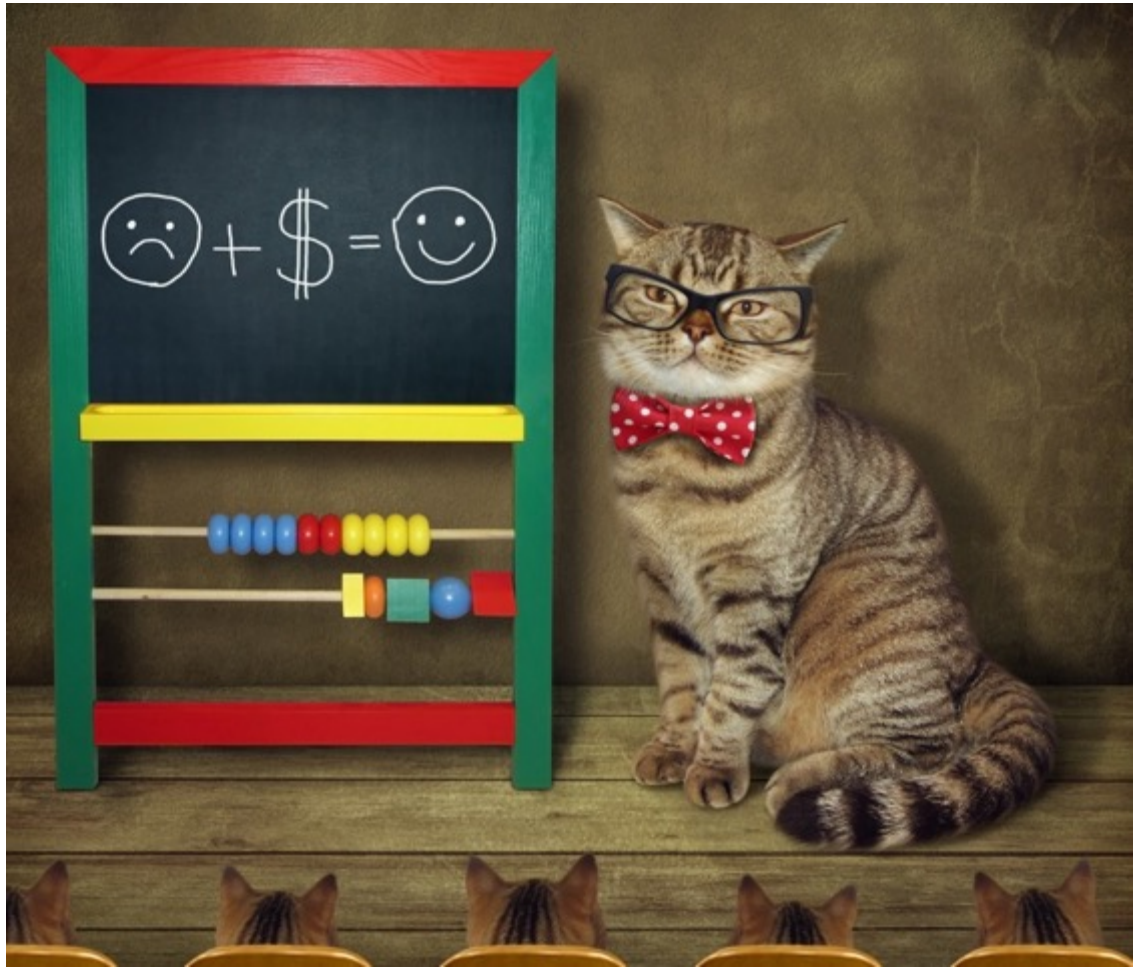

https://mitre-attack.github.io/attack-navigator/enterprise/#

# Let's play with some analytics

ATT&CK scripts:
https://github.com/mitre-attack/attack-scripts

Jupyter notebook:
https://mybinder.org/v2/gh/mitre-attack/attack-scripts/master

ATT&CK Python client:
https://github.com/hunters-forge/ATTACK-Python-Client



# Jupyter Notebooks - https://mybinder.org/v2/gh/mitre-attack/attack-scripts/master

## Data Source Investigation

Let's use the ATT&CK Python Client to manually examine the techniques, list the data sources, and build a heatmap out of our selected sources.

If you're looking for less development or a more in-depth and finely-grained dive, check out:

- DeTTACK
- AttackDatamap

*Consider: What have you used to track data sources? What has worked well, and what has not worked so well?*

In [ ]:

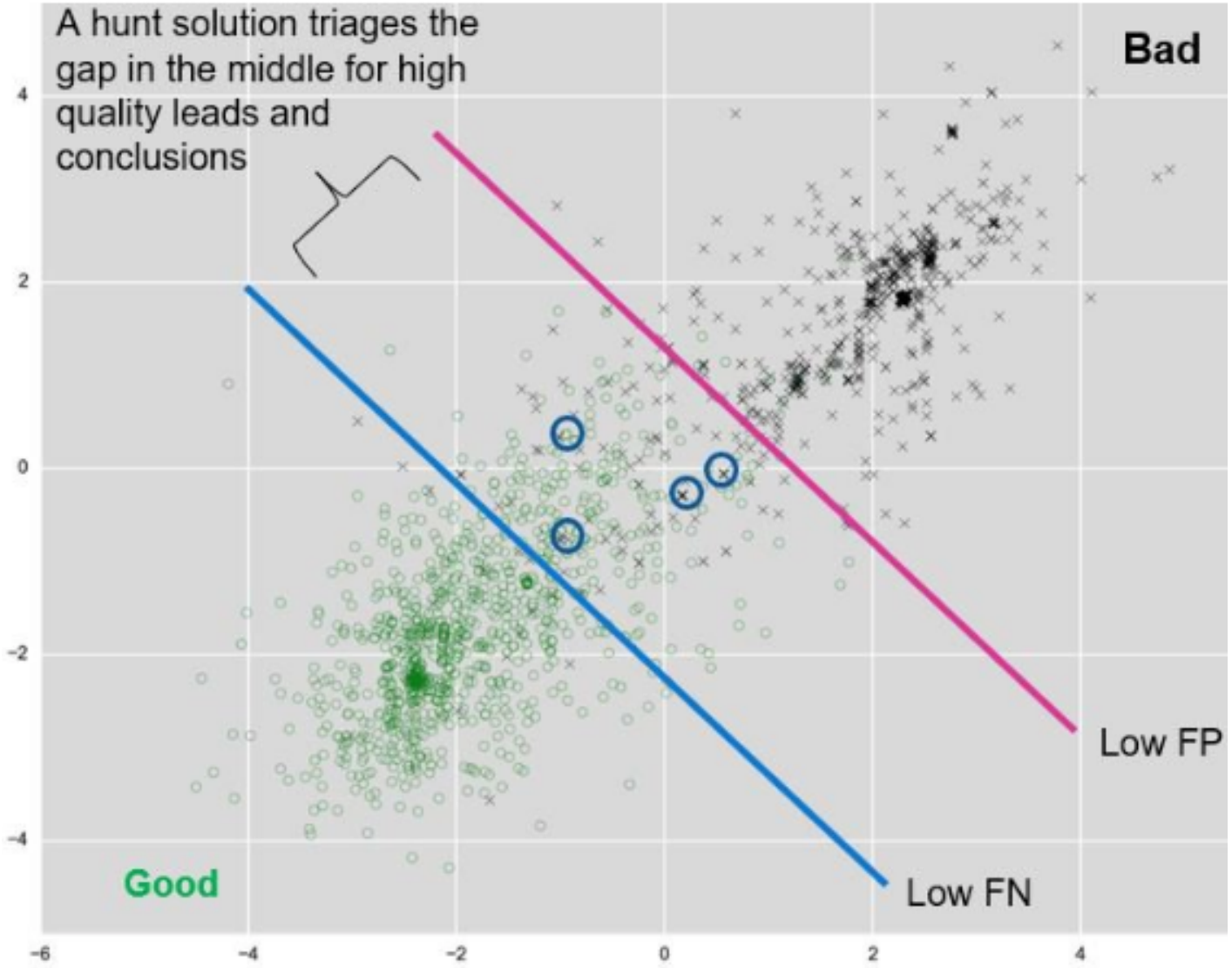
```
# Import the packages we'll need

# Some basic python and jupyter stuff
from collections import defaultdict
import json
from IPython.display import FileLink, FileLinks

# Visualization and data libraries
import altair as alt
import pandas as pd

# ATT&CK Python Client, by @HuntersForge (https://github.com/hunters-forge/ATTACK-Python-Client)
from attackcti import attack_client

# Because this is in Jupyter notebooks we need to enable that renderer for the altair charts to work
alt.renderers.enable('notebook')
```




# Visibility vs Detection

**Most products focused on detection are not good enough for hunting. Why? It's a matter of focus:**

**- Detection (and prevention)-- Few FPs**
**- Visibility and Hunting – Few FNs**

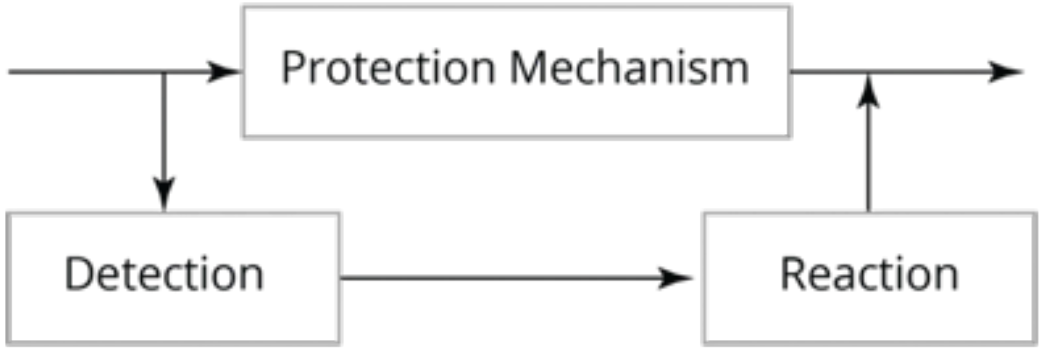
https://deltarisk.com/blog/why-most-real-time-defensive-solutions-are-poor-threat-hunting-solutions/

# Time Based Security: Protection only buys you "time"

If the attacker steals the safe and brings it home, he eventually wins. It's a matter of time.

P < D + R

## Time-based Security




*Schwartau, "If it **takes longer to detect and to respond** to an intrusion than the amount of protection time afforded by the security measures, that is if **P < D + R**, then **effective security is impossible to achieve in this system**."*

P > D + R

While prevention works, detecting and responding early help to mitigate impact.

# It's All About Time: Racing With the Adversary

- Attacker is breaking through the front door
- Contained compromise
- Scoped, investigated, mitigated
- Cheap $$$$
- Effective security
- Reusable Intel

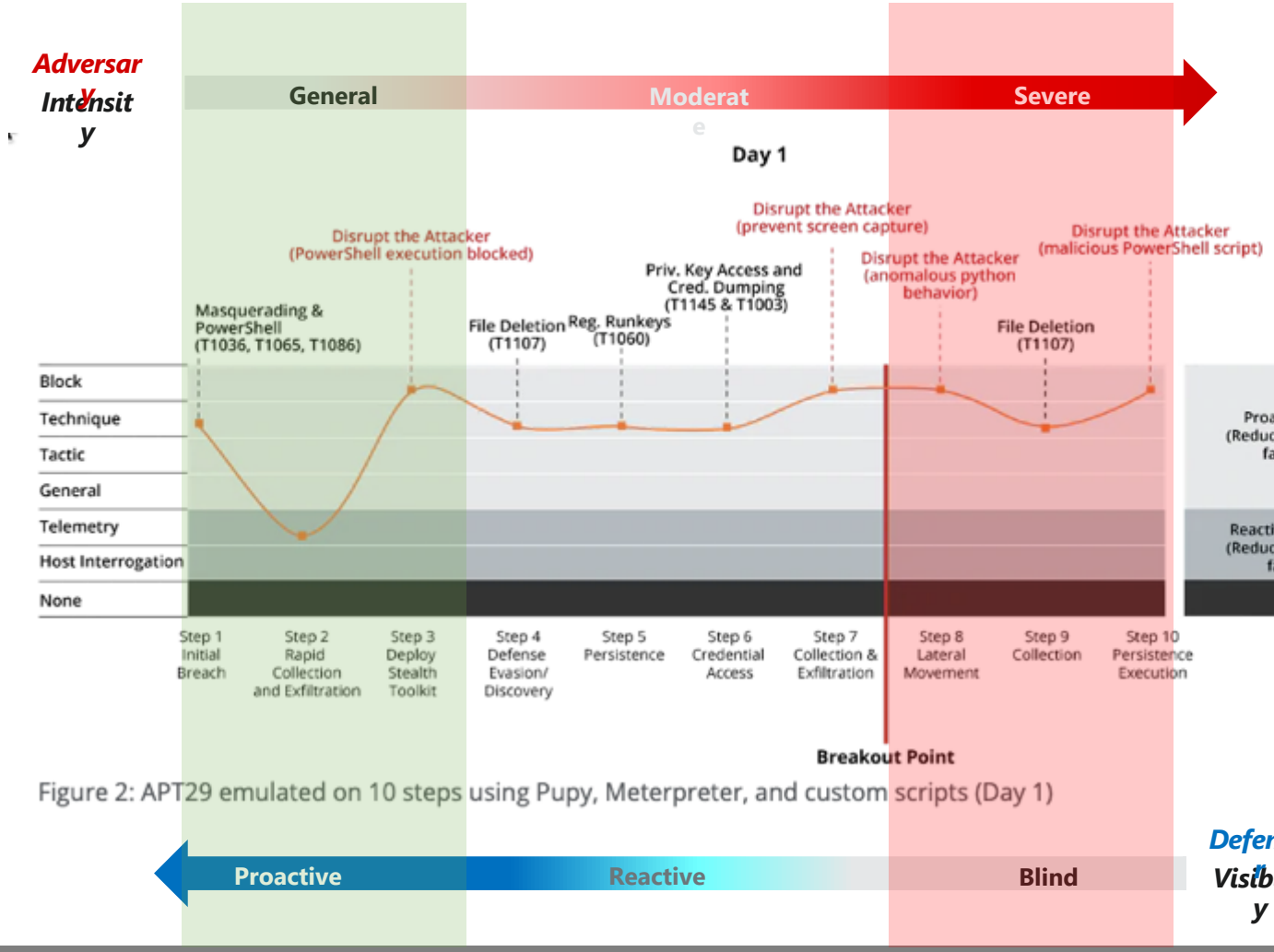
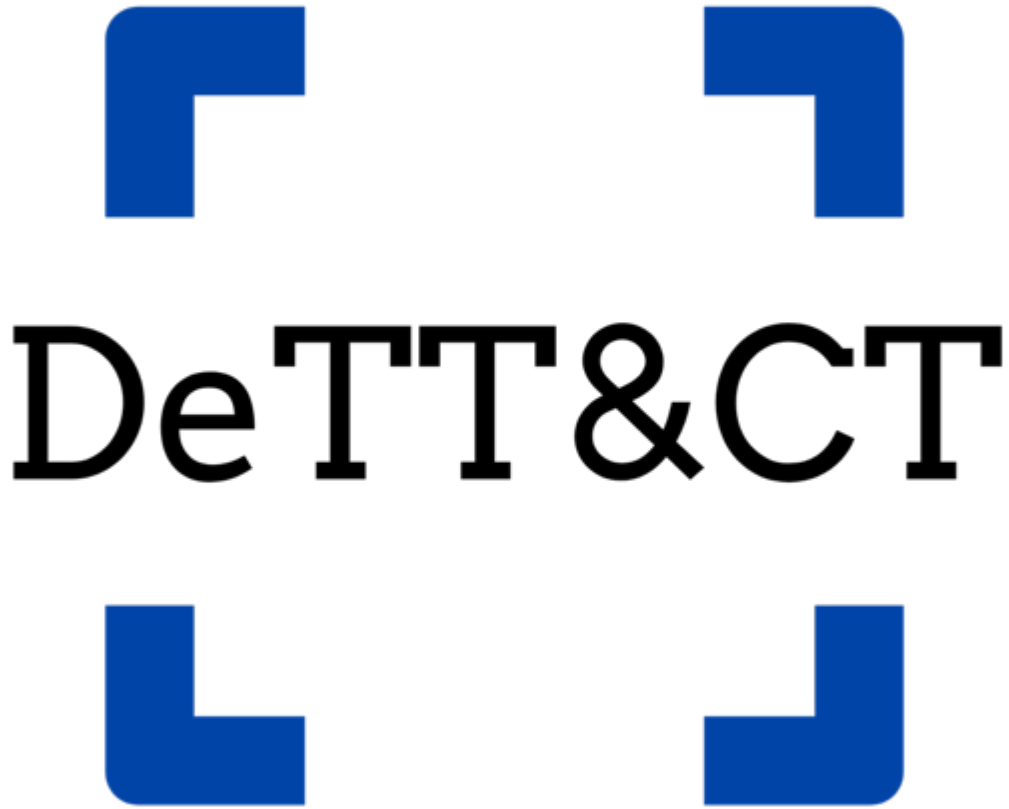



Figure 2: APT29 emulated on 10 steps using Pupy, Meterpreter, and custom scripts (Day 1)

- Attacker left the building
- High-Impact Incident
- Potential Breach $$$$
- Lack of visibility, detection, and effective response
- Reduced value Intel




Framework to manage, assess and compare the quality of:

- Data sources
- Visibility
- Detection
- Threat actors

https://github.com/rabobank-cdc/DeTTECT

# Data Sources

https://github.com/rabobank-cdc/DeTTECT/wiki/Data-sources

Almost 60 different sources mapped to different platforms

Home > Techniques > Enterprise > Standard Application Layer Protocol

## Standard Application Layer Protocol

Adversaries may communicate using a common, standardized application layer protocol such as HTTP, HTTPS, SMTP, or DNS to avoid detection by blending in with existing traffic. Commands to the remote system, and often the results of those commands, will be embedded within the protocol traffic between the client and server.

For connections that occur internally within an enclave (such as those between a proxy or pivot node and other nodes), commonly used protocols are RPC, SSH, or RDP.

https://attack.mitre.org/techniques/T1071/

ID: T1071

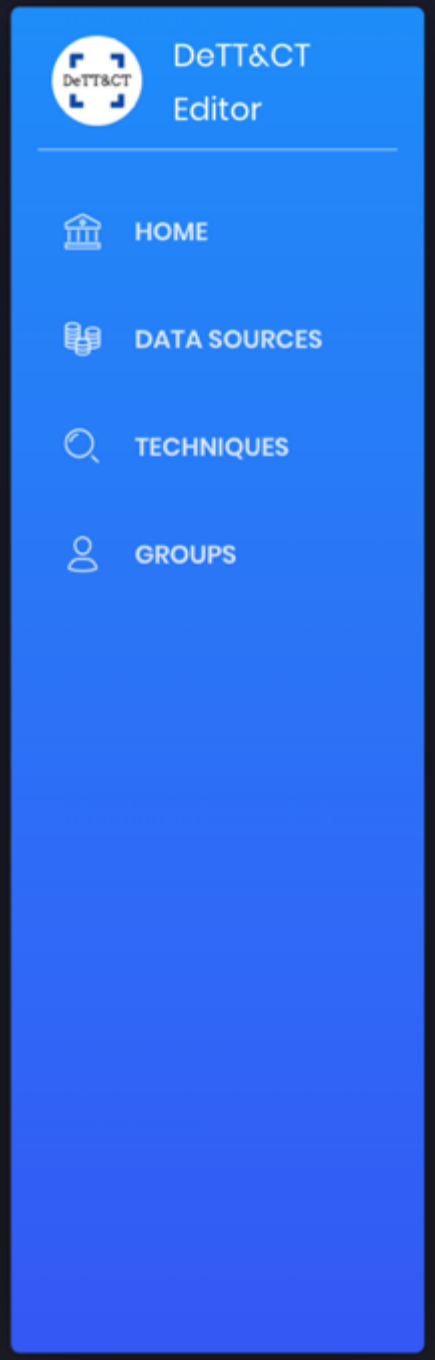
Tactic: Command And Control

Platform: Linux, macOS, Windows

Data Sources: Packet capture, Netflow/Enclave netflow, Process use of network, Malware reverse engineering, Process monitoring

Requires Network: Yes

Version: 1.0




# Data Sources

New file | Select YAML file

**File details**

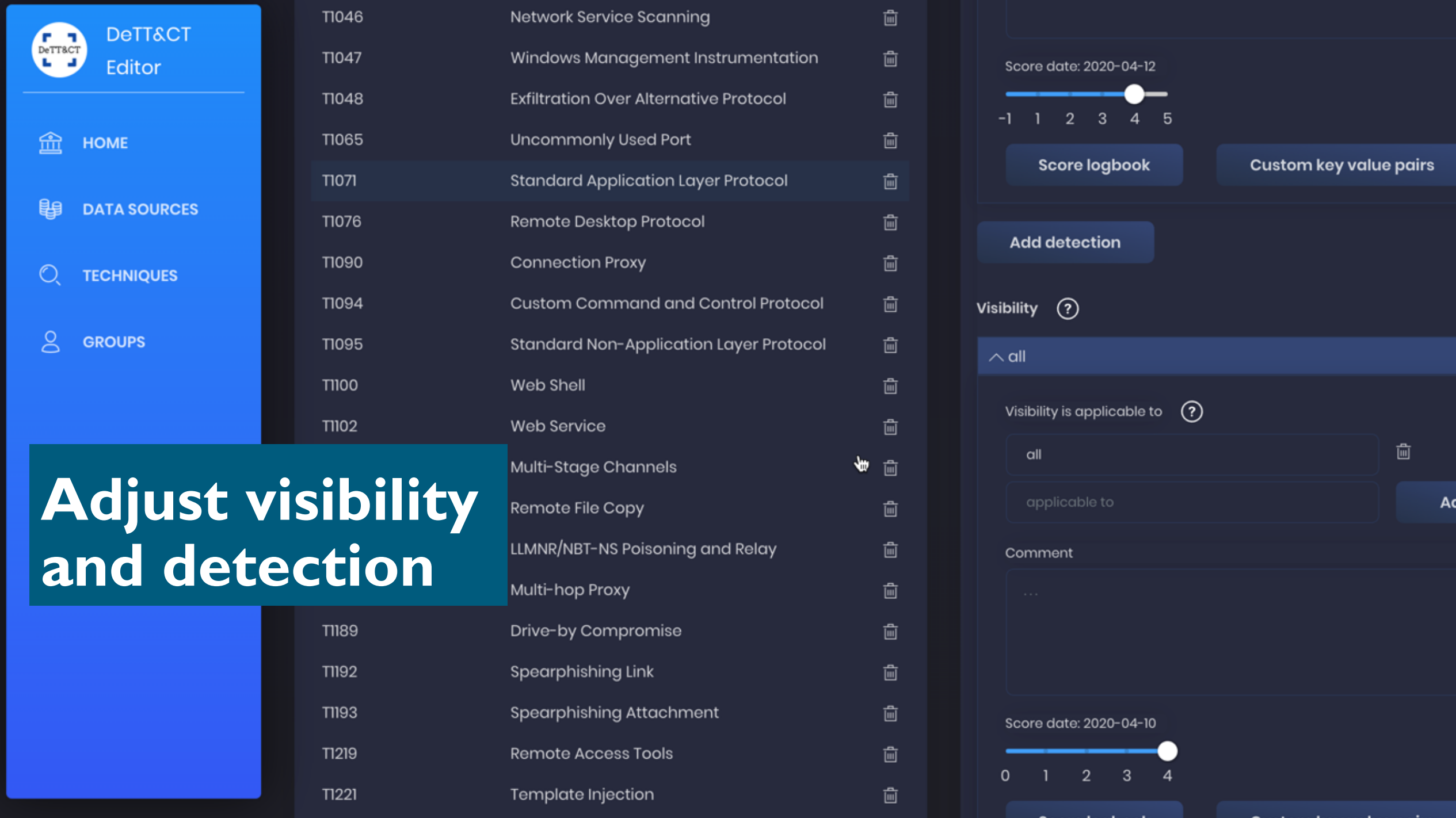
Filename: data-sources-new.yaml
File type: data-source-administration
Version: 1.0
Name: example
Platform: all, Windows, Linux, macOS, AWS, GCP, Azure, Azure AD, Office 365, SaaS

Save YAML file

Add data source

filter

| Name | Date registered | Products |
|---|---|---|
| DNS records | 2020-04-11 | Zeek sensor, DNS Logs from Windows Server |
| Netflow/Enclave netflow | 2020-04-11 | Zeek sensor |
| Network intrusion detection system | 2020-04-11 | Zeek sensor |

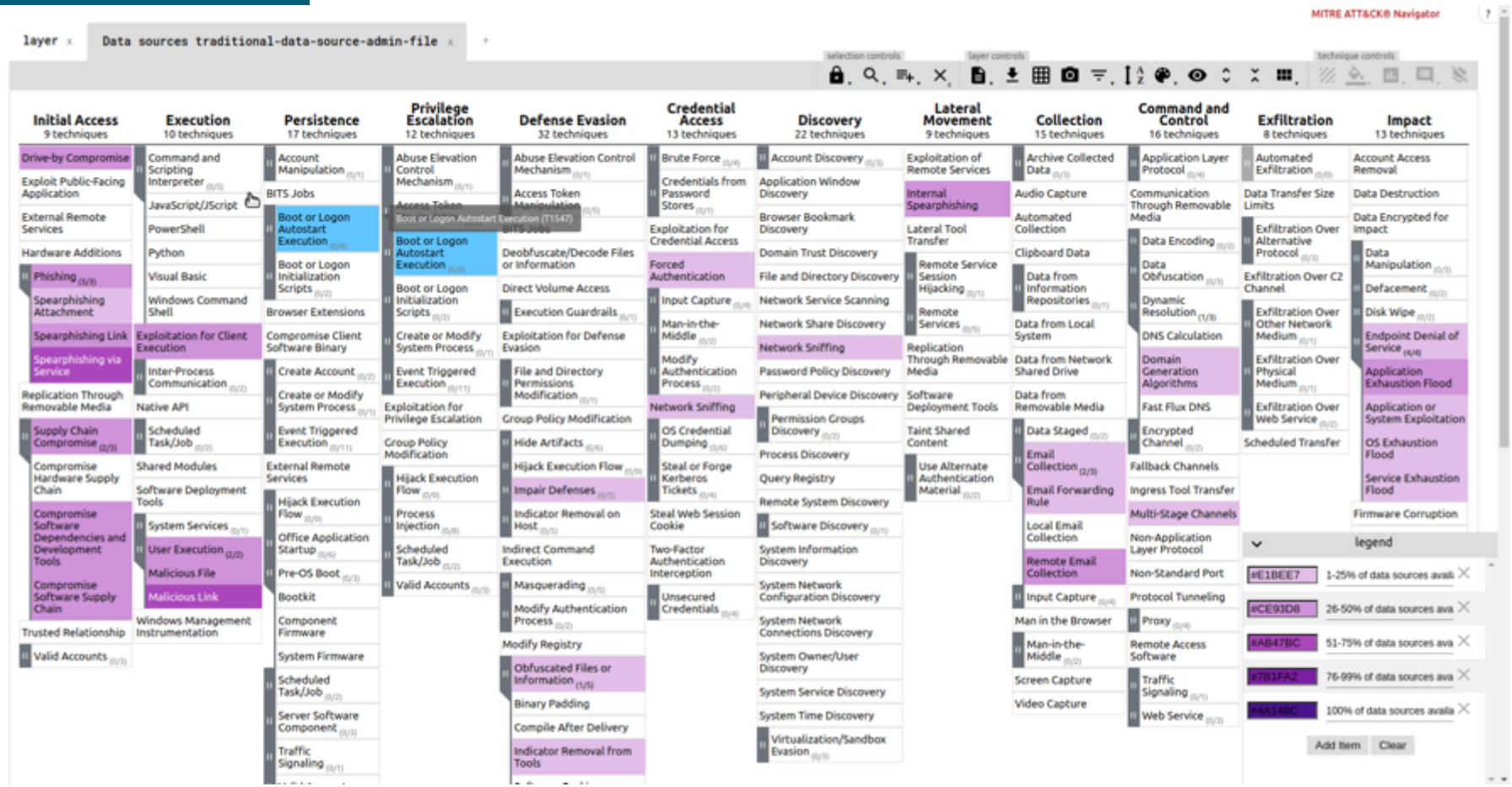

DeTT&CT Editor
HOME
DATA SOURCES
TECHNIQUES
GROUPS
T1046 Network Service Scanning
T1047 Windows Management Instrumentation
T1048 Exfiltration Over Alternative Protocol
T1065 Uncommonly Used Port
T1071 Standard Application Layer Protocol
T1076 Remote Desktop Protocol
T1090 Connection Proxy
T1094 Custom Command and Control Protocol
T1095 Standard Non-Application Layer Protocol
T1100 Web Shell
T1102 Web Service
Multi-Stage Channels
Remote File Copy
LLMNR/NBT-NS Poisoning and Relay
Multi-hop Proxy
T1189 Drive-by Compromise
T1192 Spearphishing Link
T1193 Spearphishing Attachment
T1219 Remote Access Tools
T1221 Template Injection
Score date: 2020-04-12
-1 1 2 3 4 5
Score logbook
Custom key value pairs
Add detection
Visibility
all
Visibility is applicable to
all
applicable to
Comment
Score date: 2020-04-10
0 1 2 3 4
Adjust visibility and detection

# Visibility

https://github.com/rabobank-cdc/DeTTECT/wiki/Visibility-coverage

Generates a JSON file with the coverage of visibility based on the techniques administration template: **python dettect.py v -ft sample-data/techniques-administration-endpoints.yaml -fd input/data-sources-sec530.yaml -l**

Legend indicates the type of visibility based on the input given



# Detection

https://github.com/rabobank-cdc/DeTTECT/wiki/Detection-coverage

Generates a JSON file with the coverage of detection based on the techniques administration template: : **python dettect.py d -ft sample-data/techniques-administration-endpoints.yaml -l**

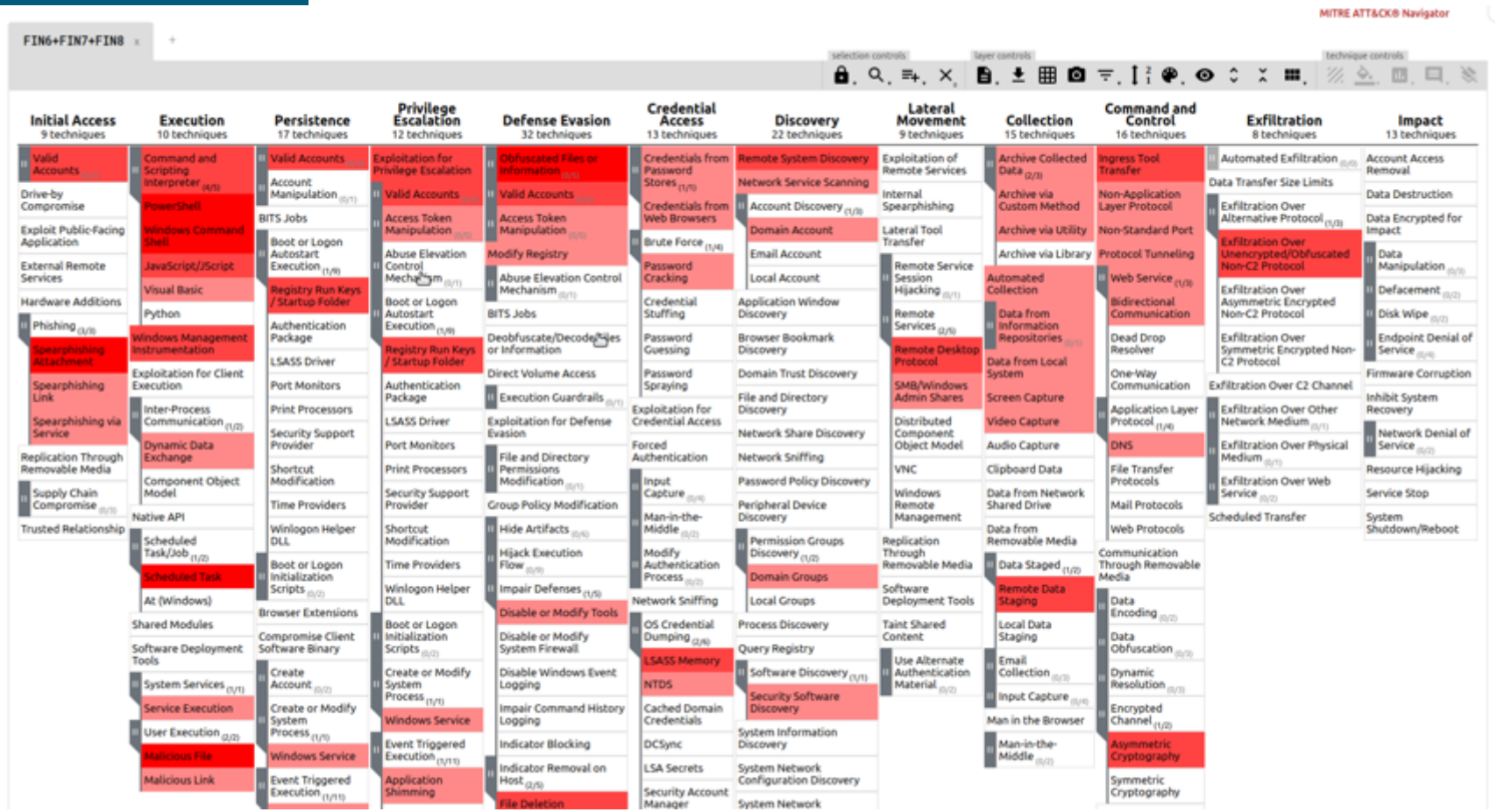
Legend indicates the quality of detection based on the input given

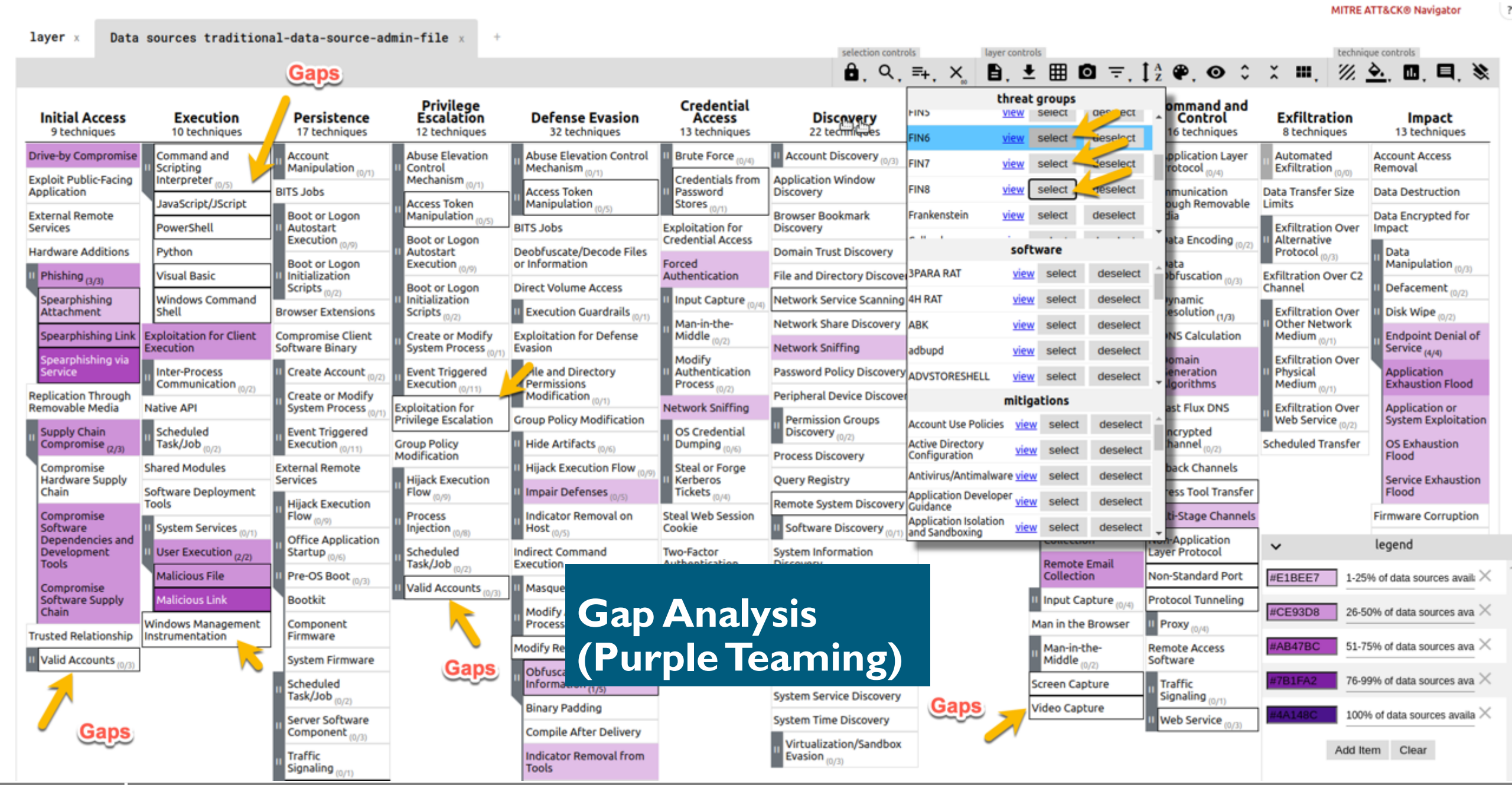
# Threat Actors

https://github.com/rabobank-cdc/DeTTECT/wiki/Threat-actor-group-mapping

Generates a JSON file with a heatmap of the groups we're interested in (FIN6, FIN7, FIN8): **python dettect.py g -g 'fin7,fin8,fin6'**

Results are sorted by most common techniques (darker red)




MITRE ATT&CK® Navigator
layer
Data sources traditional-data-source-admin-file
Gaps
Initial Access
9 techniques
Drive-by Compromise
Exploit Public-Facing Application
External Remote Services
Hardware Additions
Phishing (3/3)
Spearphishing Attachment
Spearphishing Link
Spearphishing via Service
Replication Through Removable Media
Supply Chain Compromise (2/3)
Compromise Hardware Supply Chain
Compromise Software Dependencies and Development Tools
Compromise Software Supply Chain
Trusted Relationship
Valid Accounts (0/3)
Execution
10 techniques
Command and Scripting Interpreter (0/5)
JavaScript/JScript
PowerShell
Python
Visual Basic
Windows Command Shell
Exploitation for Client Execution
Inter-Process Communication (0/2)
Native API
Scheduled Task/Job (0/2)
Shared Modules
Software Deployment Tools
System Services (0/1)
User Execution (2/2)
Malicious File
Malicious Link
Windows Management Instrumentation
Persistence
17 techniques
Account Manipulation (0/1)
BITS Jobs
Boot or Logon Autostart Execution (0/9)
Boot or Logon Initialization Scripts (0/2)
Browser Extensions
Compromise Client Software Binary
Create Account (0/2)
Create or Modify System Process (0/1)
Event Triggered Execution (0/11)
External Remote Services
Hijack Execution Flow (0/9)
Office Application Startup (0/6)
Pre-OS Boot (0/3)
Bootkit
Component Firmware
System Firmware
Scheduled Task/Job (0/2)
Server Software Component (0/3)
Traffic Signaling (0/1)
Privilege Escalation
12 techniques
Abuse Elevation Control Mechanism (0/1)
Access Token Manipulation (0/5)
Boot or Logon Autostart Execution (0/9)
Boot or Logon Initialization Scripts (0/2)
Create or Modify System Process (0/1)
Event Triggered Execution (0/11)
Exploitation for Privilege Escalation
Group Policy Modification
Hijack Execution Flow (0/9)
Process Injection (0/8)
Scheduled Task/Job (0/2)
Valid Accounts (0/3)
Gaps
Defense Evasion
32 techniques
Abuse Elevation Control Mechanism (0/1)
Access Token Manipulation (0/5)
BITS Jobs
Deobfuscate/Decode Files or Information
Direct Volume Access
Execution Guardrails (0/1)
Exploitation for Defense Evasion
File and Directory Permissions Modification (0/1)
Group Policy Modification
Hide Artifacts (0/6)
Hijack Execution Flow (0/9)
Impair Defenses (0/5)
Indirect Command Execution
Masque
Modify
Process
Modify Re
Obfusca
Information (1/5)
Binary Padding
Compile After Delivery
Indicator Removal from Tools
Credential Access
13 techniques
Brute Force (0/4)
Credentials from Password Stores (0/1)
Exploitation for Credential Access
Forced Authentication
Input Capture (0/4)
Man-in-the-Middle (0/2)
Modify Authentication Process (0/2)
Network Sniffing
OS Credential Dumping (0/6)
Steal or Forge Kerberos Tickets (0/4)
Steal Web Session Cookie
Two-Factor
Discovery
22 techniques
Account Discovery (0/3)
Application Window Discovery
Browser Bookmark Discovery
Domain Trust Discovery
File and Directory Discover
Network Service Scanning
Network Share Discovery
Network Sniffing
Password Policy Discovery
Peripheral Device Discover
Permission Groups Discovery (0/2)
Process Discovery
Query Registry
Remote System Discovery
Software Discovery (0/1)
System Information
System Service Discovery
System Time Discovery
Virtualization/Sandbox Evasion (0/3)
Gaps
Remote Email Collection
Input Capture (0/4)
Man in the Browser
Man-in-the-Middle (0/2)
Screen Capture
Video Capture
Command and Control
16 techniques
Application Layer Protocol (0/4)
Communication Through Removable Media
Data Encoding (0/2)
Data Obfuscation (0/3)
Dynamic Resolution (1/3)
DNS Calculation
Domain Generation Algorithms
Fast Flux DNS
Encrypted Channel (0/2)
Fallback Channels
Ingress Tool Transfer
Multi-Stage Channels
Non-Application Layer Protocol
Non-Standard Port
Protocol Tunneling
Proxy (0/4)
Remote Access Software
Traffic Signaling (0/1)
Web Service (0/3)
Exfiltration
8 techniques
Automated Exfiltration (0/0)
Data Transfer Size Limits
Exfiltration Over Alternative Protocol (0/3)
Exfiltration Over C2 Channel
Exfiltration Over Other Network Medium (0/1)
Exfiltration Over Physical Medium (0/1)
Exfiltration Over Web Service (0/2)
Scheduled Transfer
Impact
13 techniques
Account Access Removal
Data Destruction
Data Encrypted for Impact
Data Manipulation (0/3)
Defacement (0/2)
Disk Wipe (0/2)
Endpoint Denial of Service (4/4)
Application Exhaustion Flood
Application or System Exploitation
OS Exhaustion Flood
Service Exhaustion Flood
Firmware Corruption
threat groups
FIN6 view select deselect
FIN7 view select deselect
FIN8 view select deselect
Frankenstein view select deselect
software
3PARA RAT view select deselect
4H RAT view select deselect
ABK view select deselect
adbupd view select deselect
ADVSTORESHELL view select deselect
mitigations
Account Use Policies view select deselect
Active Directory Configuration view select deselect
Antivirus/Antimalware view select deselect
Application Developer Guidance view select deselect
Application Isolation and Sandboxing view select deselect
legend
#E1BEE7 1-25% of data sources avail
#CE93D8 26-50% of data sources ava
#AB47BC 51-75% of data sources ava
#7B1FA2 76-99% of data sources ava
#4A148C 100% of data sources availa
Add Item
Clear
Gap Analysis (Purple Teaming)

# Additional Resources

Thinking Red, Acting Blue - https://www.sans.org/webcasts/defensible-security-architecture-engineering-2-thinking-red-acting-blue-mindset-actions-109710

The Githubification of InfoSec by John Lambert - https://medium.com/@johnlatwc/the-githubification-of-infosec-afbdbfaad1d1

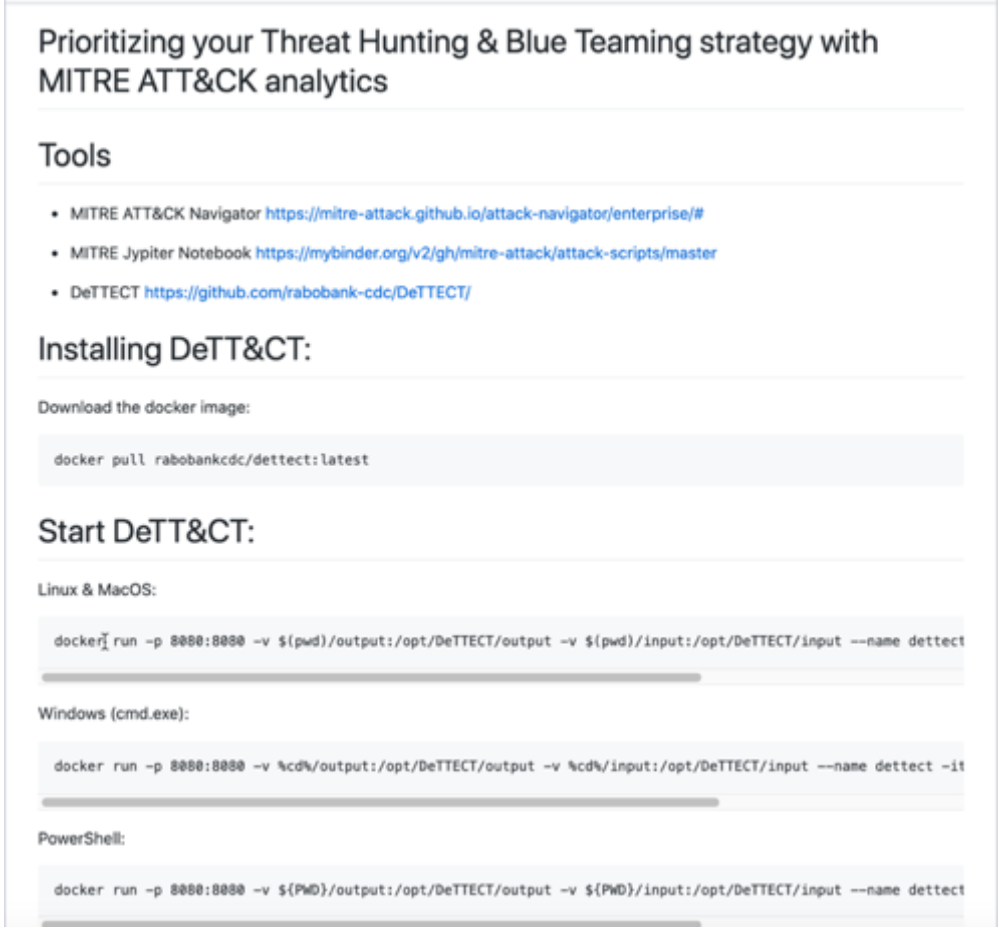
DETT&CT - https://github.com/rabobank-cdc/DeTTECT

MITRE's Red Teaming dataset - https://github.com/mitre/brawl-public-game-001

Playbook for this talk (step by step commands): https://github.com/aboutsecurity/Talks-and-Presentations/blob/master/c0r0n4con.md

MITRE Assistant by dfirence (Carlos Diaz)

https://github.com/dfirence/mitre-assistant

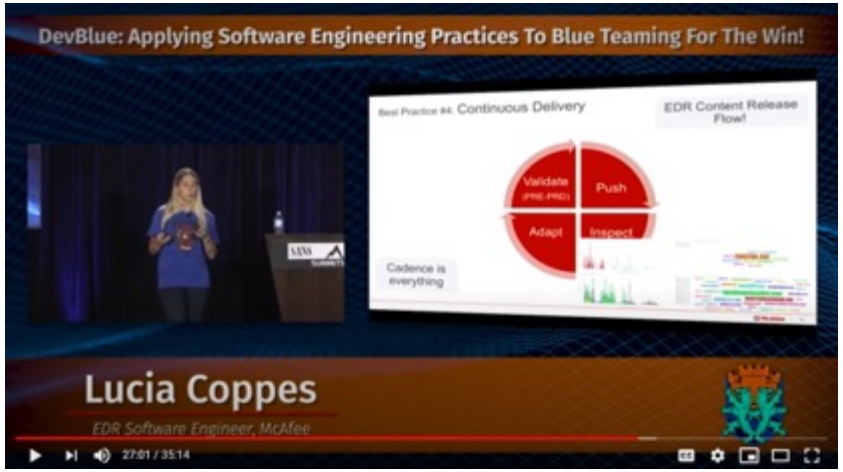


# Purple Teaming & SOC Engineering

*https://medium.com/anton-on-security/can-we-have-detection-as-code-96f869cfdc79

*Anton Chuvakin on Detection Engineering (Set 2020)**

- *100% Detection content versioning*
- *Proper "QA" for detection content*
- *Content (code) reuse and modularity of detection content*
- *Metrics and improvement are also key*
- *Full CI/CD pipeline for detections to continuously build, refine, deploy and run detection logic*





https://www.youtube.com/watch?v=H3bsVOlGZfg

# Cross-org Training, Purple Teaming & Risk Mitigation

- SOC rotations
- Customer calls
- SANS courses
- Research
- Defend The Flag!



# Presentation based on SEC530: Defensible Security Architecture and Engineering

## Thank you!! Follow @SecurityMapper & @aboutsecurity for updates and new webinars!




## References

https://www.sans.org/course/defensible-security-architecture-and-engineering