Quantify Your Hunt:

Not Your Parent's Red Teaming Redux

Agenda for this talk

- Who we are
- Why we're giving this talk with these updates
- Understanding effectiveness
- Things you might be already measuring
- Enterprise ATT&CK, the practitioner's choice of knowledge bases
- Measuring, data quality and ATT&CK
- Statistics, Data science and how they help

Introductions

Honestly, we're not as interesting as this topic

Devon Kerr @_devonkerr_

Principal Threat Researcher for R&D @endgameinc, I do these things:

- Detection and response capability development
- Endgame SOC lead
- RTA contributor and simulation advocate
- Hunt desk reference co-author

Former Incident Response lead at Mandiant with more than 6 years of IR, forensics, analyst training and security program development.

Knowledge areas include Incident management and response, living off the land using native features, adversary tradecraft, simulation and threat detection.

Roberto Rodriguez @Cyb3rWard0g

- Adversary Detection Analyst @**SpecterOps**
- https://github.com/Cyb3rWard0g
  - ThreatHunter-Playbook
  - Hunting ELK (HELK)
  - ATTACK-Python-Client
  - Open Source Security Event Metadata (OSSEM)
- Former:
  - Capital One, Senior Threat Hunter

Why this talk?

Why again?

Hunt teams still struggle with …

- Aligning hunting campaigns & business priorities
- Providing transparency to senior leadership
- Showing progress over time
- Mapping gaps to data sources and security controls
  - Coverage of adversary techniques is much more than a green check or red “x”
- Assessing the effectiveness of the program and any tools used during engagements
- Developing and implementing parity with ATT&CK
- Just getting started, candidly

Effective Hunting

How are you effective? What does being effective even mean?

____ for effective threat hunting

Efficiency Efficacy

Effectiveness

Efficiency

The way resources are used (or wasted),
How much I make the most of the resources I have

Effectiveness

Accomplishes the goals (to be efficacious) employing the best and most economic methodology (to be efficient).

Efficacy

It doesn't matter how we do it, but only on what we accomplish

Efficiency

- Choosing an adversary model
- Assessing quality of data
  - Do we even have the data?
- Utilizing the right technology
- Applying the right personnel skills
- Prioritizing adversary techniques
- Enhancing data security analytics

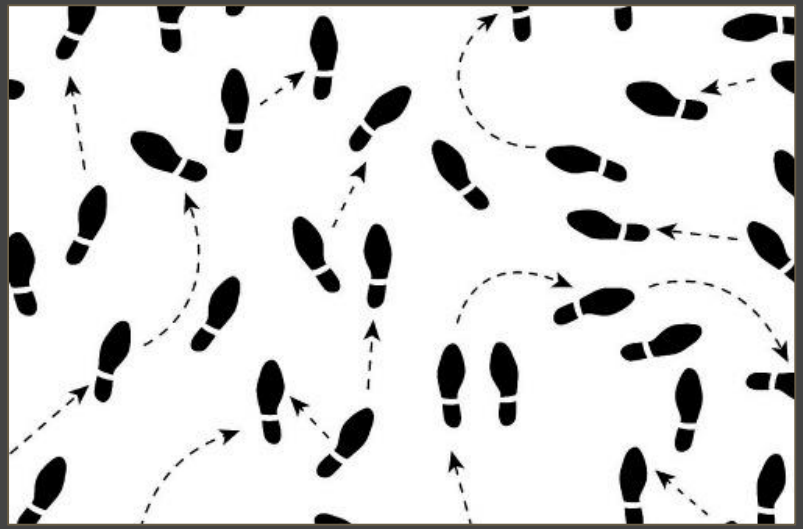
Efficacy

- Let's find evil! Can we detect it? Yes or No?
  - Signatures vs security analytics
  - Are you considering attack variations?
- Uncovering Incidents vs Validating Detection of adversaries

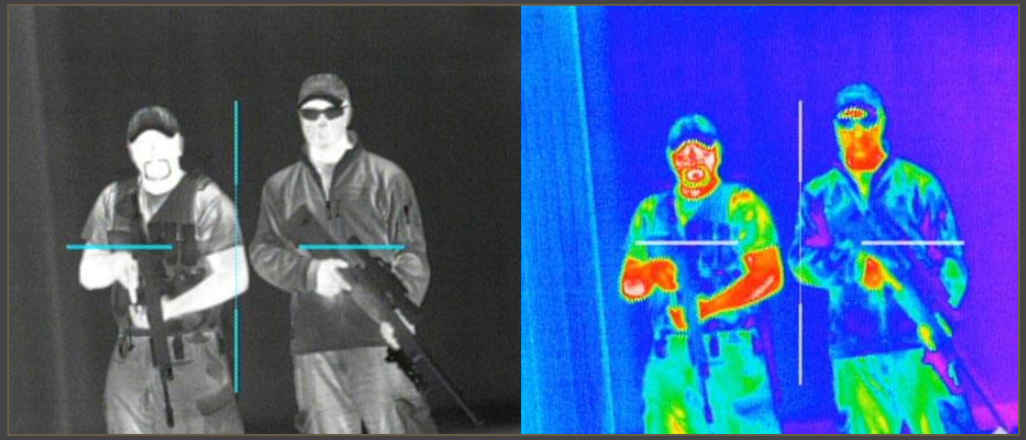
Where do I start?

How are we going to start approaching this?

Two steps back, one step forward: or history repeats itself



The Evolution of the Hunt HeatMap



How Hot Is Your Hunt Team?

https://cyberwardog.blogspot.com/2017/07/how-hot-is-your-hunt-team.html

Ready to hunt? First, Show me your data!

https://cyberwardog.blogspot.com/2017/12/ready-to-hunt-first-show-me-your-data.html

What are you potentially measuring already?

Risk Forecasting

- Choose a risk to measure
- Decompose the scenario
- Gather supporting data
- Make forecasts
- Mitigate the potential risk
- Measure again

Scenario: *"An attacker can access destructive AWS IAM permissions in the next 365 days."*

2016 - Q3: 25%

First forecast. We haven't fixed anything yet.

2016 - Q4: 23%

We have limited the destructive capability of keys in production.

2017 - Q1: 16%

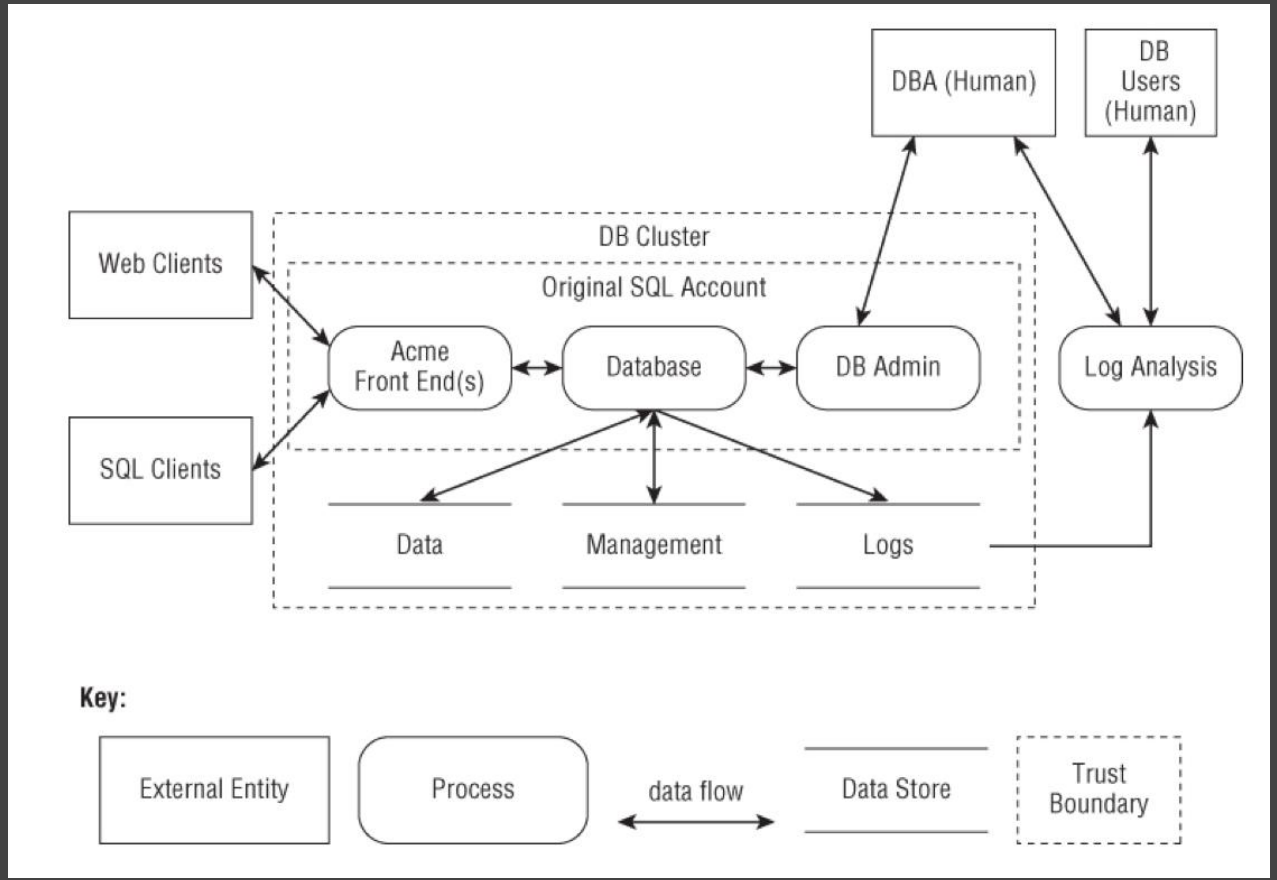
We added multifactor protection to keys used by engineers.

2017 - Q2: 10%

We took keys out of source code and use roles now.

An increase in confidence against this risk of 15%.

https://magoo.github.io/Risk-Forecasting/#173 (Ryan McGeehan @Magoo)



Threat Modeling

- Model the system
- Identify Threats
- Define how threat occurs
- Address threats
- Validate
- Measure again

Shostack, Adam. Threat Modeling: Designing for Security (p. 46). Wiley. Kindle Edition

Where do you fit “hunt”?

Threat Hunting

Identify a technique

Develop a hypothesis

Identify scope and resources

Develop Analytics

Validate & Report

Automate? & Repeat

Risk Forecasting

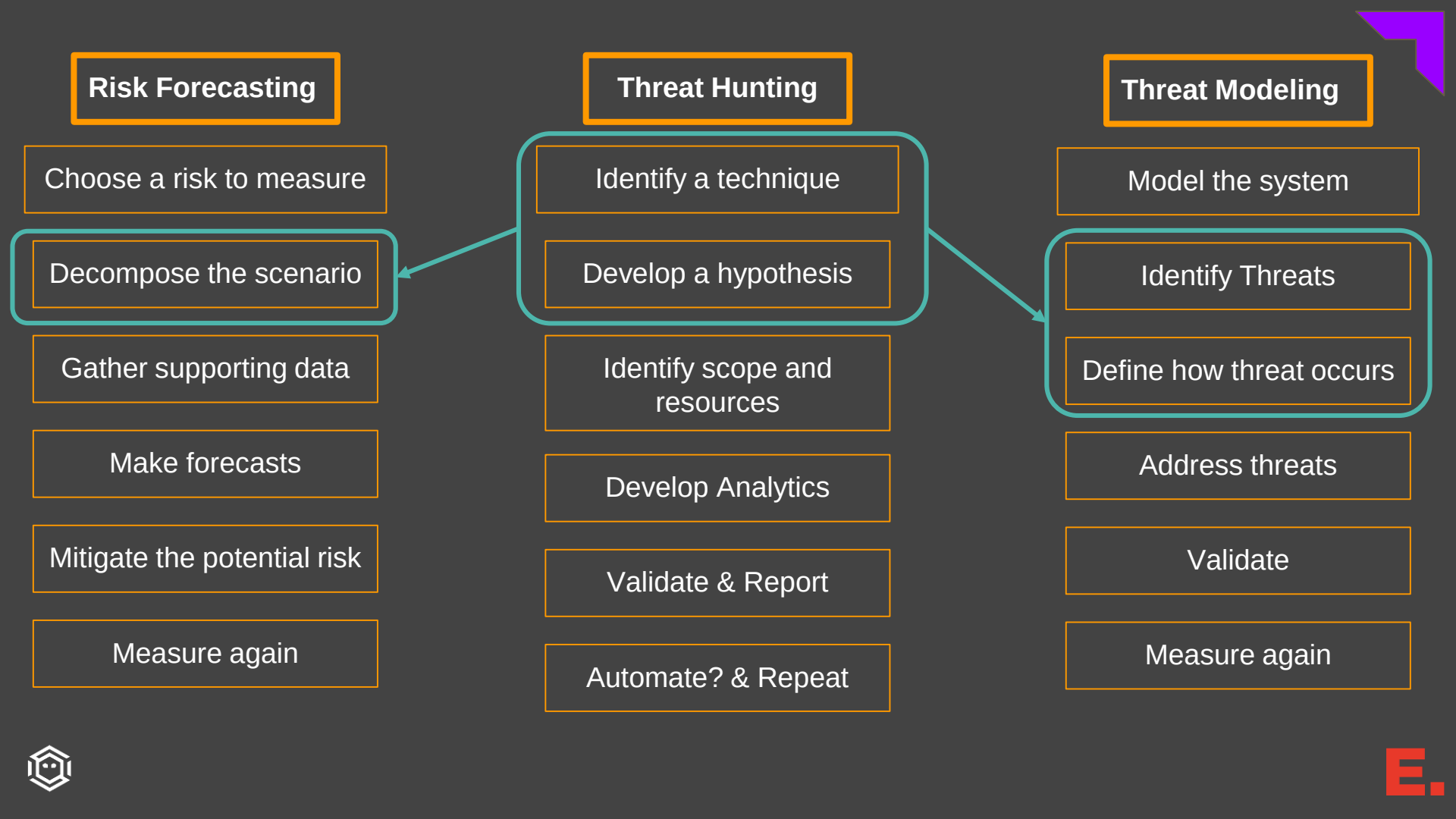
- Choose a risk to measure
- Decompose the scenario
- Gather supporting data
- Make forecasts
- Mitigate the potential risk
- Measure again

Threat Hunting

- Identify a technique
- Develop a hypothesis
- Identify scope and resources
- Develop Analytics
- Validate & Report
- Automate? & Repeat

Threat Modeling

- Model the system
- Identify Threats
- Define how threat occurs
- Address threats
- Validate
- Measure again


Risk Forecasting
Choose a risk to measure
Decompose the scenario
Gather supporting data
Make forecasts
Mitigate the potential risk
Measure again
Threat Hunting
Identify a technique
Develop a hypothesis
Identify scope and resources
Develop Analytics
Validate & Report
Automate? & Repeat
Threat Modeling
Model the system
Identify Threats
Define how threat occurs
Address threats
Validate
Measure again

Risk Forecasting

- Choose a risk to measure
- Decompose the scenario
- Gather supporting data
- Make forecasts
- Mitigate the potential risk
- Measure again

Threat Hunting

- Identify scope and resources
- Develop Analytics
- Validate & Report
- Automate? & Repeat

Threat Modeling

- Model the system
- Identify Threats
- Define how threat occurs
- Address threats
- Validate
- Measure again

What can we measure from a ~~hunt~~ detection?

We need to understand what we are trying to measure from a detection perspective

- Do we have the right resources to validate the detection of identified threats?
  - What percentage of my tools help the most during a hunt?
  - What percentage of data is utilized the most during a hunt?
- How much can we cover with the current resources we have?
  - Percentage of data in relation to detected techniques
  - Percentage of successful analytics for hunt engagements
- Are we reducing the probability of attackers achieving their objective?
  - Percentage reduced each quarter after a hunting engagement. forecasting?

ENTERPRISE ATT&CK

The practitioner's choice of knowledge base

MITRE said it best

“

MITRE’s Adversarial Tactics, Techniques and Common Knowledge (ATT&CK™) is a curated knowledge base and model for cyber adversary behavior, reflecting the various phases of an adversary’s lifecycle and the platforms they are known to target. ATT&CK is useful for understanding security risk against known adversary behavior, for planning security improvements, and verifying defenses work as expected.

”

- MITRE ATT&CK -

We *really* like Enterprise ATT&CK

What's not to like:

- it is threat-agnostic, describing the purpose and effect of many techniques
- contains more than 200 categorized and curated entries
- includes forensic artifacts and references to educate analysts and decrease barrier-to-entry
- techniques are cross-referenced by threat group, *if that's important to your business (it might not be, no judgement)*

ATT&CK STATISTICS (As of April 27, 2018)

- 219 techniques
  - 187 - Windows
  - 130 - MacOS
  - 108 - Linux
- 11 Tactics
- 68 groups
- 187 Tools
- 48 Data Sources
- 39 Contributors
- 21 Bypasses

How can I measure against ATT&CK?

Explore ATT&CK

Access Token Manipulation	
Technique	
ID	T1134
Tactic	Defense Evasion, Privilege Escalation
Platform	Windows
Permissions Required	User, Administrator
Effective Permissions	SYSTEM
Data Sources	API monitoring, Access Tokens

PowerShell	
Technique	
ID	T1086
Tactic	Execution
Platform	Windows
Permissions Required	User, Administrator
Data Sources	Windows Registry, File monitoring, Process command-line parameters, Process monitoring
Supports Remote	Yes

https://www.mitre.org/sites/default/files/publications/pr-18-0944-11-mitre-attack-design-and-philosophy.pdf

Explore ATT&CK

The lowest level of permissions the adversary is required

Access Token Manipulation

Technique

ID	T1134
Tactic	Defense Evasion, Privilege Escalation
Platform	Windows
Permissions Required	User, Administrator
Effective Permissions	SYSTEM
Data Sources	API monitoring, Access Tokens

PowerShell

Technique

ID	T1086
Tactic	Execution
Platform	Windows
Permissions Required	User, Administrator
Data Sources	Windows Registry, File monitoring, Process command-line parameters, Process monitoring
Supports Remote	Yes

https://www.mitre.org/sites/default/files/publications/pr-18-0944-11-mitre-attack-design-and-philosophy.pdf

Explore ATT&CK

The lowest level of permissions the adversary is required

Permissions an adversary will attain by performing the technique

Access Token Manipulation

Technique

ID	T1134
Tactic	Defense Evasion, Privilege Escalation
Platform	Windows
Permissions Required	User, Administrator
Effective Permissions	SYSTEM
Data Sources	API monitoring, Access Tokens

PowerShell

Technique

ID	T1086
Tactic	Execution
Platform	Windows
Permissions Required	User, Administrator
Data Sources	Windows Registry, File monitoring, Process command-line parameters, Process monitoring
Supports Remote	Yes

https://www.mitre.org/sites/default/files/publications/pr-18-0944-11-mitre-attack-design-and-philosophy.pdf

Explore ATT&CK

Access Token Manipulation	
	Technique
ID	T1134
Tactic	Defense Evasion, Privilege Escalation
Platform	Windows
Permissions Required	User, Administrator
Effective Permissions	SYSTEM
Data Sources	API monitoring, Access Tokens

The lowest level of permissions the adversary is required

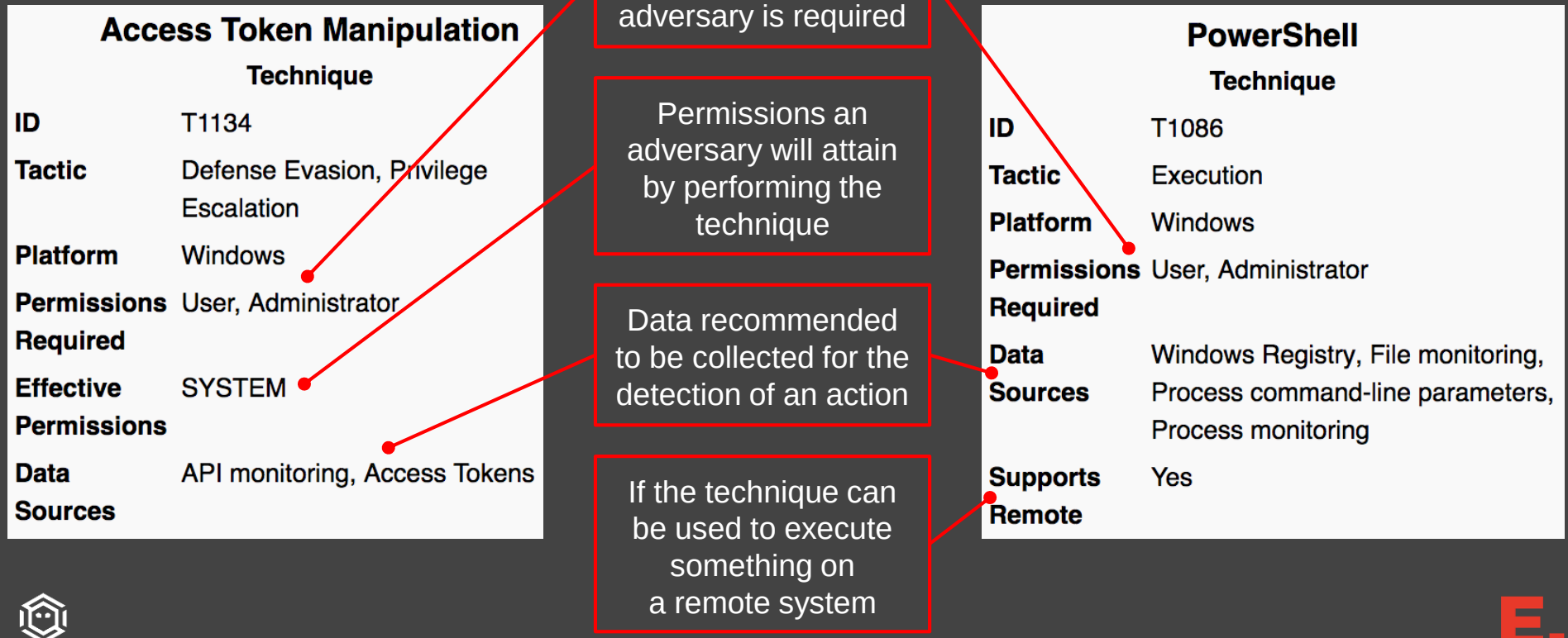
Permissions an adversary will attain by performing the technique

Data recommended to be collected for the detection of an action

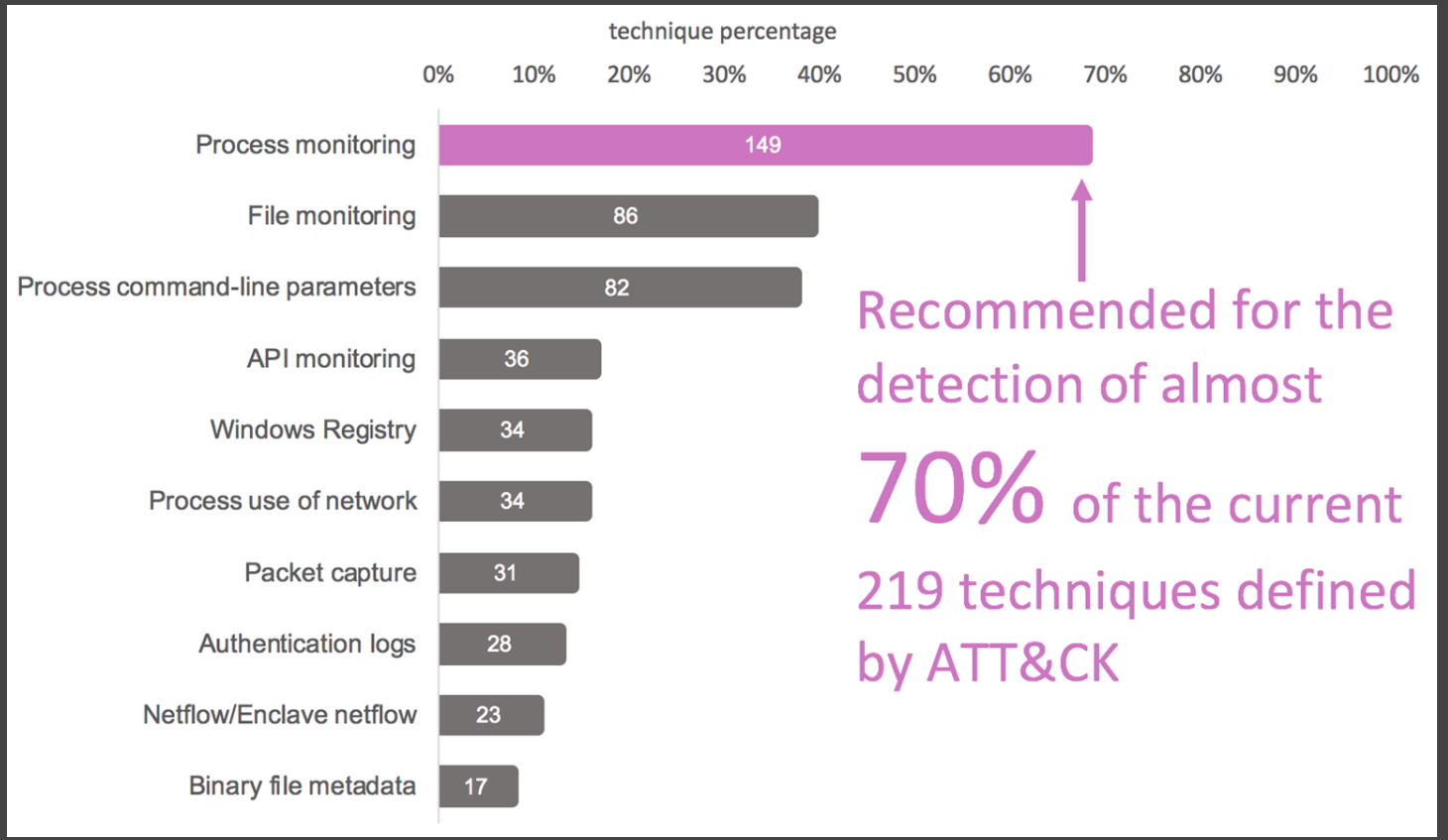
PowerShell	
	Technique
ID	T1086
Tactic	Execution
Platform	Windows
Permissions Required	User, Administrator
Data Sources	Windows Registry, File monitoring, Process command-line parameters, Process monitoring
Supports Remote	Yes

https://www.mitre.org/sites/default/files/publications/pr-18-0944-11-mitre-attack-design-and-philosophy.pdf

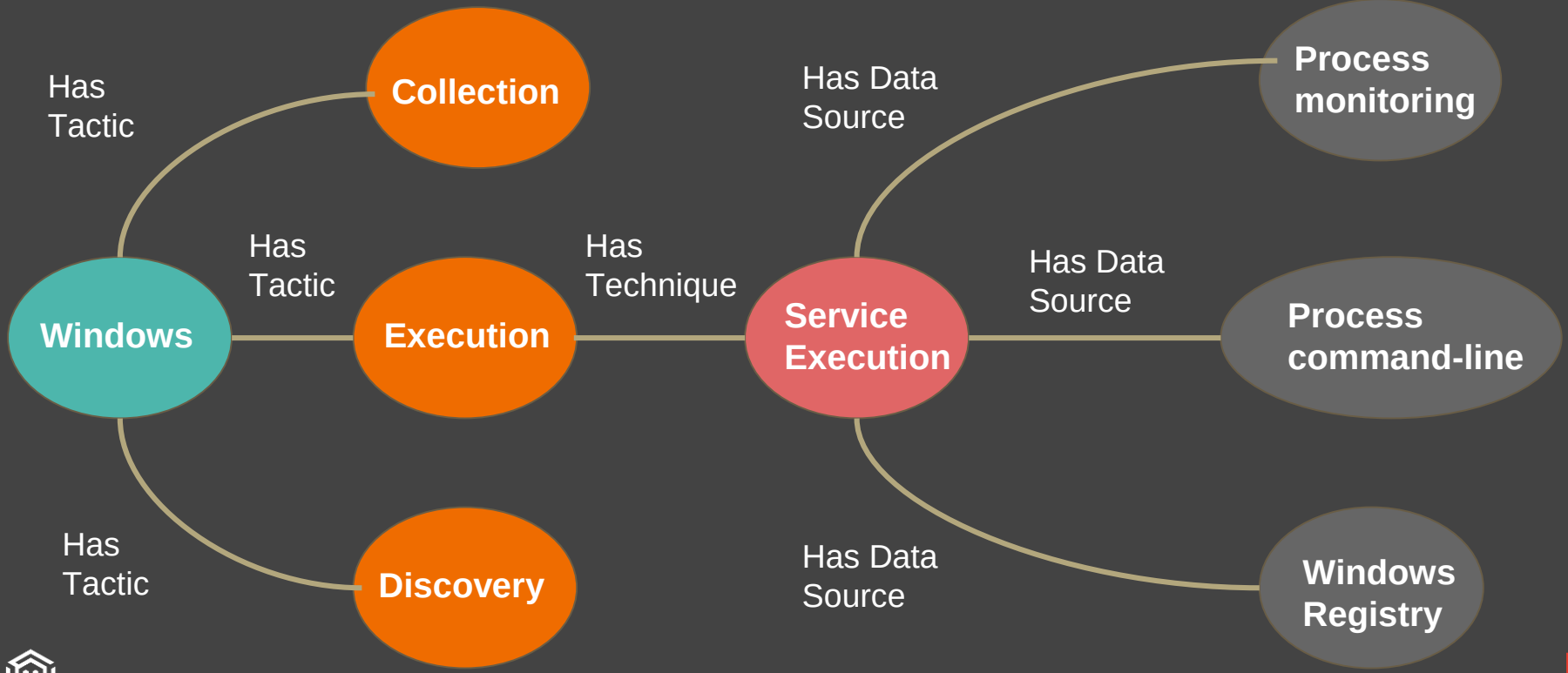
Explore ATT&CK



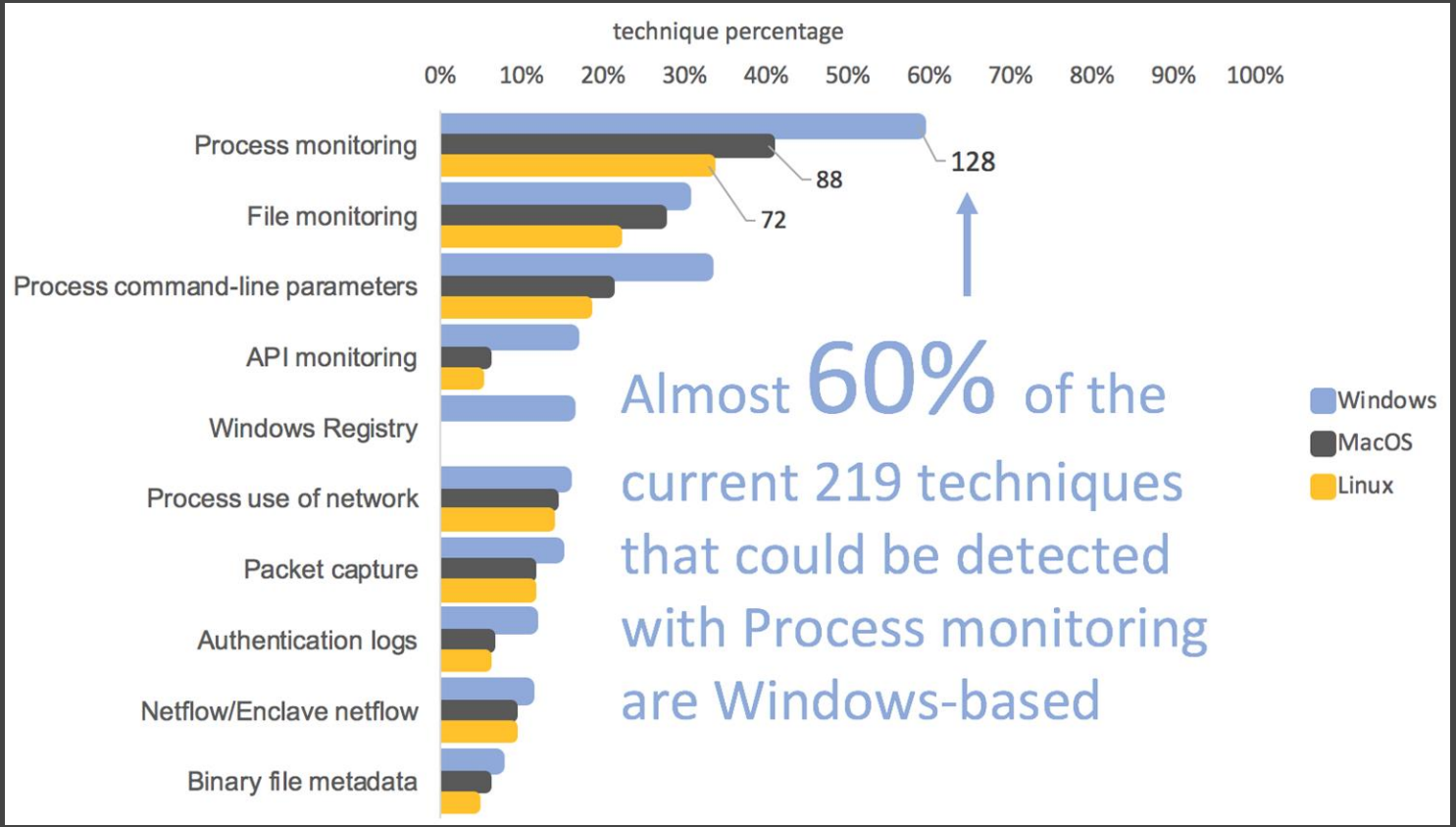
Data Sources -> Adversarial Techniques



Identify Relationships in ATT&CK



Data Sources -> Adversarial Techniques -> OS



Identify Relationships in ATT&CK

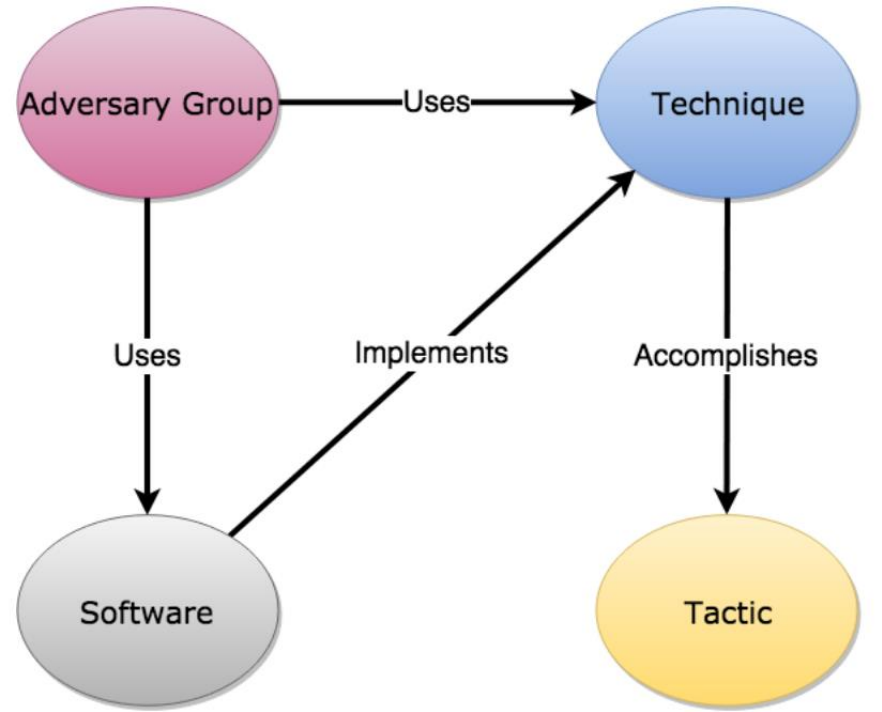
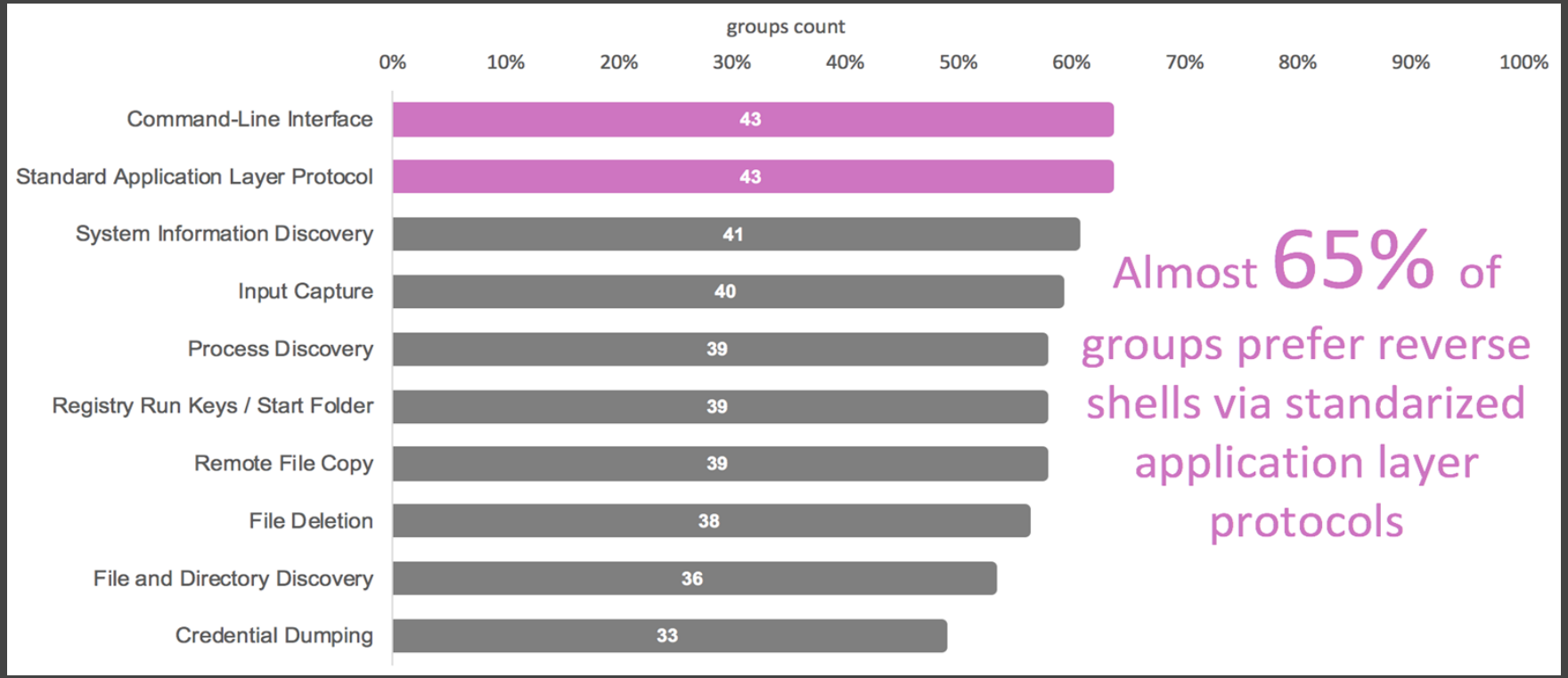


Figure 2. ATT&CK Model Relationships

https://www.mitre.org/sites/default/files/publications/pr-18-0944-11-mitre-attack-design-and-philosophy.pdf

Groups -> Adversarial Techniques



Almost 65% of groups prefer reverse shells via standarized application layer protocols

MITRE has already covered this topic, though

Part 1: Using ATT&CK to Advance Cyber Threat Intelligence

This excellent blogpost by Katie Nickels (@likethecoins) covers:

- An overview of traditional CTI
- Challenges
- How ATT&CK can help provide a way of expressing TTPs, exposing a *common language*
- Using ATT&CK to understand blind spots
- Using TTP counts as a metric to justify your CTI program

Part 2: Using ATT&CK to Advance Cyber Threat Intelligence

The second part in this series focuses on knowledge management and adversary behavior curation, which ATT&CK is perfectly designed to assist with.

Two of the major points to take away:

- Get as close to original information as possible to avoid misinterpreting a tactic or event
- Select *appropriate* information to curate

Adversary Techniques-> Data Sources -> Detection Strategy

What data sources are recommended?

Access Tokens	Detonation chamber	Loaded DLLs
Anti-virus	Digital Certificate Logs	Mail server
API monitoring	DLL monitoring	Malware reverse engineering
Application Logs	DNS records	MBR
Asset Management	EFI	Named Pipes
Authentication logs	Email gateway	Netflow/Enclave netflow
Binary file metadata	Environment variable	Network device logs
BIOS	File monitoring	Network intrusion detection system
Browser extensions	Host network interface	Network protocol analysis
Data loss prevention	Kernel drivers	Packet capture

What data sources are recommended?

PowerShell logs
Process command-line parameters
Process monitoring
Process use of network
Sensor health and status
Services
SSL/TLS inspection
System calls
Third-party application logs
User interface

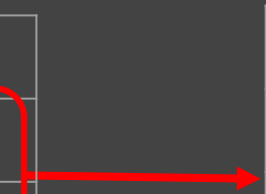
VBR
Web application firewall logs
Web logs
Web proxy
Windows Error Reporting
Windows event logs
Windows Registry
WMI Objects

Let's take a look at data sources again:

PowerShell logs
Process command-line parameters
Process monitoring
Process use of network
Sensor health and status
Services
SSL/TLS inspection
System calls
Third-party application logs
User interface

Process object attributes…

PowerShell logs
Process command-line parameters
Process monitoring
Process use of network
Sensor health and status
Services
SSL/TLS inspection
System calls
Third-party application logs
User interface



Process
process_name
process_command_line
process_path
process_parent_name
user_name
hash_sha256
host_name

What is a data model?

- A data model basically determines the structure of data and the relationships identified among each other.
- MITRE Data Model:
  - Strongly inspired by CybOX, is an organization of the objects that may be monitored from a host-based or network-based perspective.
  - https://car.mitre.org/wiki/Data_Model
- STIX™ Version 2.0. Part 4: Cyber Observable Objects
  - http://docs.oasis-open.org/cti/stix/v2.0/stix-v2.0-part4-cyber-observable-objects.html

Data Model (Defining Data Objects)

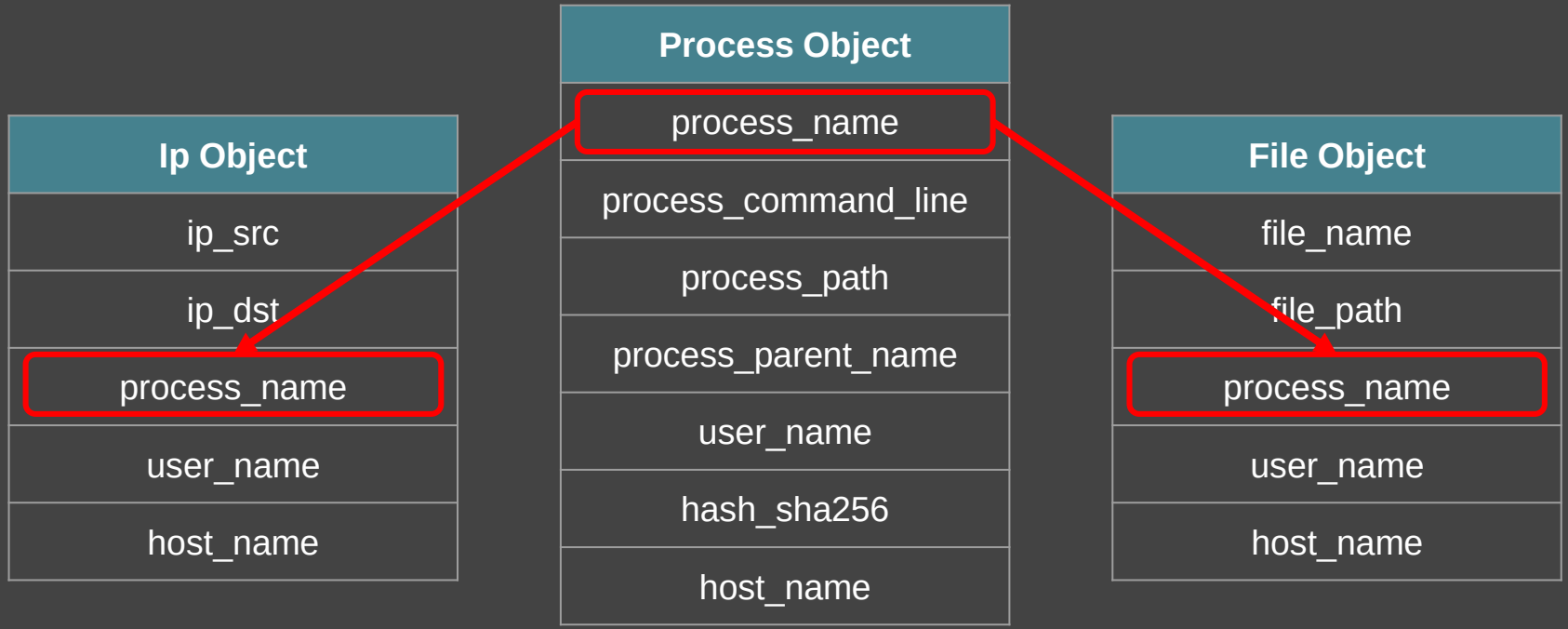
Ip Object
ip_src
ip_dst
process_name
user_name
host_name

Process Object
process_name
process_command_line
process_path
process_parent_name
user_name
hash_sha256
host_name

File Object
file_name
file_path
process_name
user_name
host_name

https://github.com/Cyb3rWard0g/OSSEM/tree/master/data_models

Data Model (Defining Data Objects)



Data Model (Defining Object Relationships)

Applicable Objects (Source)	Relationship	Applicable Objects (Destination)
Process	Created	File, Process, Win Registry Key, Service
File, Process, Win Registry Key, Service	Created_By	Process
Process	Parent_Of	Process
Process	Modified_Properties_Of	File, Win Registry Key, Service
Process	Renamed	File
File	Renamed_By	Process
Process	Connected_To	IP, Hostname

http://stixproject.github.io/data-model/1.2/cyboxVocabs/ObjectRelationshipVocab-1.1/

Example: Process use of network

PowerShell logs
Process command-line parameters
Process monitoring
Process use of network
Sensor health and status
Services
SSL/TLS inspection
System calls
Third-party application logs
User interface

VBR
Web application firewall logs
Web logs
Web proxy
Windows Error Reporting
Windows event logs
Windows Registry
WMI Objects

Process use of network: Process & IP Relationship

Applicable Objects (Source)	Relationship	Applicable Objects (Destination)
Process	Created	File, Process, Win Registry Key, Service
Process	Parent_Of	Process
File, Process, Win Registry Key, Service	Created_By	Process
Process	Modified_Properties_Of	File, Win Registry Key, Service
Process	Renamed	File
File	Renamed_By	Process
Process	Connected_To	IP, Hostname

http://stixproject.github.io/data-model/1.2/cyboxVocabs/ObjectRelationshipVocab-1.1/

Data Source: Process use of network

Process Object
process_name
process_command_line
process_path
process_parent_name
user_name
hash_sha256
host_name

Ip Object
ip_src
ip_dst
process_name
user_name
host_name

Linking it to data sensors (Sysmon)

Process Object
process_name
process_command_line
process_path
process_parent_name
user_name
hash_sha256
host_name

Sysmon (1)
Image
Command_line
User
Hashes
ParentImage

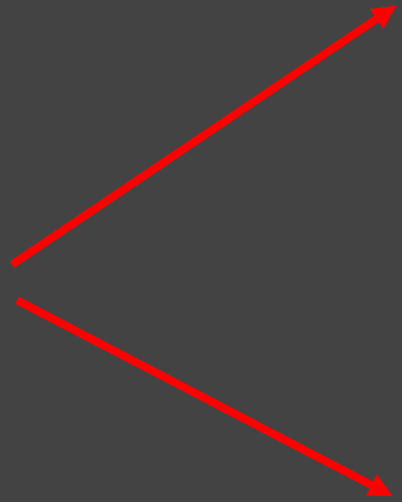
Linking it to data sensors (Windows Security)

Process Object
process_name
process_command_line
process_path
process_parent_name
user_name
hash_sha256
host_name

Windows Security (4688)
NewProcessName
CommandLine
SubjectUserName
ParentProcessName

Do I have what I need?

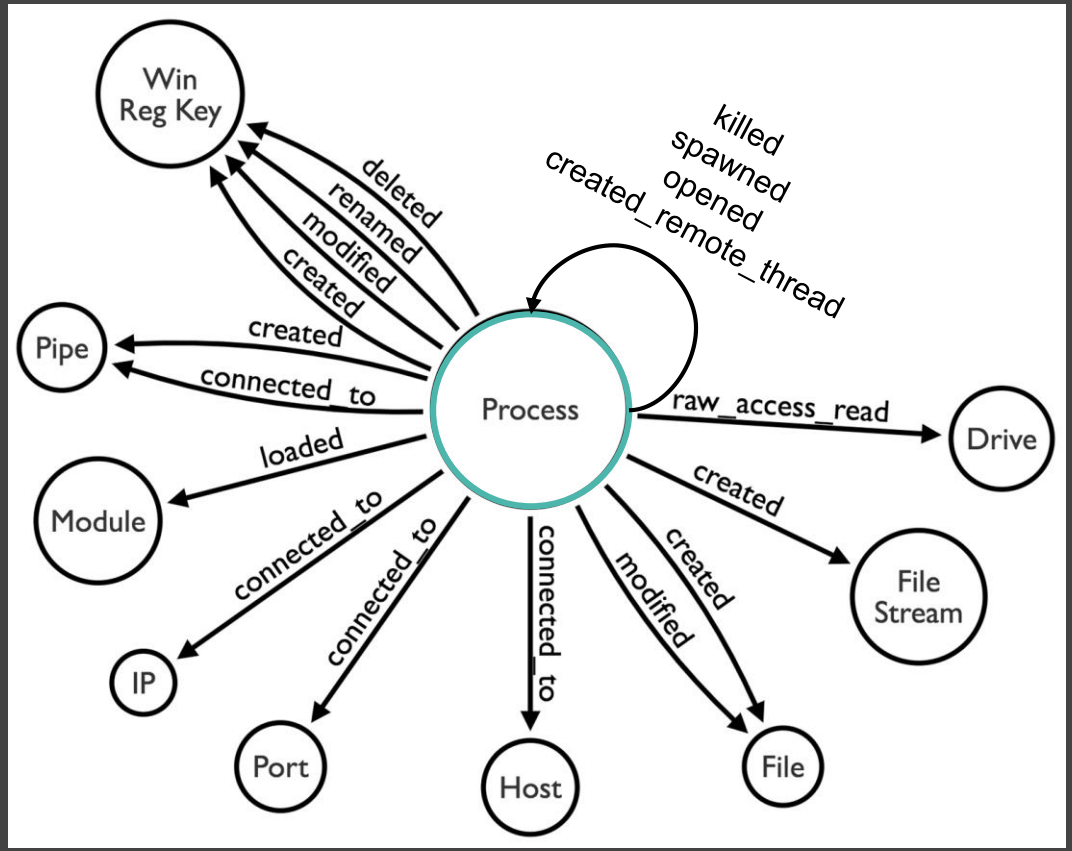
Process Object
process_name
process_command_line
process_path
process_parent_name
user_name
hash_sha256
host_name



Sysmon
Image
Command_line
User
Hashes
ParentImage

Windows Security (4688)
NewProcessName
CommandLine
SubjectUserName
ParentProcessName

Do I have what I need? (Modeling Sysmon Events)



Do I have what I need? (mmm... tell me more)

Do I have what I need? (What we propose)

Windows

Execution

Service Execution

Process Monitoring

Process Object
process_name
process_command_line
process_path
process_parent_name
user_name
hash_sha256
host_name

Sysmon

[Remember] Adversaries can influence your data..

"**UtcTime**">2018-04-11 05:25:02.955
"**ProcessGuid**">{A98268C1-9C2E-5ACD-0000-0010396CAB00}
"**ProcessId**">4756
"**Image**">C:\Windows\System32\conhost.exe
"**FileVersion**">10.0.16299.15 (WinBuild.160101.0800)
"**Description**">Console Window Host
"**Product**">Microsoft® Windows® Operating System
"**Company**">Microsoft Corporation
"**CommandLine**">\??\C:\WINDOWS\system32\conhost.exe 0xffffffff -ForceV1
"**CurrentDirectory**">C:\WINDOWS
"**User**">DESKTOP-WARDOG\wardog
"**LogonGuid**">{A98268C1-95F2-5ACD-0000-002019620F00}
"**LogonId**">0xf6219
"**TerminalSessionId**">1
"**IntegrityLevel**">Medium
"**Hashes**">SHA1=B0BF5AC2E81BBF597FAD5F349FEEB32CAC449FA2
"**ParentProcessGuid**">{A98268C1-9C2E-5ACD-0000-00100266AB00}
"**ParentProcessId**">240
"**ParentImage**">C:\Windows\System32\cmd.exe
"**ParentCommandLine**">"C:\WINDOWS\system32\cmd.exe"

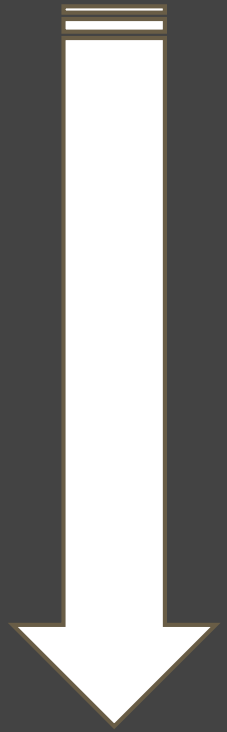
High Attacker Influence Rating!!!

https://github.com/mattifestation/BHUSA2018_Sysmon/blob/master/Whitepaper_Subverting_Sysmon.pdf

So, what can I measure *now*?

Do I know what I have?

Is this data what I need?



Not all data sources are created equal, data quality matters.

If data needed for a hunting engagement does not meet specific requirements defined by the hunt team, then the data is not considered quality data since it is affecting the intended purpose of it.

“Data are of high quality if they are fit for their intended uses in operations, decision making and planning."

”

- Julian’s Quality Handbook -

Data Quality Dimensions

Data Quality	Characteristics Description	Example Metric
Accuracy	A quality of that which is free of error. A qualitative assessment of freedom from error, with a high assessment corresponding to a small error. (FIPS Pub 11-3)	Percent of values that are correct when compared to the actual value. For example, M=Male when the subject is Male.
Completeness	Completeness is the degree to which values are present in the attributes that require them. (Data Quality Foundation)	Percent of data fields having values entered into them.
Consistency	Consistency is a measure of the degree to which a set of data satisfies a set of constraints. (Data Quality Management and Technology)	Percent of matching values across tables/files/records.
Timeliness	As a synonym for currency, timeliness represents the degree to which specified data values are up to date. (Data Quality Management and Technology)	Percent of data available within a specified threshold time frame (e.g., days, hours, minutes).
Uniqueness	The state of being the only one of its kind. Being without an equal or equivalent.	Percent of records having a unique primary key.
Validity	The quality of data that is founded on an adequate system of classification and is rigorous enough to compel acceptance. (DoD 8320.1-M)	Percent of data having values that fall within their respective domain of allowable values.

http://mitiq.mit.edu/ICIQ/Documents/IQ%20Conference%201996/Papers/DODGuidelinesonDataQualityManagement.pdf

Data Completeness

- How much data that is required/needed is available in my network?
- Are all required/needed data fields and values recorded?

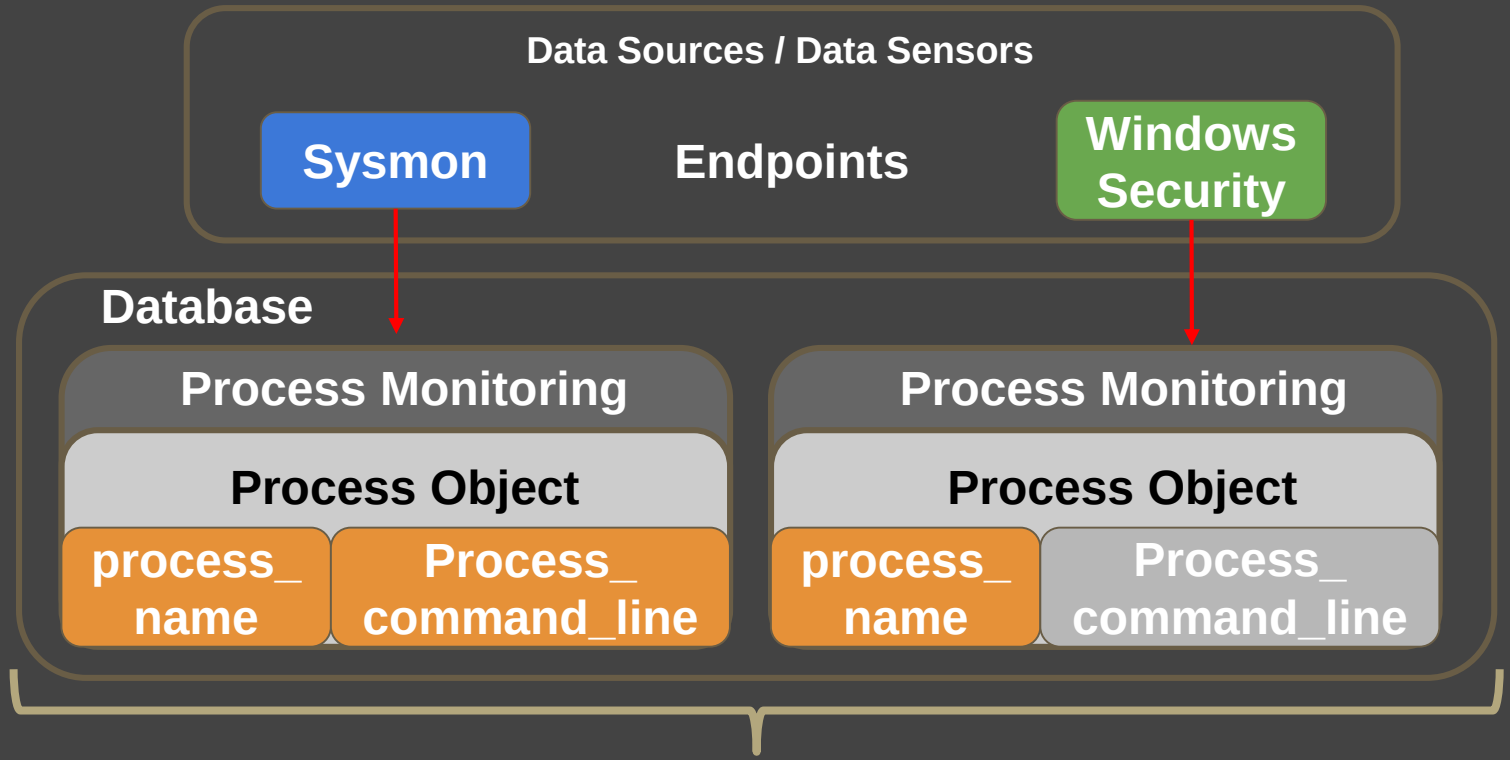
Data Consistency

- Can we match required/needed fields across data sources?

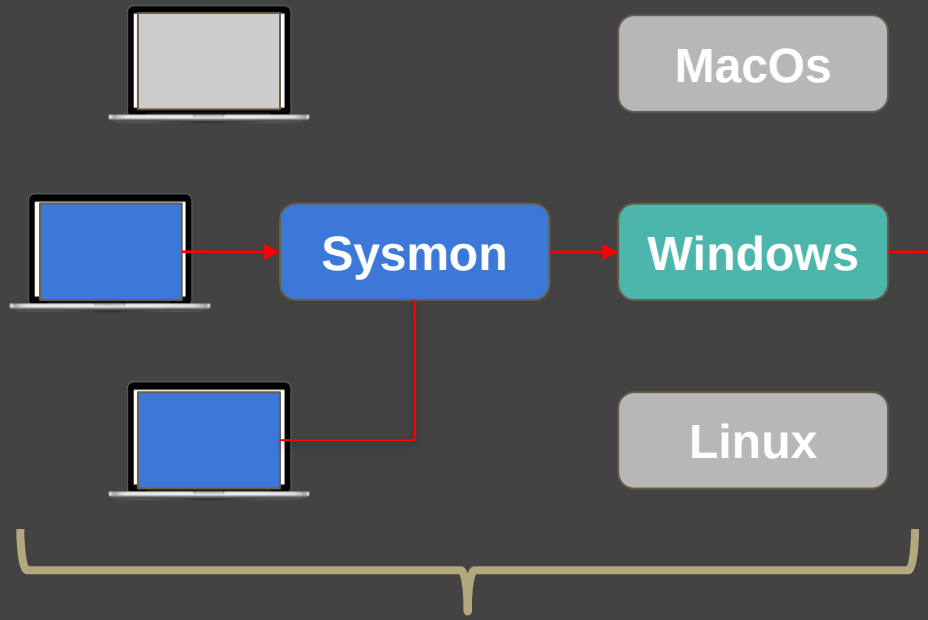
Data Timeliness

- Does my data represent reality?
- How far back in time can I hunt with required/needed data?

Completeness: Is the expected data complete?



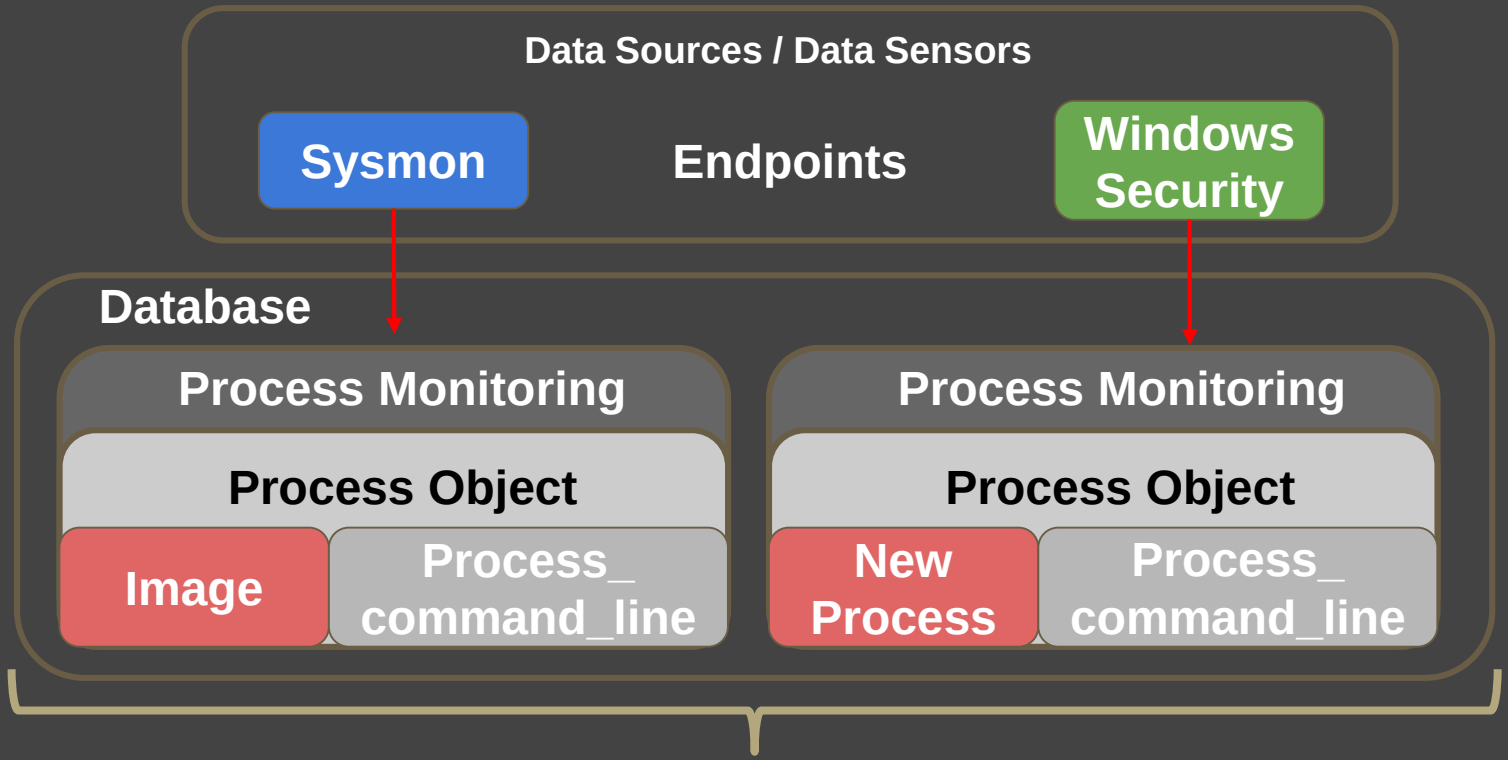
Completeness: Percentage of network covered?



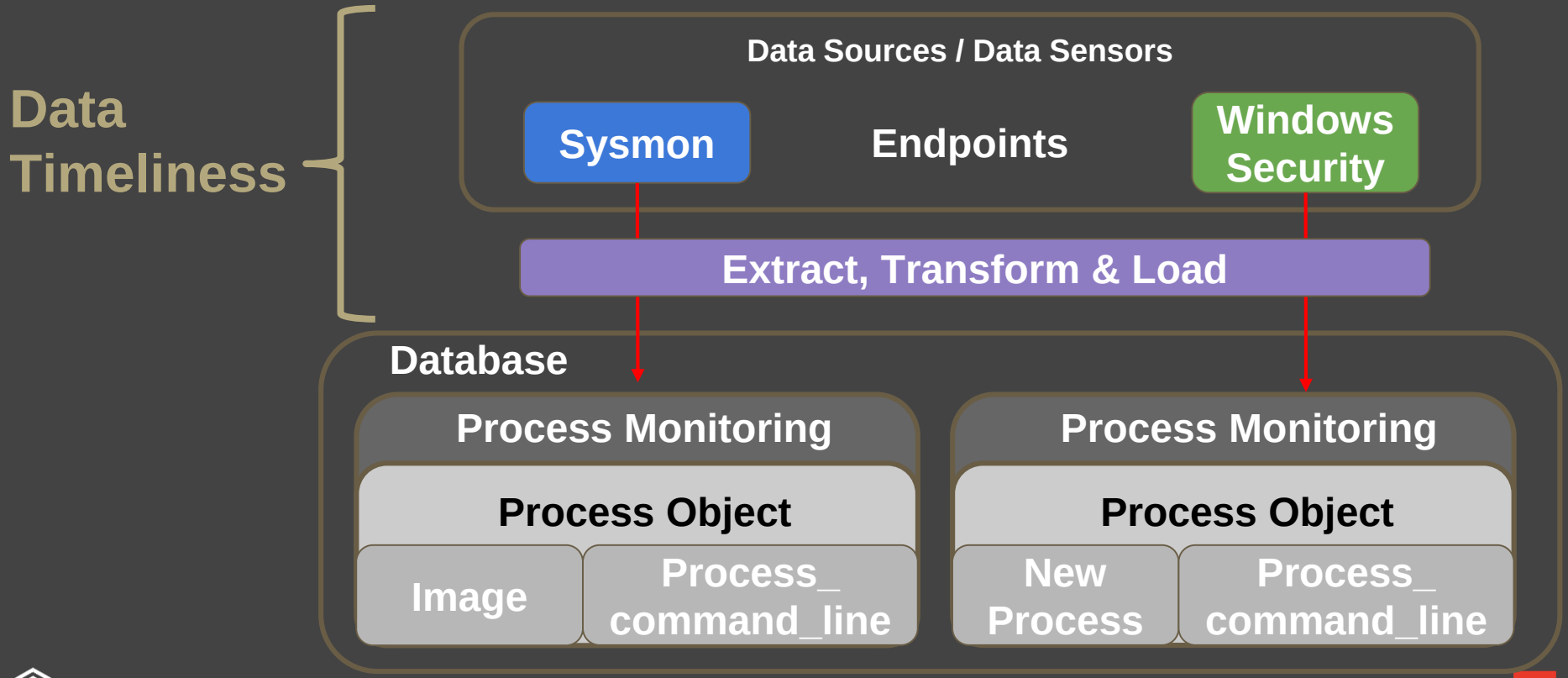
Process Object
process_name
process_command_line
process_path
process_parent_name
user_name
hash_sha256
host_name

Data Completeness

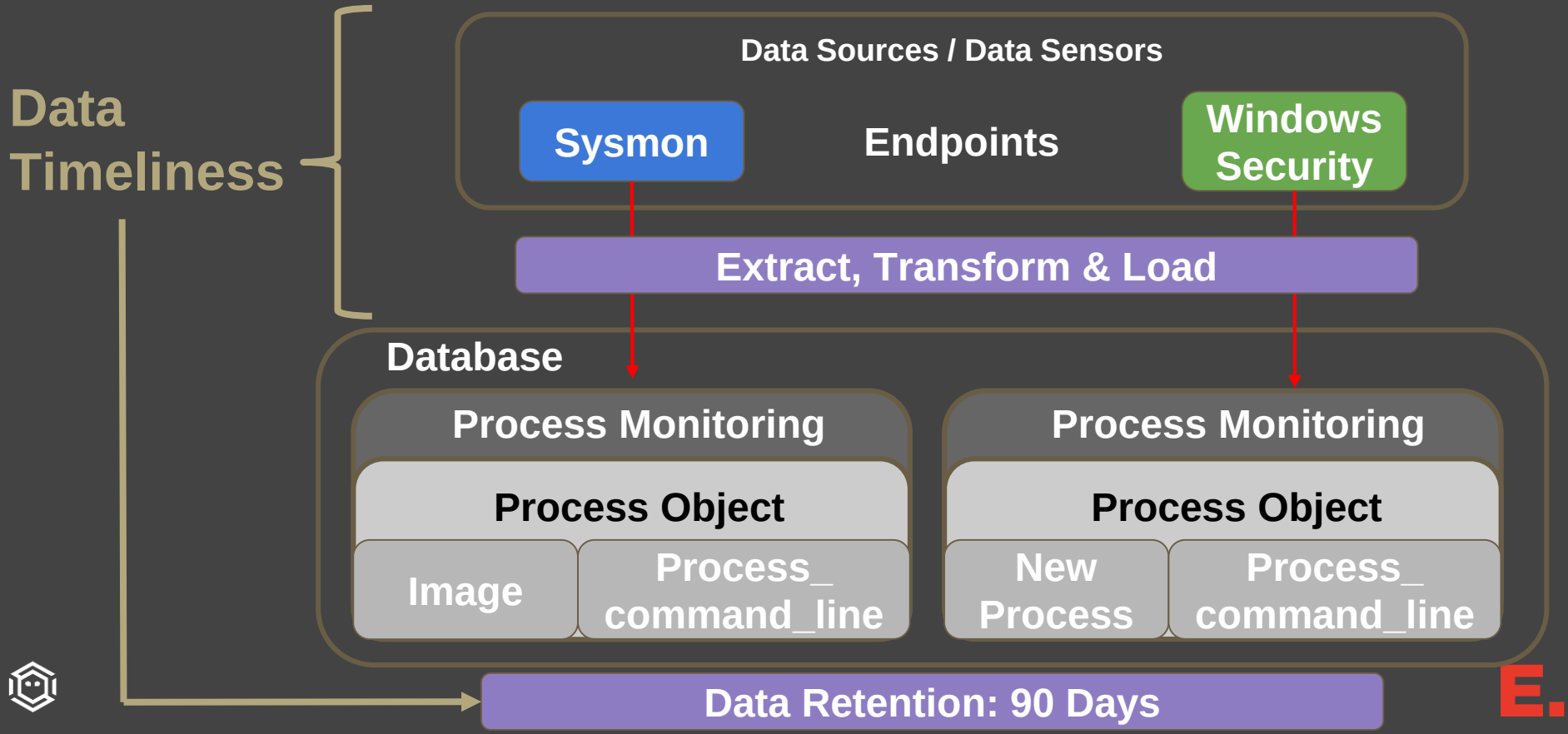
Consistency: Across all data sources?



Timeliness: Does my data represent reality?



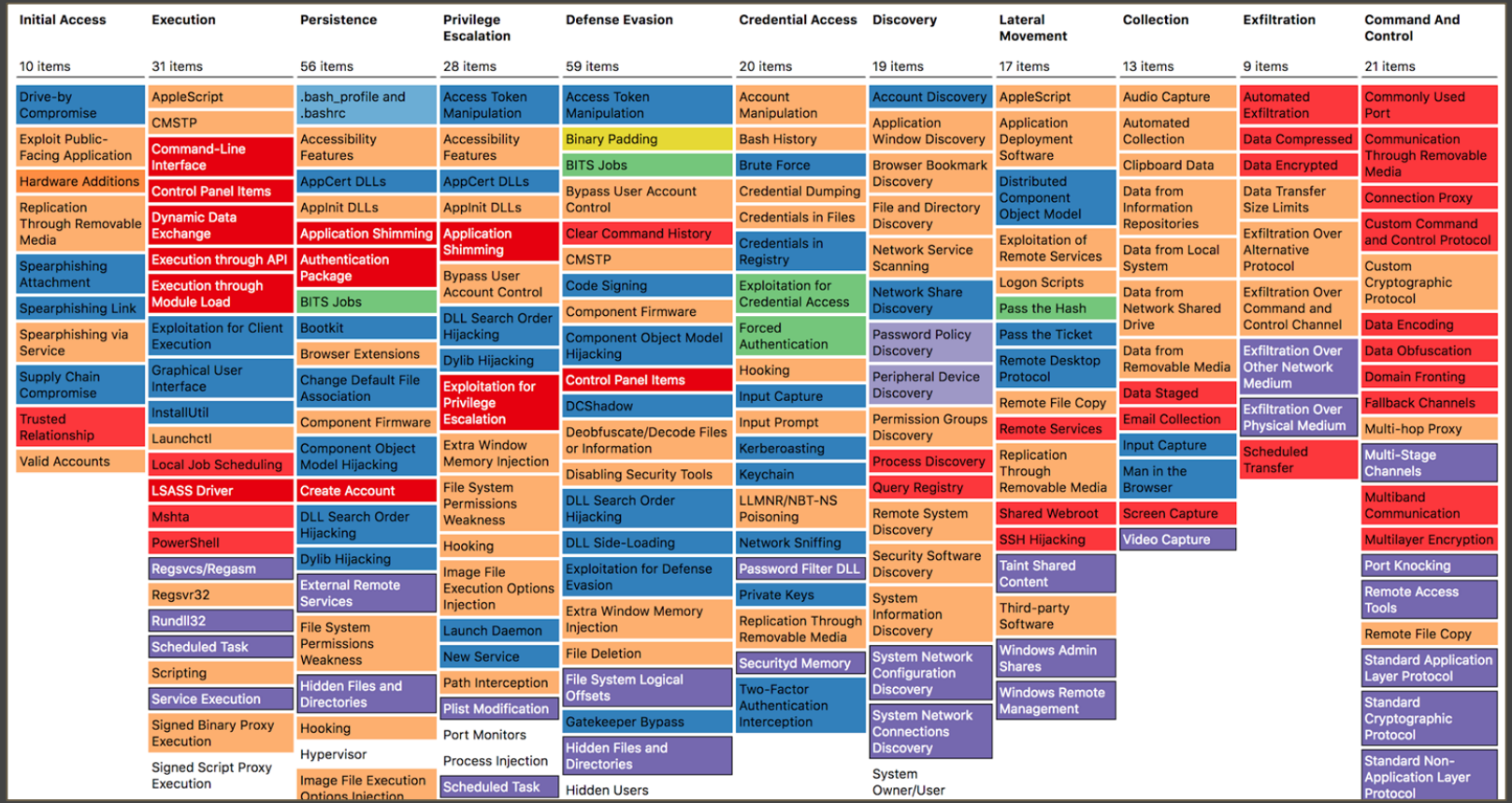
Timeliness: Does my data represent reality?



A few hunt metrics you *could* measure:

- What percentage of recommended data is available for a hunt?
- What percentage of the expected data is complete for a hunt?
- What percentage of my environment could I cover in an hunt based on the available recommend data?
- How far back in time can I hunt with recommended data?
- What percentage of my data sources are consistent across all the data provided by data sensors?
- Do I have the right technology or skills to hunt?

Now, this makes more sense?? A little bit?

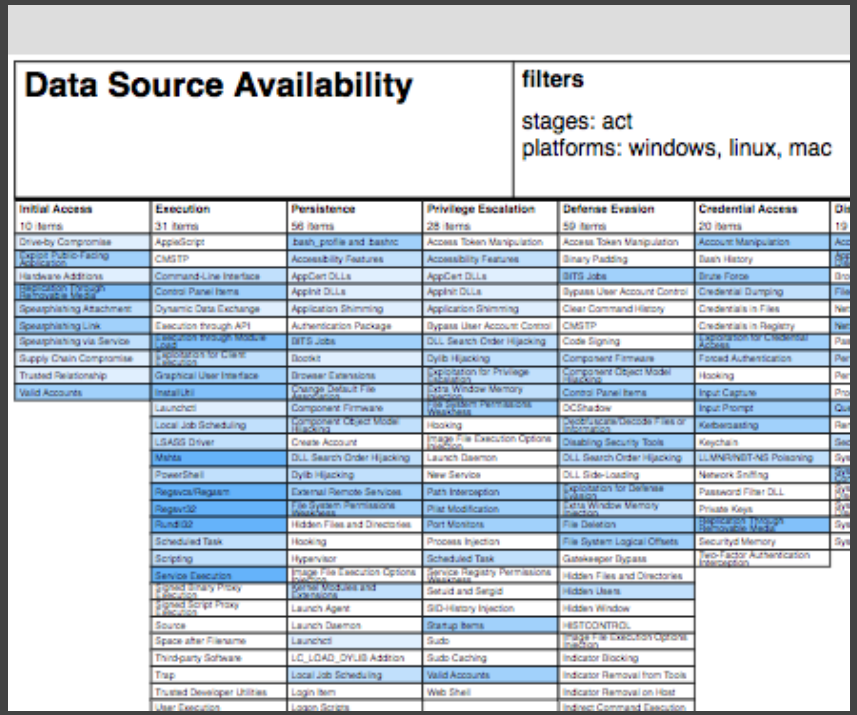


Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command And Control
10 items	31 items	56 items	28 items	59 items	20 items	19 items	17 items	13 items	9 items	21 items
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Data Compressed	Communication Through Removable Media
Hardware Additions	Command-Line Interface	AppCert DLLs	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Distributed Component Object Model	Clipboard Data	Data Encrypted	Connection Proxy
Replication Through Removable Media	Control Panel Items	AppInit DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Remote Services	Data from Information Repositories	Data Transfer Size Limits	Custom Command and Control Protocol
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Application Shimming	Clear Command History	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Local System	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Spearphishing Link	Execution through API	Authentication Package	Bypass User Account Control	CMSTP	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Network Shared Drive	Exfiltration Over Command and Control Channel	Data Encoding
Spearphishing via Service	Execution through Module Load	BITS Jobs	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Password Policy Discovery	Pass the Ticket	Data from Removable Media	Exfiltration Over Other Network Medium	Data Obfuscation
Supply Chain Compromise	Exploitation for Client Execution	Bootkit	Dylib Hijacking	Component Firmware	Forced Authentication	Peripheral Device Discovery	Remote Desktop Protocol	Data Staged	Exfiltration Over Physical Medium	Domain Fronting
Trusted Relationship	Graphical User Interface	Browser Extensions	Exploitation for Privilege Escalation	Component Object Model Hijacking	Hooking	Permission Groups Discovery	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels
Valid Accounts	InstallUtil	Change Default File Association	Extra Window Memory Injection	Control Panel Items	Input Capture	Process Discovery	Remote Services	Input Capture		Multi-hop Proxy
	Launchctl	Component Firmware	File System Permissions Weakness	DCShadow	Input Prompt	Query Registry	Replication Through Removable Media	Man in the Browser		Multi-Stage Channels
	Local Job Scheduling	Component Object Model Hijacking	Hooking	Deobfuscate/Decode Files or Information	Kerberoasting	Remote System Discovery	Shared Webroot	Screen Capture		Multiband Communication
	LSASS Driver	Create Account	Image File Execution Options Injection	Disabling Security Tools	Keychain	Security Software Discovery	SSH Hijacking	Video Capture		Multilayer Encryption
	Mshta	DLL Search Order Hijacking	Launch Daemon	DLL Search Order Hijacking	LLMNR/NBT-NS Poisoning	System Information Discovery	Taint Shared Content			Port Knocking
	PowerShell	Dylib Hijacking	New Service	DLL Side-Loading	Network Sniffing	System Network Configuration Discovery	Third-party Software			Remote Access Tools
	Regsvcs/Regasm	External Remote Services	Path Interception	Exploitation for Defense Evasion	Password Filter DLL	System Network Connections Discovery	Windows Admin Shares			Remote File Copy
	Regsvr32	File System Permissions Weakness	Plist Modification	Extra Window Memory Injection	Private Keys	System Owner/User	Windows Remote Management			Standard Application Layer Protocol
	Rundll32	Hidden Files and Directories	Port Monitors	File Deletion	Replication Through Removable Media					Standard Cryptographic Protocol
	Scheduled Task	Hooking	Process Injection	File System Logical Offsets	Securityd Memory					Standard Non-Application Layer Protocol
	Scripting	Hypervisor	Scheduled Task	Gatekeeper Bypass	Two-Factor Authentication Interception					
	Service Execution	Image File Execution Options Injection		Hidden Files and Directories						
	Signed Binary Proxy Execution			Hidden Users						
	Signed Script Proxy Execution									

A little better..

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command And Control
10 items	31 items	56 items	28 items	59 items	20 items	19 items	17 items	13 items	9 items	21 items
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Account Discovery	AppleScript	Audio Capture	Automated Exfiltration	Commonly Used Port
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Application Window Discovery	Application Deployment Software	Automated Collection	Data Compressed	Communication Through Removable Media
Hardware Additions	Command-Line Interface	AppCert DLLs	AppCert DLLs	BITS Jobs	Brute Force	Browser Bookmark Discovery	Distributed Component Object Model	Clipboard Data	Data Encrypted	Connection Proxy
Replication Through Removable Media	Control Panel Items	AppInit DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	File and Directory Discovery	Exploitation of Remote Services	Data from Information Repositories	Data Transfer Size Limits	Custom Command and Control Protocol
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Application Shimming	Clear Command History	Credentials in Files	Network Service Scanning	Logon Scripts	Data from Local System	Exfiltration Over Alternative Protocol	Custom Cryptographic Protocol
Spearphishing Link	Execution through API	Authentication Package	Bypass User Account Control	CMSTP	Credentials in Registry	Network Share Discovery	Pass the Hash	Data from Network Shared Drive	Exfiltration Over Command and Control Channel	Data Encoding
Spearphishing via Service	Execution through Module Load	BITS Jobs	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Password Policy Discovery	Pass the Ticket	Data from Removable Media	Exfiltration Over Other Network Medium	Data Obfuscation
Supply Chain Compromise	Exploitation for Client Execution	Bootkit	Dylib Hijacking	Component Firmware	Forced Authentication	Peripheral Device Discovery	Remote Desktop Protocol	Data Staged	Exfiltration Over Physical Medium	Domain Fronting
Trusted Relationship	Graphical User Interface	Browser Extensions	Exploitation for Privilege Escalation	Component Object Model Hijacking	Hooking	Permission Groups Discovery	Remote File Copy	Email Collection	Scheduled Transfer	Fallback Channels
Valid Accounts	InstallUtil	Change Default File Association	Extra Window Memory Injection	Control Panel Items	Input Capture	Process Discovery	Remote Services	Input Capture		Multi-hop Proxy
	Launchctl	Component Firmware	File System Permissions Weakness	DCShadow	Input Prompt	Query Registry	Replication Through Removable Media	Man in the Browser		Multi-Stage Channels
	Local Job Scheduling	Component Object Model Hijacking	Hooking	Deobfuscate/Decode Files or Information	Kerberoasting	Remote System Discovery	Shared Webroot	Screen Capture		Multiband Communication
	LSASS Driver	Create Account	Image File Execution Options Injection	Disabling Security Tools	Keychain	Security Software Discovery	SSH Hijacking	Video Capture		Multilayer Encryption
	Mshta	DLL Search Order Hijacking	Launch Daemon	DLL Search Order Hijacking	LLMNR/NBT-NS Poisoning	System Information Discovery	Taint Shared Content			Port Knocking
	PowerShell	Dylib Hijacking	New Service	DLL Side-Loading	Network Sniffing	System Network Configuration Discovery	Third-party Software			Remote Access Tools
	Regsvcs/Regasm	External Remote Services	Path Interception	Exploitation for Defense Evasion	Password Filter DLL	System Network Connections	Windows Admin Shares			Remote File Copy
	Regsvr32	File System Permissions Weakness	Plist Modification	Extra Window Memory Injection	Private Keys		Windows Remote Management			Standard Application Layer Protocol
	Rundll32	Hidden Files and Directories		File Deletion	Replication Through Removable Media					Standard Cryptographic Protocol
	Scheduled Task	Hooking		File System Logical Offsets	Securityd Memory					Standard Non-
	Scripting			Gatekeeper Bypass	Two-Factor Authentication Interception					
	Service Execution									
	Signed Binary Proxy									

We visualize each type of thing *separately*

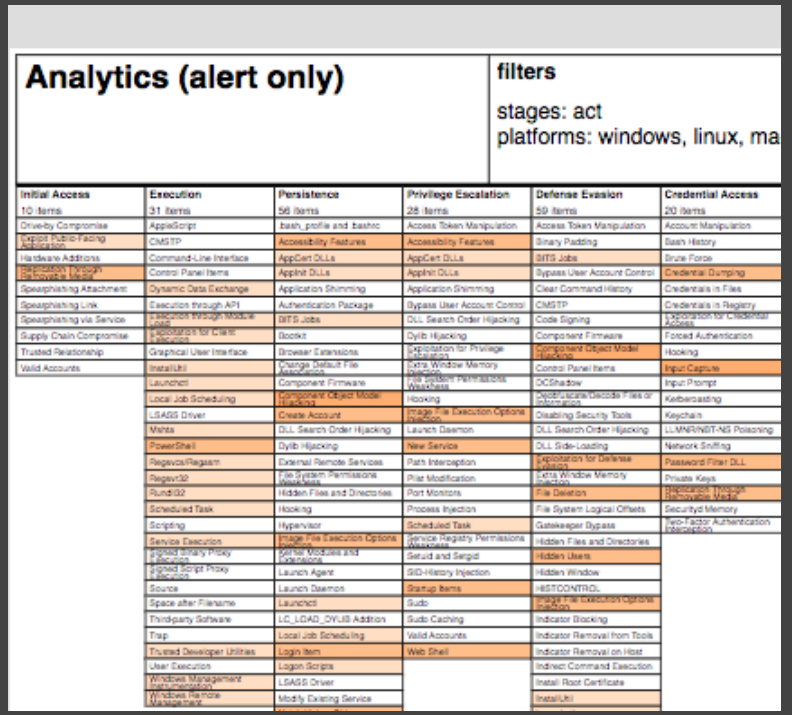


Data Source Availability

filters

stages: act
platforms: windows, linux, mac

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Di
10 items	31 items	56 items	28 items	59 items	20 items	19
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation	Ac
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History	Ap
Hardware Additions	Command-Line Interface	AppCert DLLs	AppCert DLLs	BITS Jobs	Brute Force	Br
Replication Through Removable Media	Control Panel Items	AppInit DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping	File
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Application Shimming	Clear Command History	Credentials in Files	Ne
Spearphishing Link	Execution through API	Authentication Package	Bypass User Account Control	CMSTP	Credentials in Registry	Ne
Spearphishing via Service	Execution through Module Load	BITS Jobs	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access	Pa
Supply Chain Compromise	Exploitation for Client Execution	Bootkit	Dylib Hijacking	Component Firmware	Forced Authentication	Pe
Trusted Relationship	Graphical User Interface	Browser Extensions	Exploitation for Privilege Escalation	Component Object Model Hijacking	Hooking	Pe
Valid Accounts	InstallUtil	Change Default File Association	Extra Window Memory Injection	Control Panel Items	Input Capture	Pr
	Launchctl	Component Firmware	File System Permissions Weakness	DCShadow	Input Prompt	Qu
	Local Job Scheduling	Component Object Model Hijacking	Hooking	Deobfuscate/Decode Files or Information	Kerberoasting	Re
	LSASS Driver	Create Account	Image File Execution Options Injection	Disabling Security Tools	Keychain	Se
	Mshta	DLL Search Order Hijacking	Launch Daemon	DLL Search Order Hijacking	LLMNR/NBT-NS Poisoning	Sy
	PowerShell	Dylib Hijacking	New Service	DLL Side-Loading	Network Sniffing	Sy
	Regsvcs/Regasm	External Remote Services	Path Interception	Exploitation for Defense Evasion	Password Filter DLL	Sy
	Regsvr32	File System Permissions Weakness	Plist Modification	Extra Window Memory Injection	Private Keys	Sy
	Rundll32	Hidden Files and Directories	Port Monitors	File Deletion	Replication Through Removable Media	Sy
	Scheduled Task	Hooking	Process Injection	File System Logical Offsets	Securityd Memory	Sy
	Scripting	Hypervisor	Scheduled Task	Gatekeeper Bypass	Two-Factor Authentication Interception	
	Service Execution	Image File Execution Options Injection	Service Registry Permissions Weakness	Hidden Files and Directories		
	Signed Binary Proxy Execution	Kernel Modules and Extensions	Setuid and Setgid	Hidden Users		
	Signed Script Proxy Execution	Launch Agent	SID-History Injection	Hidden Window		
	Source	Launch Daemon	Startup Items	HISTCONTROL		
	Space after Filename	Launchctl	Sudo	Image File Execution Options Injection		
	Third-party Software	LC_LOAD_DYLIB Addition	Sudo Caching	Indicator Blocking		
	Trap	Local Job Scheduling	Valid Accounts	Indicator Removal from Tools		
	Trusted Developer Utilities	Login Item	Web Shell	Indicator Removal on Host		
	User Execution	Logon Scripts		Indirect Command Execution		



Analytics (alert only)

filters

stages: act
platforms: windows, linux, ma

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access
10 items	31 items	56 items	28 items	59 items	20 items
Drive-by Compromise	AppleScript	.bash_profile and .bashrc	Access Token Manipulation	Access Token Manipulation	Account Manipulation
Exploit Public-Facing Application	CMSTP	Accessibility Features	Accessibility Features	Binary Padding	Bash History
Hardware Additions	Command-Line Interface	AppCert DLLs	AppCert DLLs	BITS Jobs	Brute Force
Replication Through Removable Media	Control Panel Items	AppInit DLLs	AppInit DLLs	Bypass User Account Control	Credential Dumping
Spearphishing Attachment	Dynamic Data Exchange	Application Shimming	Application Shimming	Clear Command History	Credentials in Files
Spearphishing Link	Execution through API	Authentication Package	Bypass User Account Control	CMSTP	Credentials in Registry
Spearphishing via Service	Execution through Module Load	BITS Jobs	DLL Search Order Hijacking	Code Signing	Exploitation for Credential Access
Supply Chain Compromise	Exploitation for Client Execution	Bootkit	Dylib Hijacking	Component Firmware	Forced Authentication
Trusted Relationship	Graphical User Interface	Browser Extensions	Exploitation for Privilege Escalation	Component Object Model Hijacking	Hooking
Valid Accounts	InstallUtil	Change Default File Association	Extra Window Memory Injection	Control Panel Items	Input Capture
	Launchctl	Component Firmware	File System Permissions Weakness	DCShadow	Input Prompt
	Local Job Scheduling	Component Object Model Hijacking	Hooking	Deobfuscate/Decode Files or Information	Kerberoasting
	LSASS Driver	Create Account	Image File Execution Options Injection	Disabling Security Tools	Keychain
	Mshta	DLL Search Order Hijacking	Launch Daemon	DLL Search Order Hijacking	LLMNR/NBT-NS Poisoning
	PowerShell	Dylib Hijacking	New Service	DLL Side-Loading	Network Sniffing
	Regsvcs/Regasm	External Remote Services	Path Interception	Exploitation for Defense Evasion	Password Filter DLL
	Regsvr32	File System Permissions Weakness	Plist Modification	Extra Window Memory Injection	Private Keys
	Rundll32	Hidden Files and Directories	Port Monitors	File Deletion	Replication Through Removable Media
	Scheduled Task	Hooking	Process Injection	File System Logical Offsets	Securityd Memory
	Scripting	Hypervisor	Scheduled Task	Gatekeeper Bypass	Two-Factor Authentication Interception
	Service Execution	Image File Execution Options Injection	Service Registry Permissions Weakness	Hidden Files and Directories	
	Signed Binary Proxy Execution	Kernel Modules and Extensions	Setuid and Setgid	Hidden Users	
	Signed Script Proxy Execution	Launch Agent	SID-History Injection	Hidden Window	
	Source	Launch Daemon	Startup Items	HISTCONTROL	
	Space after Filename	Launchctl	Sudo	Image File Execution Options Injection	
	Third-party Software	LC_LOAD_DYLIB Addition	Sudo Caching	Indicator Blocking	
	Trap	Local Job Scheduling	Valid Accounts	Indicator Removal from Tools	
	Trusted Developer Utilities	Login Item	Web Shell	Indicator Removal on Host	
	User Execution	Logon Scripts		Indirect Command Execution	
	Windows Management Instrumentation	LSASS Driver		Install Root Certificate	
	Windows Remote Management	Modify Existing Service		InstallUtil	

What about DETECTION of adversary techniques?

Does measuring the quality of my data really help me find compromise?

Using statistics to universally measure

Credit to the *Towards Data Science* blog and their article “Beyond Accuracy: Precision and Recall”

Threat Hunting vs Detection

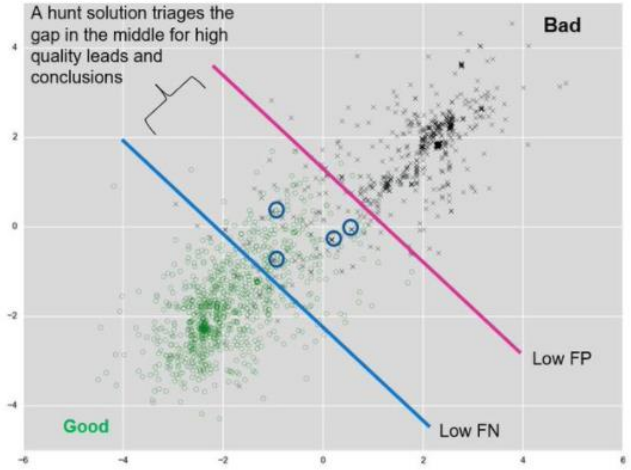
Chris Gerritz @gerritzc · Mar 15

Why most prevention and detection tools make poor #ThreatHunting solutions in one slide. @InfocyteInc

Threat Hunting vs Detection - The Optimization Problem

Why do most defensive tools make poor hunt tools?

- Prevention and real-time detection solutions are **optimized for low False Positive (FP) Alerting**
- Hunt solutions are **optimized for low False Negatives (FN)**
  - For Hunting: Anomalies, Outliers, and Suspicious Activity are **leads**, not FPs to be tuned out
  - A good hunt solution sorts and scores leads, then enables a quick path to verify and investigate to a conclusion



Precision is being tolerant of False Positives

True Positive - a malicious thing you correctly identify as malicious

False Positive - a benign thing you incorrectly identify as malicious

Precision = (True Positives/(True Positives + False Positives))

Example:

- 100 events
- 74 TPs
- 26 FPs
- 0.74 precision

Recall is how well you find malicious activity

True Positive - a malicious thing you correctly identify as malicious

False Negative - a malicious thing you incorrectly identify as benign

Recall = (True Positives/(True Positives + False Negatives))

Example:

- 100 events
- 55 TPs
- 21 FPs
- 24 FNs
- 0.69 recall

But be wary of extremes

Perfect recall finds all the threat activity while also generating tons of false positives.

Perfect precision might find nothing to be malicious, missing all the threat activity without generating any false positives.

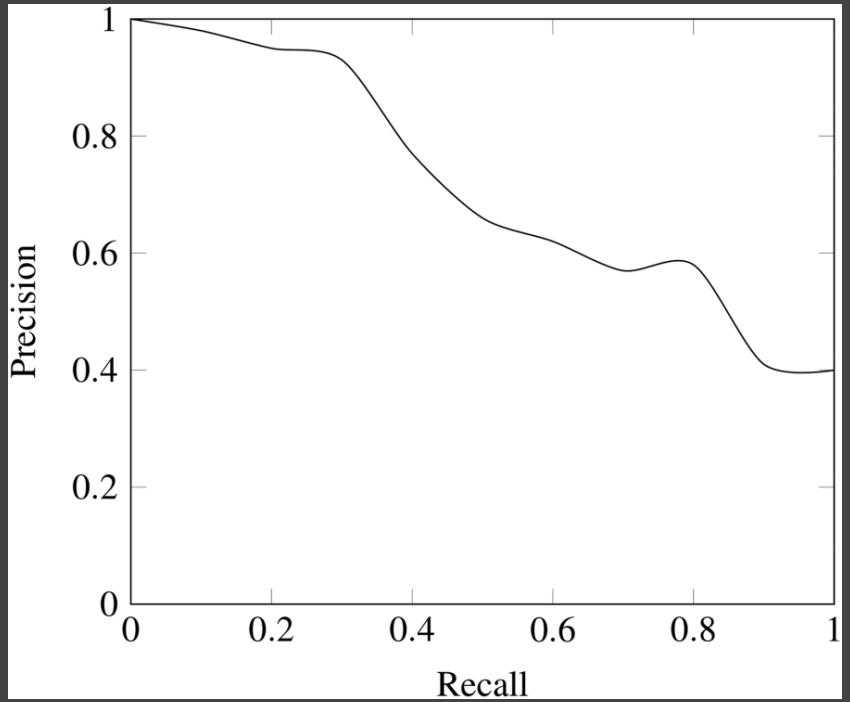


Image credit: *https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c*

Wait a darn minute, you skipped something!

How do you determine False Negatives?

1. Adversary simulation tools will show you
2. Your red team will tell you
3. Third parties doing offensive security work for you will deliver a report on it

Note:

- Many organizations will begin testing in a lab or other contrived environment which need to be representative of your enterprise
- If you're hunting, your environment will absolutely change between measurements
- This is best effort
- Vendors often have helpful telemetry, so they have a leg up on measuring recall
- Data quality directly influences precision and recall

Quantifying detections using precision and recall

An F1 Score is a value between 0 and 1 that describes how well you judge something to be malicious using precision and recall:

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Image credit: *https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c*

Those examples from before gave us a precision of 0.74 and a recall of 0.69, let's see our F1 score:

F1 = 2 * (0.51/1.43) = 0.71

But this value doesn't mean anything to us - at least not *yet*.

https://en.wikipedia.org/wiki/F1_score

F1 scores are a kind of tolerance for classification

Threshold	TP	FP	TN	FN
0.0	50	50	0	0
0.1	48	47	3	2
0.2	47	40	9	4
0.3	45	31	16	8
0.4	44	23	22	11
0.5	42	16	29	13
0.6	36	12	34	18
0.7	30	11	38	21
0.8	20	4	43	33
0.9	12	3	45	40
1.0	0	0	50	50

What does a 0.71 mean? It means we're capable of finding 60% of malicious things.

Image credit: *https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c*

F1 scores are a kind of tolerance for classification

Threshold	TP	FP	TN	FN
0.0	50	50	0	0
0.1	48	47	3	2
0.2	47	40	9	4
0.3	45	31	16	8
0.4	44	23	22	11
0.5	42	16	29	13
0.6	36	12	34	18
0.7	30	11	38	21
0.8	20	4	43	33
0.9	12	3	45	40
1.0	0	0	50	50

Image credit: *https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c*

Do you have a requirement to find 80% of malicious things? Try a threshold of 0.4.

Work backwards to figure out acceptable precision and recall values for this goal.

We can even try to find ideal thresholds!

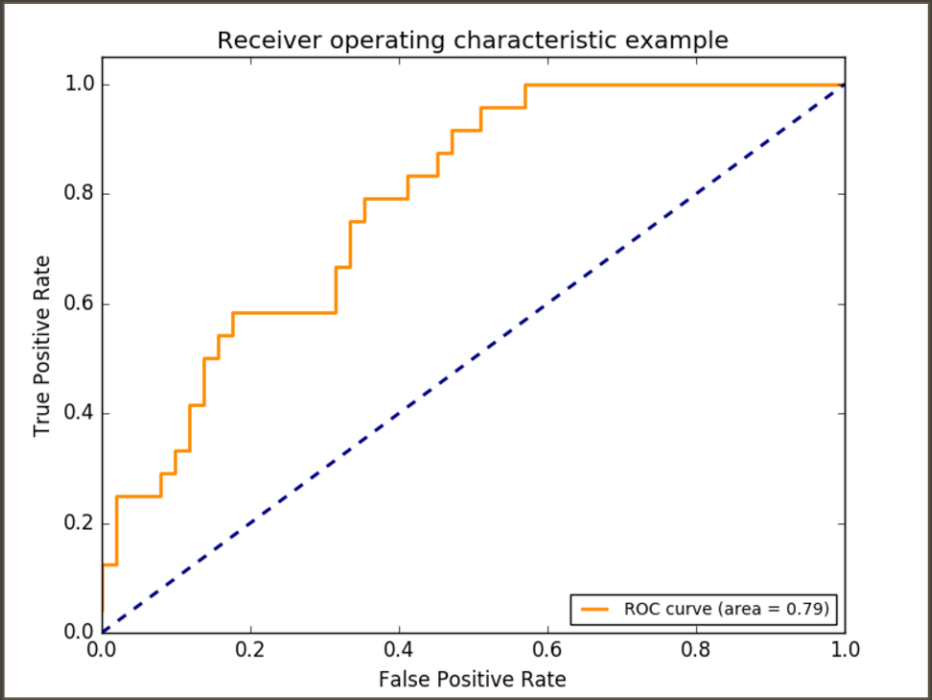


Image credit: qiita.com

Data science gives us tools for judging how well a classifier works, like how well we judge something to be malicious - this is what many machine learning malware classifiers do.

First, a Receiver Operator Characteristic (ROC) curve looks at TP and FP rates on X and Y axes to help display how well our chosen threshold (F1 score) works.

Measuring the area under the curve (AUC) for ROC values is the *quantification* of how well our classification works given our tolerance for TP and FP values.

Plot these on a regular basis to determine if your detection improves or not.

Closing thoughts

We could keep going with this topic

Links and things (if we talked about it, here it is):

Thing	Link
MITRE ATT&CK	https://attack.mitre.org
MITRE CAR	https://car.mitre.org/wiki/Main_Page
MITRE common data model	https://car.mitre.org/wiki/Data_Model
Roberto's Github (Sysmon configs, hunter's playbook, HELK, OSSEM, Invoke-AttackAPI)	https://github.com/Cyb3rWard0g
Endgame Github (RTA, other things)	https://github.com/endgameinc/

Thank you

If you didn't get a chance to ask a question, come chat with us in person or reach out via social media!

Also we are thankful to the Beyond Data Science blog, MITRE, the ATT&CK team, authors of simulation tools and everyone out there helping change the landscape for defenders.

This is an appendix

All this is stuff we wanted you to have

Quick wins

Count	Technique
92	Remote File Copy
92	Standard Application Layer Protocol
91	Command-Line Interface
85	System Information Discovery
75	File and Directory Discovery
70	Credential Dumping
68	Process Discovery
67	Registry Run Keys /Start Folder
62	File Deletion
57	Input Capture

# Techniques	Data Source
149	Process monitoring
86	File monitoring
82	Process command-line parameters
36	API monitoring
34	Windows Registry
34	Process use of network
31	Packet capture
28	Authentication logs
23	Netflow/Enclave netflow
17	Binary File Metadata

Data Sources -> Adversary Techniques

# Techniques	Name
149	Process monitoring
86	File monitoring
82	Process command-line parameters
36	API monitoring
34	Windows Registry
34	Process use of network
31	Packet capture
28	Authentication logs
23	Netflow/Enclave netflow

# Techniques	Name
17	Binary file metadata
16	DLL monitoring
16	Network protocol analysis
14	Windows event logs
12	Loaded DLLs
9	System calls
8	SSL/TLS inspection
8	Malware reverse engineering
6	Anti-virus

Data Sources -> Adversary Techniques

# Techniques	Name
6	Data loss prevention
5	Application Logs
4	Network device logs
4	Windows Error Reporting
4	Network intrusion detection system
4	User interface
4	Web proxy
3	Kernel drivers
3	Services

# Techniques	Name
3	Email gateway
3	Third-party application logs
2	Mail server
2	Detonation chamber
2	MBR
2	Environment variable
2	BIOS
2	Host network interface
1	Web logs

Data Sources -> Adversary Techniques

# Techniques	Name
1	Asset Management
1	Web application firewall logs
1	EFI
1	DNS records
1	Browser extensions
1	Sensor health and status
1	Named Pipes
1	VBR
1	PowerShell logs

# Techniques	Name
1	Access Tokens
1	Digital Certificate Logs
1	WMI Objects