

Network Threat Hunting

Predavač: Aleksandar Mirković



Threat Hunting

- Introduction to Threat, Dwell Time, Cyber Security Problems
- Introduction to Threat Hunting
- Threat Hunting People, Process, Tools & Technology
- Threat Hunting Framework
 - Pyramid of Pain
 - Cyber Kill Chain
 - MITRE ATT&CK
- Detection Engineering
 - Data Source Visibility (Endpoint & Network)
 - MITRE SHIELD
- Types of Threat Hunting
- Threat Hunting Use Case
- Threat Hunting Case Study



New Threat Paradigm

■ Traditional Threat Definition:

- Threat = Capability + Intent

New Threat Definition:

- Threat = Capability + Intent + Knowledge
 - **Capability** includes tools and ability to access
 - **Intent** is the motivation
 - **Knowledge** is specific, sophisticated ability to operate within a system/network after gaining access

New Threat Paradigm most applicable to high level threats

The Attacker's Advantage

- They only need to be successful once
- Determined, skilled and often funded adversaries
- Custom malware, 0days, multiple attack vectors, social engineering
- Can be Persistent



The Defender Disadvantage

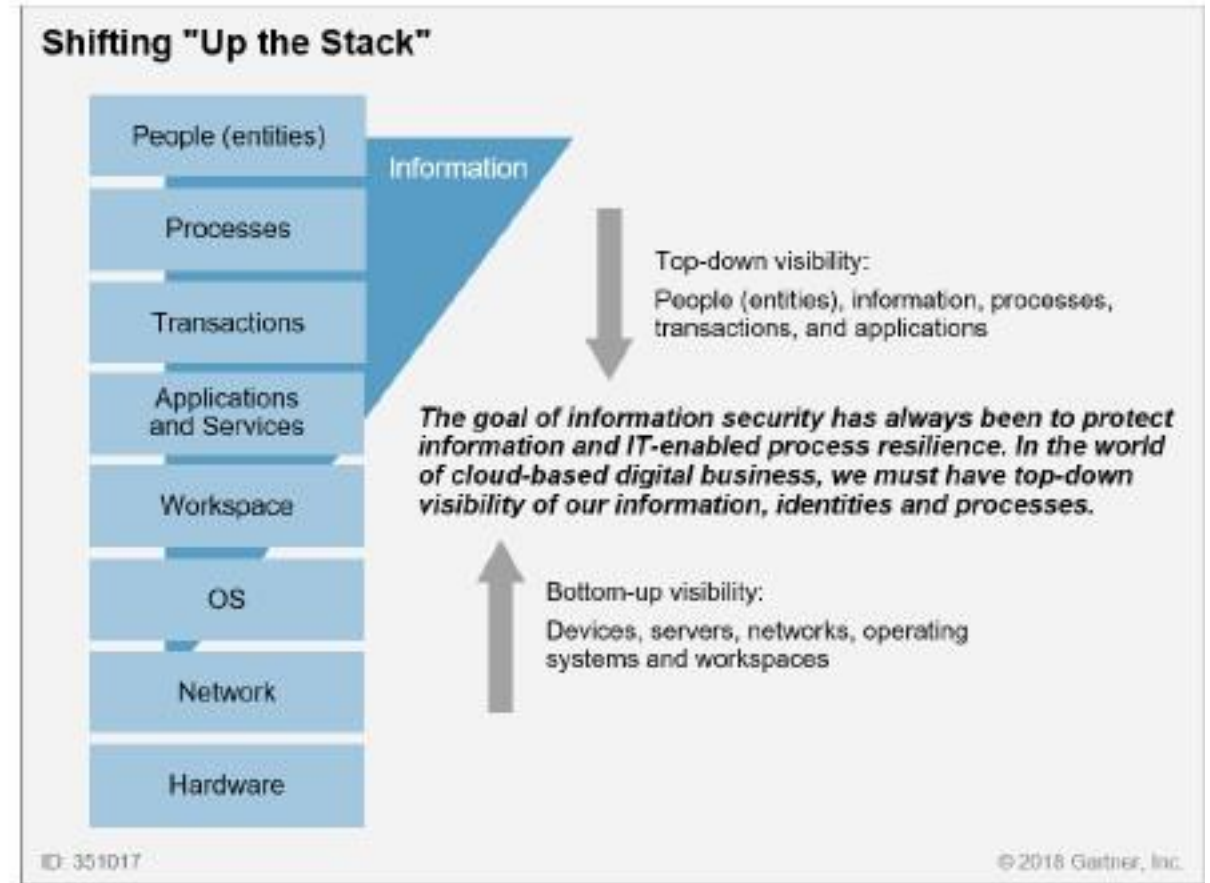
- Unsung Hero.
- Understaffed, jack of all trades, underfunded
- Increasing complex IT infrastructure:
 - Moving to the cloud
 - Virtualization
 - Bring your own device
- Prevention controls fail to block everything
- Hundreds of systems and vulnerabilities to patch



Business Drivers

1. **Predict & Prevent** costly data breaches, security incidents, and disruptions to IT Services.
2. **Reduce costs and increase efficiency** in your cyber security operation
3. Extend **detection and response** capabilities with context correlated from across your endpoint, network, and cloud assets.
4. **Maximize** your existing **investment**

Shifting "Up the Stack" to Identities, Data and Transactions



Source: Gartner (April 2018)

Dwell Time



Dwell time is calculated as the number of days an attacker is present in a victim network before they are detected. The median represents a value at the midpoint of a data set sorted by magnitude.



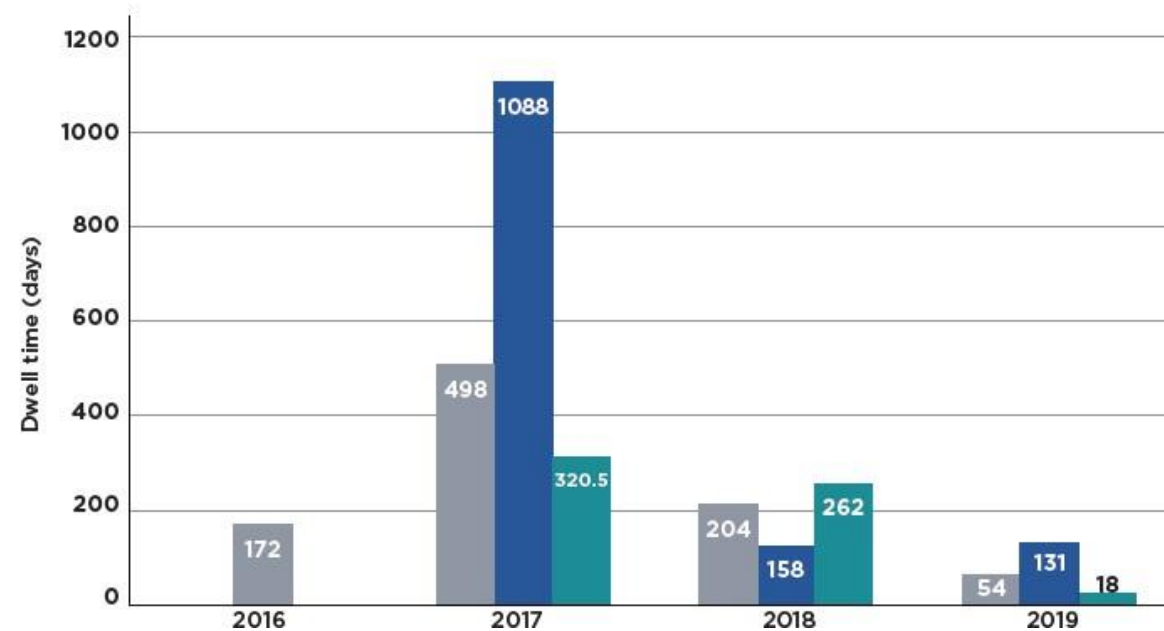
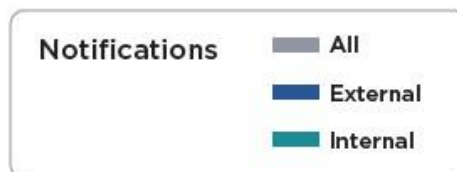
Median Dwell Time

204 > 54

DAYS IN 2018

DAYS IN 2019

APAC MEDIAN DWELL TIME



Mandiant M-Trend Report 2020



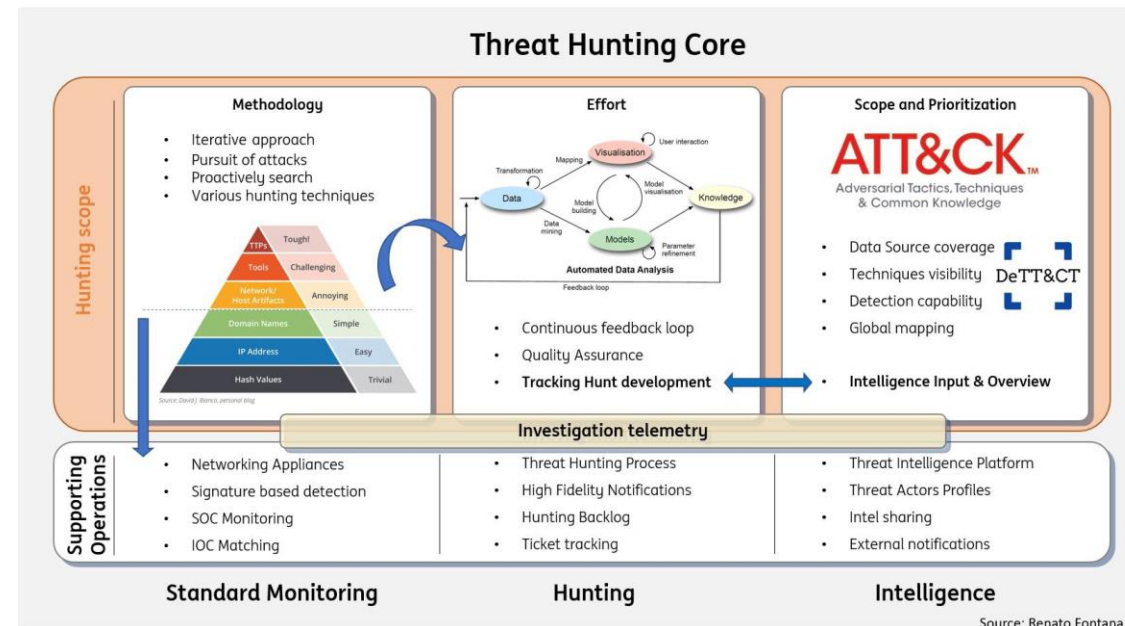
Problems

- Both Endpoint and Networks always have a certain level of vulnerability
- Organizations are struggling to prevent adversaries from getting into their networks.
- Advanced adversaries can remain hidden for months, sometimes years, before detection.

Without knowing the current state of compromise, we have an incomplete picture of Our Cyber Security Posture.

Introduction to Threat Hunting

- Threat hunting is a Proactive cyber defense approach. Threat hunting processes perform proactive and iterative discovery through networks, endpoints, and other infrastructure to detect and respond to cybersecurity threats that sometimes evade existing security solutions.



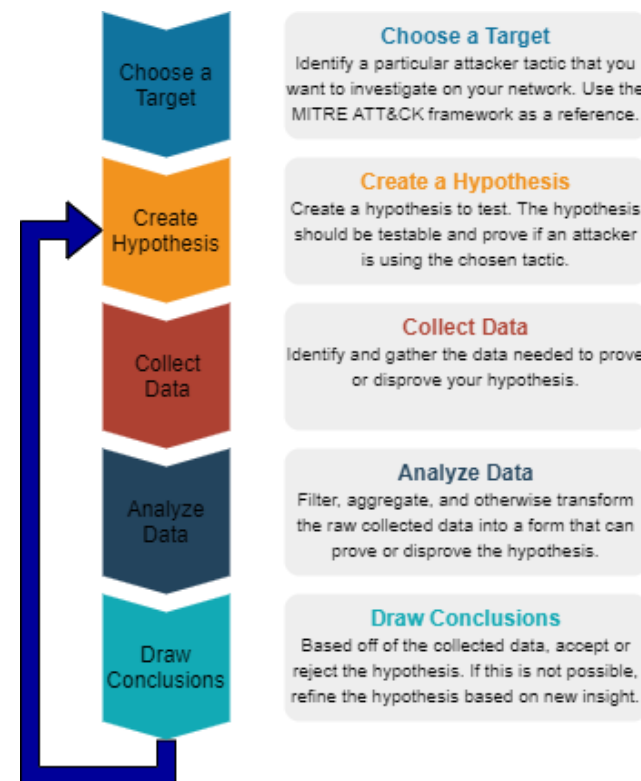
<https://twitter.com/Rcfontana/status/1262407505776381952>

Introduction to Threat Hunting

- Threat hunting is an proactive cyber defense activity. It is "the process of proactively and iteratively searching through networks to detect and isolate advanced threats that evade existing security solutions."
- This is in different to traditional threat management measures, such as firewalls, intrusion detection systems (IDS), malware sandbox (computer security) and SIEM systems, which typically involve an investigation of evidence-based data after there has been a warning of a potential threat.

Threat Hunting Principle

- Presumptions of Compromise : Your prevention technology will eventually fall or have already failed without your knowledge. With Adoption Assume breach mentality will increase your awareness of compromised assets



<https://www.clearnetwork.com/cyber-threat-hunting-what-why-and-how/>



Threat Hunting Benefit

- Finding adversaries who have gotten past your current security protection
- Continuous improvement of your detection capabilities
- With your existing technology, you can not have oversight of everything that's happening, at this point threat hunting help your organization
- Supports faster and early detection of potential compromise
- Increasing awareness of your environment and attack surface
- One of method to improve your data collection



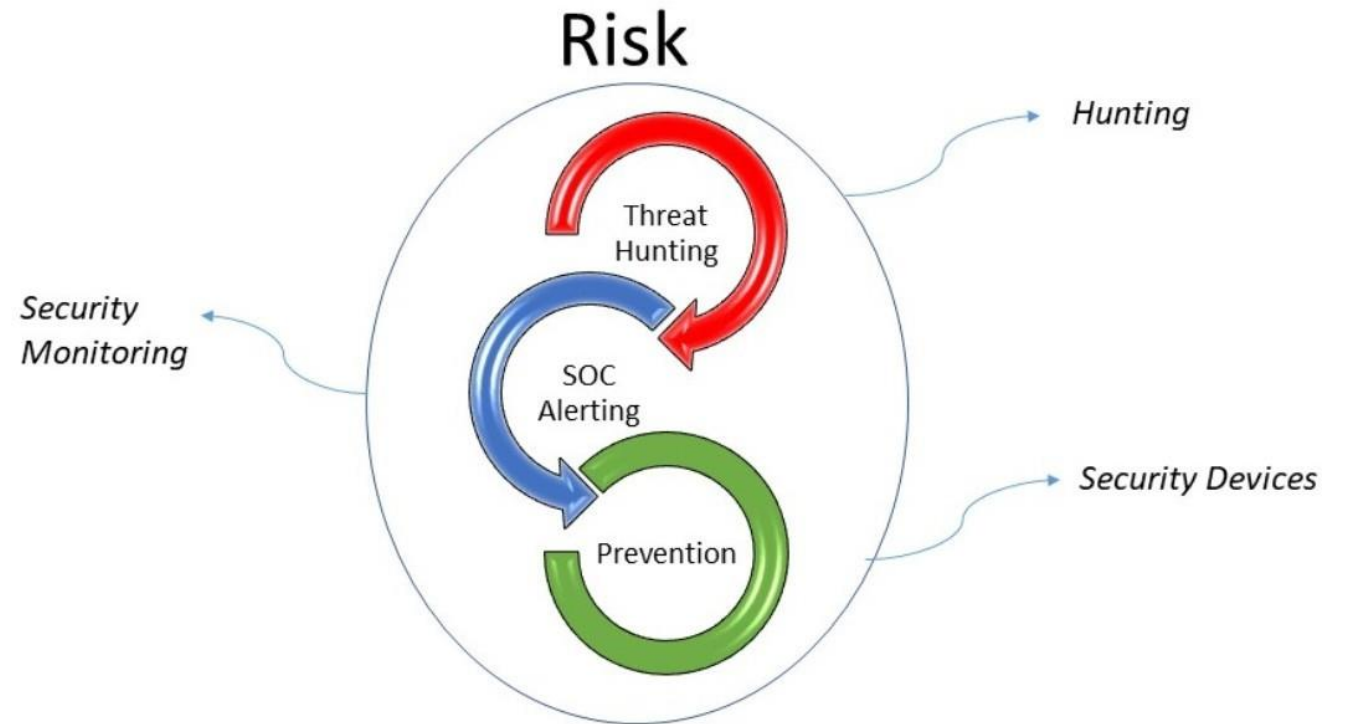
What is it for?

BUSINESS :

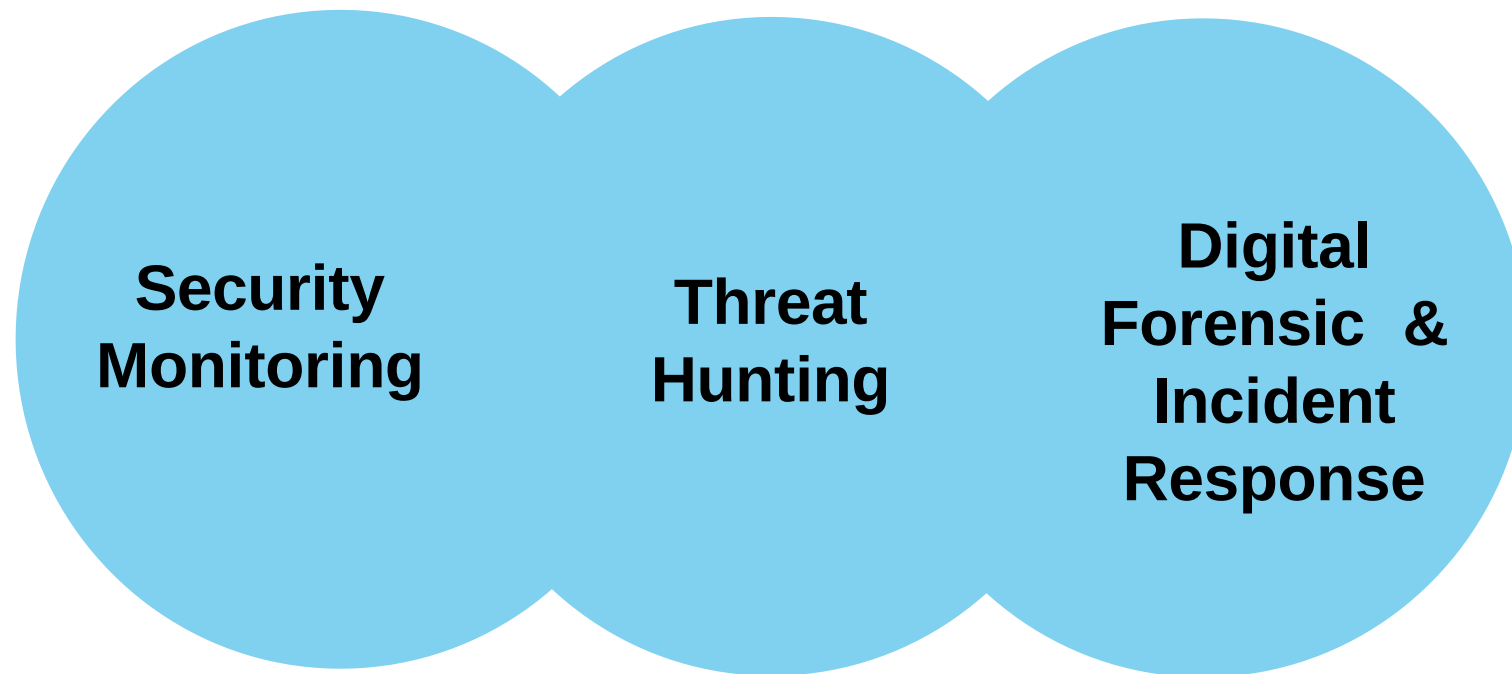
- Minimize residual risks
- Minimize the dwell time (time between attack and detection)

TECHNICAL :

- Advanced [targeted] attacks detection
- Non-malware attacks detection
- TTPs based detection



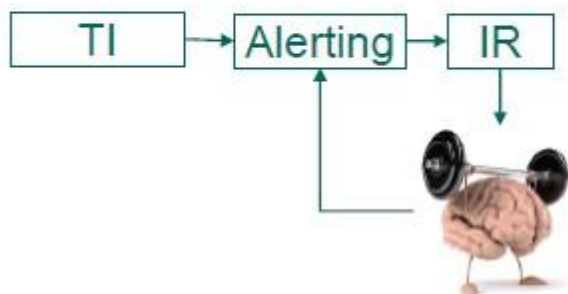
Sec Mon vs Threat Hunting vs DFIR



Threat Hunting Vs Alert Based Investigation

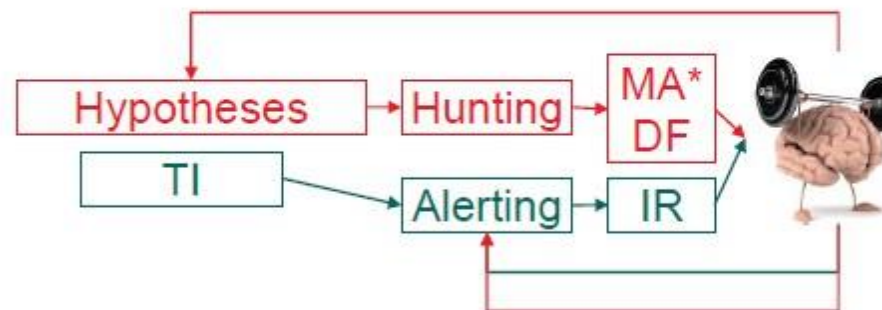
SOC/Alerting

- Reactive
- Detect/forget



Hunting/Mining

- Proactive
- Repeated searches



* MA – malware analysis, DF – digital forensics, IR – incident response

Source : https://2016.zeronights.ru/wp-content/uploads/2016/12/ZN16-KHS-Th_Soldatov.pdf



Threat Hunting vs Compromise Assessment

- What is the Main Differences Between Threat Hunting and Compromise Assessment?
- Basically Threat Hunting and Compromise Assessment is a same activity, but the main difference are :
 - ✓ Situation & Condition : TH -> Assuming Compromise will happen ; and CA -> Compromise is Already happened
 - ✓ Location & Object : TH -> All Object Within Organization ; CA -> Selected Network Segment / Zone Suspected for Compromised Area
 - ✓ Actor (Who performed the activities?) : TH -> Empowered SOC Team (part of SOC Team) ; CA -> Mostly from DFIR Team

Hunting VS Reactive Response

Hunting Organization

- Actively looking for Incidents
 - ✓ Known malware and variant
 - ✓ Patterns of activity : evil vs normal
 - ✓ Threat Intelligence

Reactive Organization

- Incident Starts when notification comes in
 - ✓ Call from government agency
 - ✓ Vendor / threat information
 - ✓ (NIDS, SIEM, Firewall, etc) Alert



Threat Hunting Activity



People - Threat Hunter Skillset (1)

- **Analytical Mindset** : Having a mindset of curiosity, Ability to generate and investigate hypotheses. As an analyst, it's increasingly important to be specific in what questions you're looking to answer during threat hunting.
- **Operating System** : Knowledge of Operating System internals, OS security mechanisms, knowledge of typical security issues of different operating systems,
- **Network Architecture**: understanding how computer networks work, OSI Layer, knowledge of TCP/IP, knowledge of basic protocols (DNS, DHCP, HTTP, SMTP, FTP, SMB);
- **Attack Methods/TTPs / Attack Life Cycle** : Knowledge of specific attack vectors, understanding how an attacker attempts to penetrate your network, which attack vectors and tools he/she can use on different attack stages;

People - Threat Hunter Skillset (2)

- **Log Analysis** : knowledge of different log sources and event types generated by different sources, the ability to analyze logs for anomalies and pivot between data sources to see the big picture;
- **Network Analysis** : the ability to read and understand packet capture data and determine the malicious nature of network traffic;
- **Cyber Threat Intelligence** : Having a skill and knowledge to leverage threat intelligence for threat hunting purposes, always seek for new information from threat intelligence report,
- **Malware Analysis** : Malware analysis a highly specialized skill that aims to determine the origin and purpose of an identified instance of malware.
- **Tools for Threat Hunting** : Understand how to use security analytics platform (e.g. ELK) and SIEM, how to use packet sniffer, how open PCAP, how to see and export logs in OS, how to collect logs from different source, etc

Process – Threat Hunting

While skilled threat hunters are one of the key for successful Threat Hunting capability, threat hunting process is also very important. Having a formal hunting process is ensured the consistency and efficiency across all hunts process.

Threat Hunting Life Cycle



SQRRL Threat Hunting Loop

<https://medium.com/@sqrrldata/the-hunting-loop-10c7d451dec8>



Process – Threat Hunting (1)

1. Creating a Hypotheses

Threat Hunting begins with questions, such as “How would a threat actor infiltrate our infrastructure?”

These questions then need to be broken down into specific and measurable hypotheses that state :

- **What is my crown jewel asset?**
- **What threats might be present in the network?**
- **How can we identified the threat actors?**

Hypotheses cannot be generated by tools. It is defined by threat hunter mindset and knowledge based on the condition in each of their environment.



Process – Threat Hunting (2)

Example Hypotheses

Threat Actor:

An organisational threat assessment identified Lazarus Group as a high priority threat. Techniques attributed to this threat actor are detailed within MITRE's ATT&CK Navigator.

We therefore hypothesis that if this threat actor is present in our network, we would be able to detect evidence of multiple techniques being deployed, in a manner consistent with their known attack paths.

Tool:

CTI and our situational awareness suggests that our organisation is currently vulnerable to a variant of the WannaCry ransomware, as SMBv1 is still used.

We therefore hypothesis that if our network is infected with WannaCry, we will see an increase in the rate of file renaming.

Technique:

Lateral Movement, via *Exploitation of Remote Services*, can be performed by exploiting vulnerability MS17-10. Specifically, this can be done via the Metasploit framework with a module that uses a Server Message Block (SMB) request of a specific size to attempt compromise.

We therefore hypothesise that we can see evidence of this technique being used by isolating this SMB request in our network logs.

<https://hodigital.blog.gov.uk/wp-content/uploads/sites/161/2020/03/Detecting-the-Unknown-A-Guide-to-Threat-Hunting-v2.0.pdf>



Process – Threat Hunting (3)

2. Investigate via Tools and Technique

Once observations have led to hypotheses being generated, these then need to be tested using all the relevant tools and techniques. The importance of Data sources and detection engineering capability from the organization, determine the result of this process.

Existing tools owned by organization, such as a **SIEM or security analytic platform, EDR, TIP** can be used to query the data, from basic searching to more advanced data science techniques, and also visualization can help threat hunters in identifying anomalies and anomalous patterns

Process – Threat Hunting (4)

3. Uncover New Pattern and TTPs

The objective of testing a hypothesis created by the threat hunters in the first process in threat hunting, is to prove whether the hypotheses is prove or not proven. Even if the hypotheses result is not proven, It does not necessarily mean that no malicious activity is present or the hunters create a wrong hypotheses. It can be the current visibility in the organization is not enough or the tools that used by threat hunters is not good enough to help them to investigate the case. In the future maybe this hypotheses can reveal a new TTPs that might be unknown before. The valid hypotheses then become the iterative process as a baseline.

Process – Threat Hunting (5)

4. Inform and Enrich Analytics

Successful hunting process and then should be automated to make the efficient process for the threat hunters to reduce Threat Hunting team's time and to limit them from continuously repeating the same process. This can be done in many ways, such scheduling a saved search, developing a new analytic within existing tools, or providing feedback to a supervised machine learning algorithm.

Let the security analytic platform repeat the successful hunting process from the previous activity of threat hunting, and the threat hunter then finding a new hypotheses to uncover the malicious process which unidentified before.

Tools and Technology – Threat Hunting

We already discuss about people and process in thereat hunting. Tools and Technology is also in need for threat hunting activities. While skilled people and effective processes are the critical factors for a successful Threat Hunting capability, tooling is of course still required to collect and interrogate data, automate analytics, and work collaboratively.

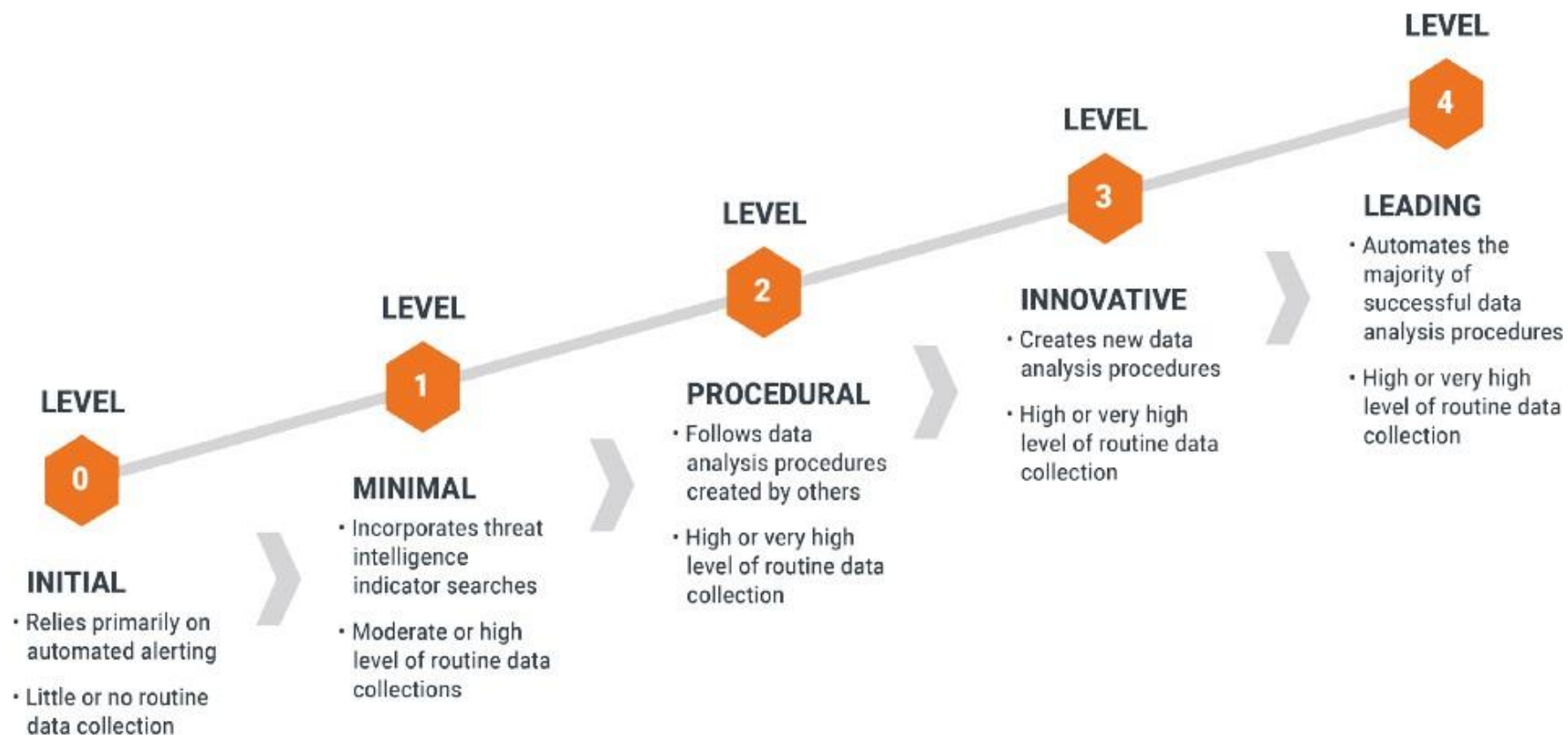
Existing security tools employed by SOC in your organization such as **SIEM, Security Analytic, EDR, Cyber Threat Intelligence Platform, DFIR tools**, can be used and utilized for threat hunting activities. Additional tools such as open source tools might be combined with existing tools to help threat hunters speed up the hunting process and analysis.

Tools and Technology – Threat Hunting

One of the part that also can help for efficiency in threat hunting process is Threat Hunting Playbook. The playbook consist of all hypotheses and step process for hunting created by threat hunters and prevent the threat hunters doing the same hunting process repetitively. The playbook can be also included the sample of dataset from previous hunt activity to help new threat hunters understand what this playbook talking about.

Example of Open Source Threat Hunting Playbook : **Jupyter Notebook and Mordor Datasets** (By Roberto Rodriguez). (<https://medium.com/threat-hunters-forge/threat-hunter-playbook-mordor-datasets-binderhub-open-infrastructure-for-open-8c8aee3d8b4>)

Threat Hunting Maturity Model



SQRRL Hunting Maturity Model

<https://medium.com/@sqrrldata/the-cyber-hunting-maturity-model-6d506faa8ad5>



Threat Hunting Capability Maturity Model

Threat Hunting Capability Maturity Model	Level 1 INITIAL	Level 2 MANAGED	Level 3 DEFINED	Level 4 QUANTITATIVELY MANAGED	Level 5 OPTIMISING
People 	- Existing SOC analysts - Resourcing needs not known - Training needs not known - Performance not managed - Lack of career development plan - Normal systems behaviour not sufficiently understood	- Threat Hunting lead - Informal view of resourcing - Informal view of training - Performance is qualitatively managed - Career development informally managed - Normal systems behaviour is moderately understood	- Dedicated threat hunters - Formal recruitment plan - Formal training plan - Performance expectations defined with role profiles - Formalised career development plan - Normal systems behaviour is fully understood	- SOC analysts rotated for L&D - Succession plans in place - Training completion tracked - Metrics utilised for team performance - Mission critical systems identified	- Teams integrated across SOC - Resourcing needs integrated - Training needs integrated - Improvement plans to address underperformance - Situational awareness
Process 	- Hypothesis generation is unstructured <i>Hunts occur ad-hoc, if at all</i> <i>Little or no data collected</i> - Little understanding of anomalies indicative of malicious activity - Abnormalities not routinely searched for	- CTI and Domain Expertise used to generate hypotheses and prioritisation by lead - Hunts occur occasionally <i>Moderate data collection from key areas</i> <i>Basic threat feeds with IOCs utilised</i> - Targeting of IOCs at bottom of POP	- Formal hunting process - Hunts occur regularly <i>High data collection from key areas</i> <i>CTI and previous experience used to detect malicious activity</i> - Targeting of IOCs in middle of POP	- Manual risk scoring e.g. Crown Jewels - Hunts occur frequently <i>Moderate data collection from most of estate</i> <i>CTI tailored to organisation</i> - Targeting of IOCs at top of POP	- Automated risk scoring e.g. machine learning - Hunts occur continuously <i>High data collection from full estate</i> - Hunt analytics and IOCs shared across community - Automated TTP and campaign tracking
Tools 	<i>Reactive SOC tools</i> - Little or no automation - Little or no documentation produced	- Basic searching via text or SQL-like queries <i>Automatic matching of IOCs</i> - Documentation using basic office suites	- Statistical analysis techniques - Library of hunt procedures automated on regular schedule - Central workflow and knowledge repository tools - Lab environments used to aid hypothesis generation and testing	- Visualisation tools utilised, and analytics tested for effectiveness - Library of hunt procedures automated on frequent schedule - Dashboards utilised	- Machine learning is leveraged, with horizon scanning maintained - Library of hunt procedures automated continuously - Central workflow and knowledge repository are integrated and shared
Note: Items in <i>italics</i> are not strictly part of a Threat Hunting capability, but are essential prerequisites and enablers.					

<https://hodigital.blog.gov.uk/wp-content/uploads/sites/161/2020/03/Detecting-the-Unknown-A-Guide-to-Threat-Hunting-v2.0.pdf>

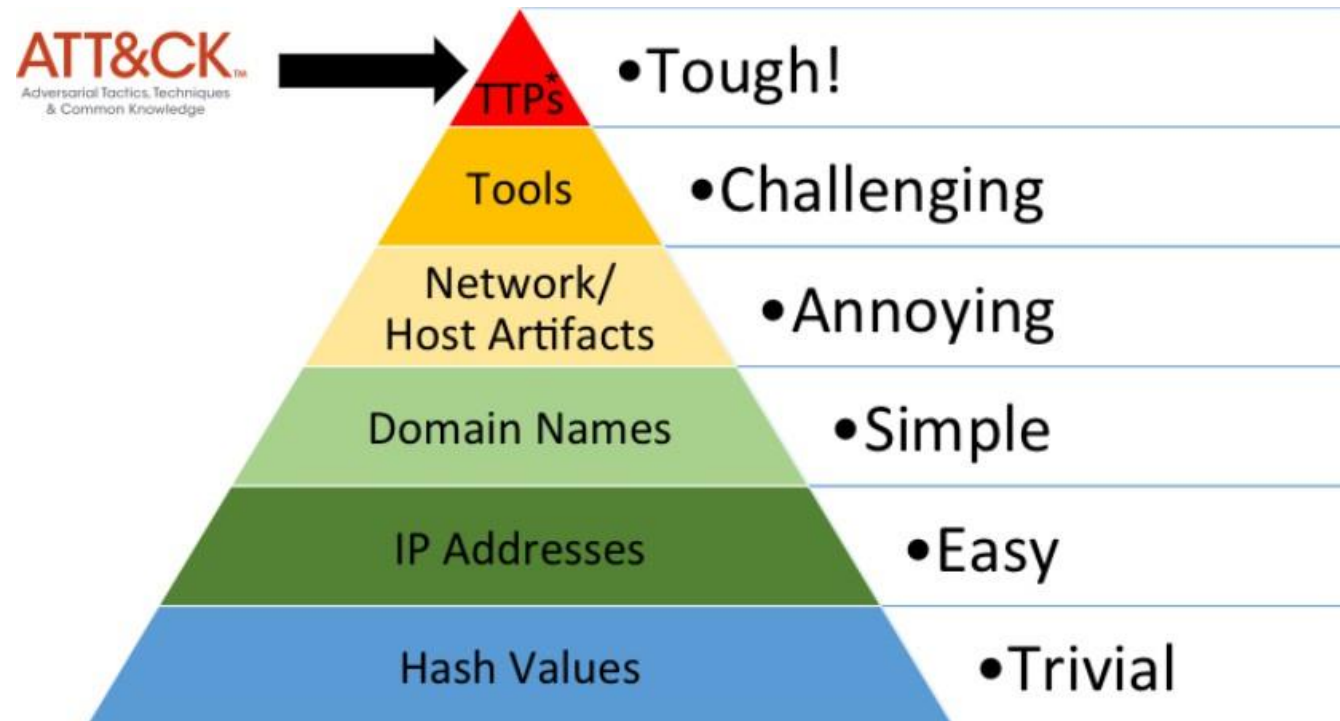
Threat Hunting Framework

Threat Hunting needs a framework that can be a baseline or foundation for the threat hunters when starting they hunting process. The common framework in cyber security used by threat hunting are :

- a. Pyramid of Pain
- b. Cyber Kill Chain
- c. MITRE ATT&CK



Pyramid of Pain

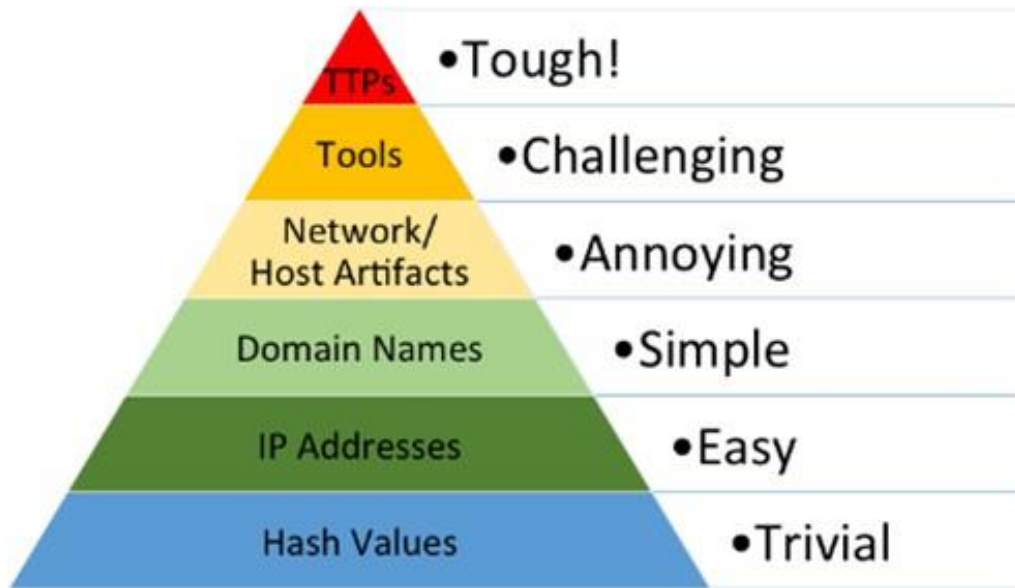


David Bianco Pyramid of Pain

Source : <https://www.slideshare.net/KatieNickels/putting-mitre-attck-into-action-with-what-you-have-where-you-are>



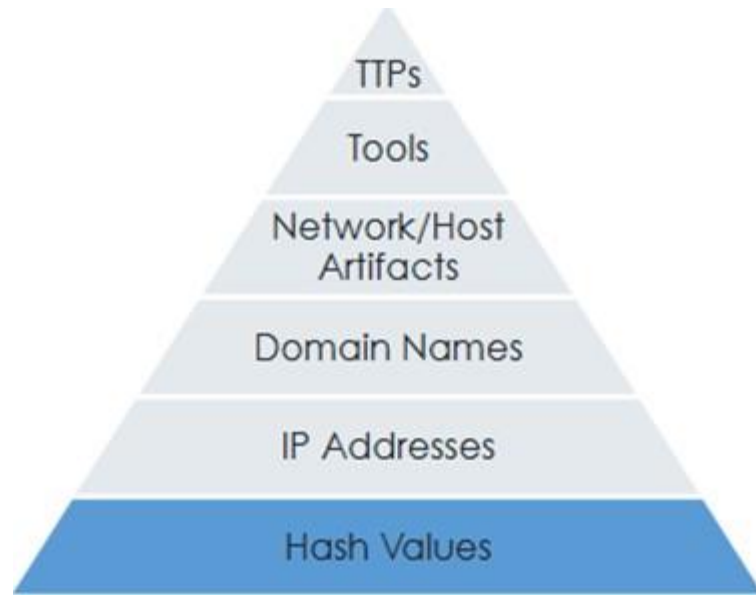
Pyramid of Pain



- Pyramid of pain represents the usefulness of your intelligence
- The higher of the stacks, the more adversaries have to expend for the resources.
- It also indicates to gather the artifacts or threat intelligence from adversaries



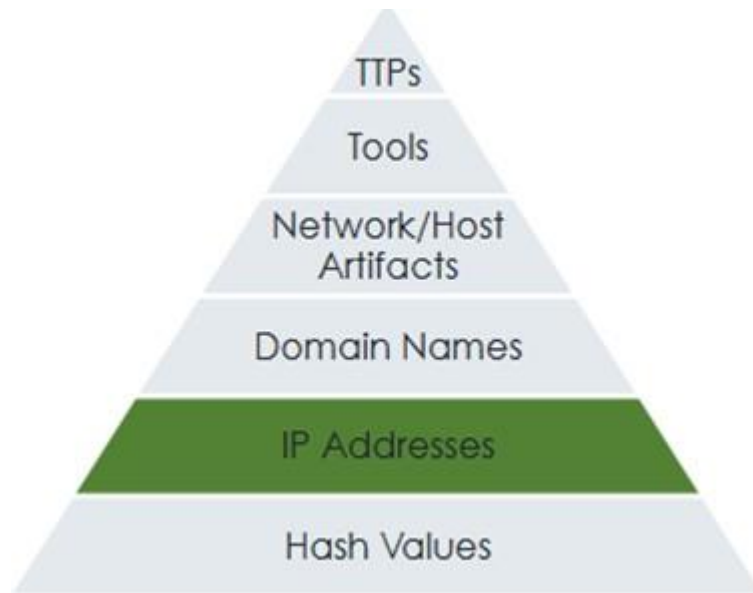
Pyramid of Pain : Hashes



- Hash is so far the **highest confidence level** from artifacts collected or gathered from intel resources
- But, hash is **very easy to change**. Adversaries only need a lil bit effort to modify and create a new hash for their tradecraft
- It is maybe the reason why hash positioned **in the bottom of the pyramid stack**

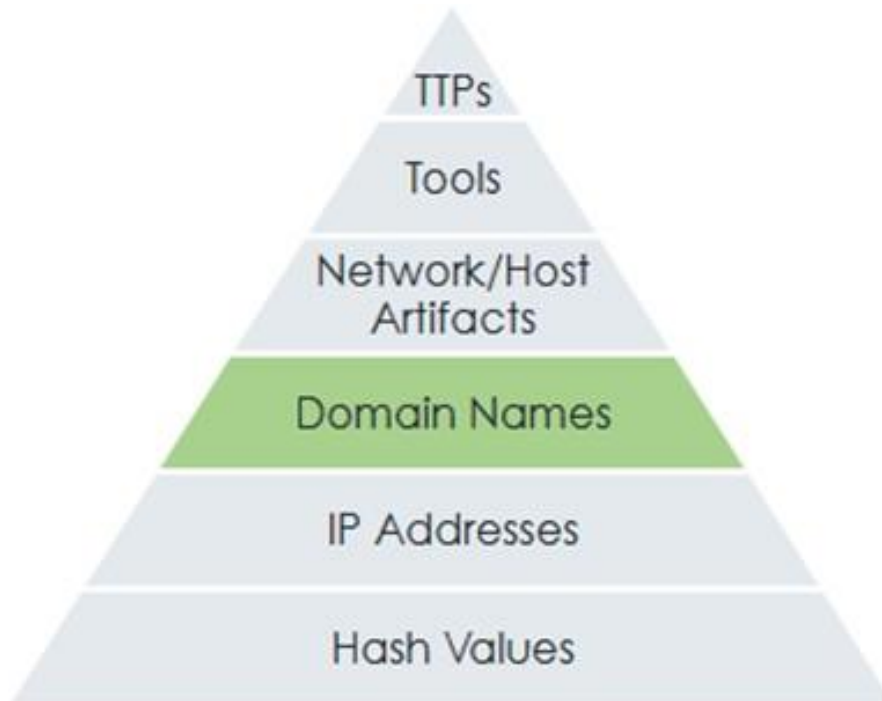


Pyramid of Pain : IP Address



- Attacker mostly not using their real IP Address. Adversary used VPN, Proxy, ToR, Compromised Server to hide from their real IP Address.
- They can change the IP address for their infrastructure once it is blocked / blacklisted. Only need some money and effort to move to the new IP for their infrastructure. More effort and money than hash, therefore IP Address positioned 1 level up from hash in the Pyramid of Pain

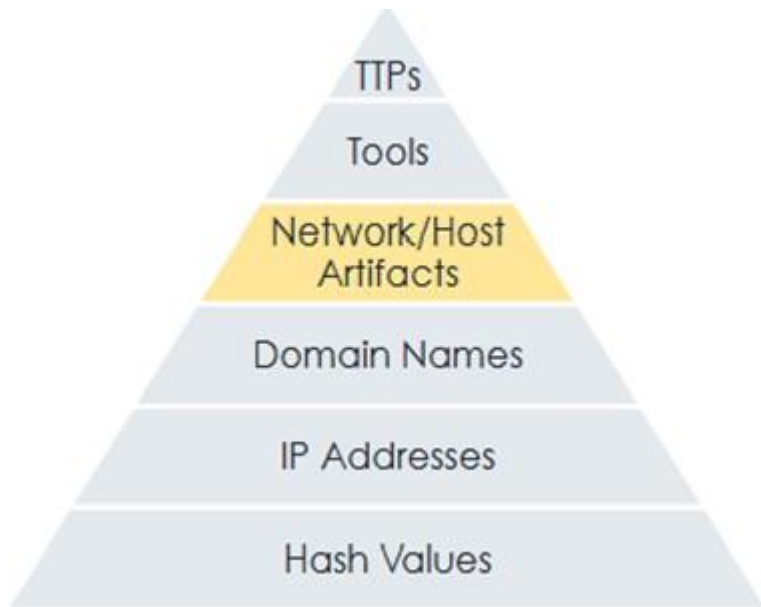
Pyramid of Pain : Domain



- Almost easy as IP Address to change the domain name. But need more time (Domain propagation in DNS)
- Need some registration, and for some reason they mostly hide the whois for domain privacy offered by domain registrars. Need more money for this services.
- Need to define the domain name. And it is not easy. Sometimes adversaries made bot to automatically create a new domain using certain algorithm (DGA)

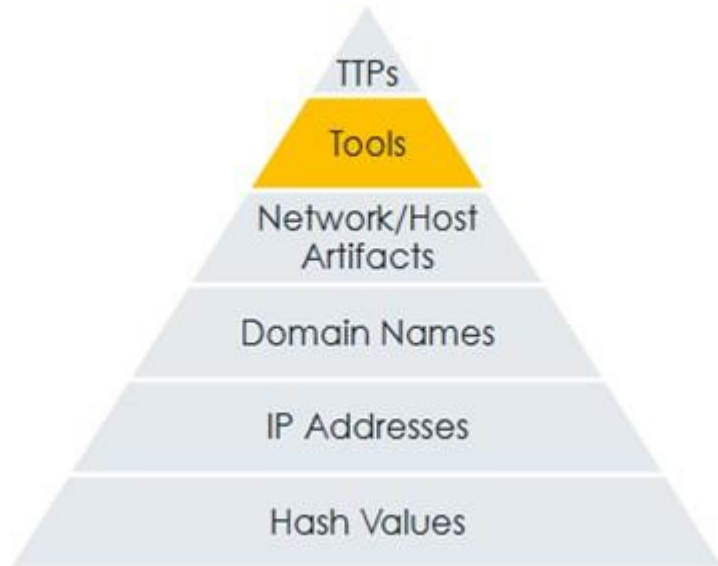


Pyramid of Pain : Network / Host Artifact



- Network Artifacts : indicators of activities performed by the adversaries on the network. Anything communicated over the network by the adversary can be referred to as network artifact, which includes URI patterns, SMTP mailer values, HTTP user agent, and the like.
- Host Artifact : Indicators of activities performed by the adversaries on the hosts. Artifacts like registry keys or values created by malware. Files or directories injected in specific locations, and the like are considered as host artifacts.

Pyramid of Pain : Tools



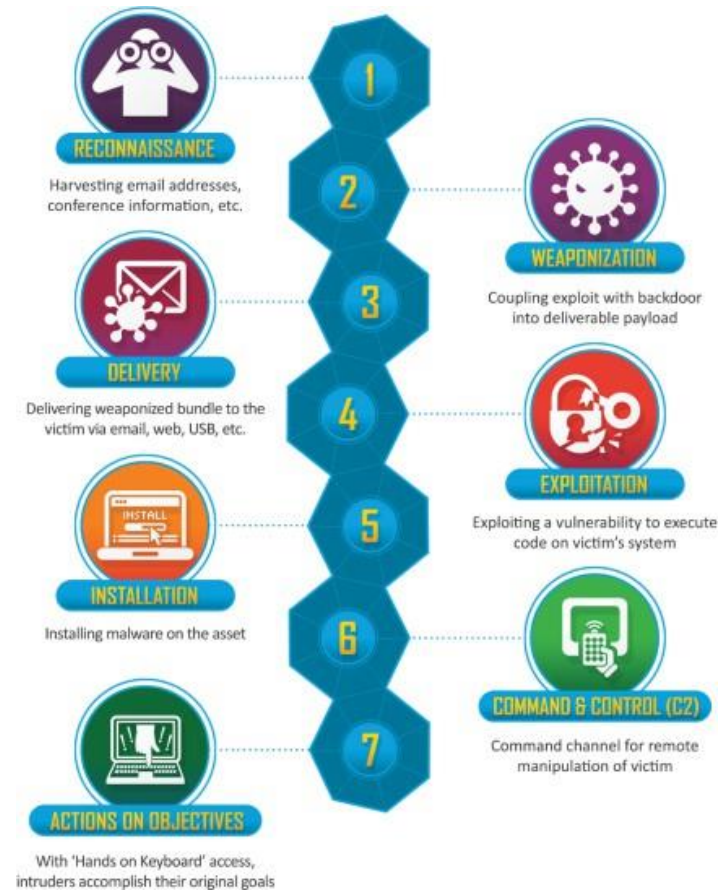
- Software used by the adversary to accomplish their mission
- This can include software designed to create malicious documents for spearphishing, backdoors used to establish CNC, or password cracking tools or other software that adversaries may want to use for post-exploitation activities.
- Considered to be more difficult than all previous stack in pyramid of pain, because sometimes adversaries **need to create their custom tools and obfuscate it to evade the detection and prevention technology.**

Pyramid of Pain : TTPs



- The very Top Level in Pyramid of Pain, indicate the most painful (especially for blue teamers and defenders)
- Need to combine all the stack below to define the attacker Tactic, Technique and Procedures + Combining with Threat Intelligence to define attacker motivation and attribution
- If Blue Teamers, Defenders, and Threat Hunters can reach at this point for detection and response of the adversaries activities, the adversaries only have 2 options : **Give Up on their mission or creating their TTPs from the scratch.** (<http://detect-respond.blogspot.com/2013/03/the-pyramid-of-pain.html>)

Cyber Kill Chain



<https://www.lockheedmartin.com/en-us/capabilities/cyber/cyber-kill-chain.html>



Threat Hunting Framework

Threat Hunting Framework Based on MITRE ATT&CK Framework

- <https://attack.mitre.org/>

Tactic {

Initial Access	Execution	Persistence	Privilege Escalation	Defense Evasion	Credential Access	Discovery	Lateral Movement	Collection	Exfiltration	Command and Control
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	BITS Jobs	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	Binary Padding	Brute Force	Browser Bookmark	Distributed Component	Clipboard Data	Data Encrypted	Connection Proxy
Replication Through Removable Media	Control Panel Items	AppInit DLLs	AppInit DLLs	Bye Acc					Data Transfer Size Limits	Custom Command and Control

Technique

Procedure

Drive-by Compromise

A drive-by compromise is when an adversary gains access to a system through a user visiting a website over the normal course of browsing. With this technique, the user's web browser is targeted for exploitation. This can happen in several ways, but there are a few main components:

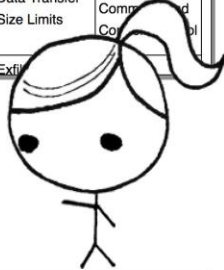
Multiple ways of delivering exploit code to a browser exist, including:

- A legitimate website is compromised where adversaries have injected some form of malicious code such as JavaScript, iFrames, cross-site scripting.
- Malicious ads are paid for and served through legitimate ad providers.
- Built-in web application interfaces are leveraged for the insertion of any other kind of object that can be used to display web content or contain a script that executes on the visiting client (e.g. forum posts, comments, and other user controllable web content).

Often the website used by an adversary is one visited by a specific community, such as government, a particular industry, or region, where the goal is to compromise a specific user or set of users based on a shared interest. This kind of targeted attack is referred to a strategic web compromise or watering hole attack. There are several known examples of this occurring.^[1]

Drive-by Compromise Technique

ID	T1199
Tactic	Initial Access
Platform	Linux, Windows, macOS
Permissions Required	User
Data Sources	Packet capture, Network device logs, Process use of network, Web proxy, Network intrusion detection system, SSL/TLS inspection



Sources : https://threatexpress.com/redteaming/mitre_attack/

MITRE ATT&CK Framework

- MITRE ATT&CK™ is a globally-accessible knowledge base of **adversary tactics and techniques based on real-world observations. The ATT&CK knowledge base is used as a foundation for the development of specific threat models and methodologies** in the private sector, in government, and in the cybersecurity product and service community.

With the creation of ATT&CK, MITRE is fulfilling its mission to solve problems for a safer world — by bringing communities together to develop more effective cybersecurity. ATT&CK is open and available to any person or organization for use at no charge

MITRE ATT&CK Matrix

Reconnaissance 10 techniques	Resource Development 6 techniques	Initial Access 9 techniques	Execution 10 techniques	Persistence 18 techniques	Privilege Escalation 12 techniques	Defense Evasion 37 techniques	Credential Access 14 techniques	Discovery 25 techniques	Lateral Movement 9 techniques	Collection 17 techniques	Command and Control 16 techniques	Exfiltration 9 techniques	Impact 13 techniques
Active Scanning (2)	Acquire Infrastructure (4)	Drive-by Compromise	Command and Scripting Interpreter (3)	Account Manipulation (4)	Abuse Elevation Control Mechanism (4)	Abuse Elevation Control Mechanism (4)	Brute Force (4)	Account Discovery (4)	Exploitation of Remote Services	Archive Collected Data (2)	Application Layer Protocol (4)	Automated Exfiltration (1)	Account Access Removal
Gather Victim Host Information (4)	Compromise Accounts (2)	Exploit Public-Facing Application	Exploitation for Client Execution	BITS Jobs	Access Token Manipulation (2)	Access Token Manipulation (2)	Credentials from Password Stores (2)	Application Window Discovery	Internal Spearphishing	Audio Capture	Communication Through Removable Media	Data Transfer Size Limits	Data Destruction
Gather Victim Identity Information (2)	Compromise Infrastructure (3)	External Remote Services	Inter-Process Communication (2)	Boot or Logon Autostart Execution (12)	Boot or Logon Autostart Execution (12)	BITS Jobs	Exploitation for Credential Access	Browser Bookmark Discovery	Automated Collection	Automated Collection	Clipboard Data	Data Encoded (2)	Data Encrypted for Impact
Gather Victim Network Information (4)	Develop Capabilities (4)	Hardware Additions	Native API	Boot or Logon Initialization Scripts (3)	Boot or Logon Initialization Scripts (3)	Deobfuscate/Decode Files or Information	Forced Authentication	Cloud Infrastructure Discovery	Remote Service Session Hijacking (2)	Clipboard Data	Data from Cloud Storage Object	Data Obfuscation (2)	Data Manipulation (2)
Gather Victim Org Information (4)	Establish Accounts (2)	Phishing (2)	Scheduled Task/Job (4)	Browser Extensions	Boot or Logon Initialization Scripts (3)	Direct Volume Access	Input Capture (4)	Cloud Service Dashboard	Remote Services (4)	Data from Configuration Repository (2)	Data from Dynamic Resolution (2)	Exfiltration Over C2 Channel	Defacement (2)
Phishing for Information (2)	Obtain Capabilities (4)	Replication Through Removable Media	Shared Modules	Compromise Client Software Binary	Create or Modify System Process (4)	Execution Guardrails (1)	Man-in-the-Middle (2)	Domain Trust Discovery	Replication Through Removable Media	Data from Information Repositories (2)	Encrypted Channel (2)	Exfiltration Over Other Network Medium (1)	Disk Wipe (2)
Search Closed Sources (2)		Supply Chain Compromise (2)	Software Deployment Tools	Event Triggered Execution (13)	Event Triggered Execution (13)	Exploitation for Defense Evasion	Modify Authentication Process (4)	File and Directory Discovery	Software Deployment Tools	Fallback Channels	Ingress Tool Transfer	Exfiltration Over Physical Medium (1)	Endpoint Denial of Service (4)
Search Open Technical Databases (3)		Trusted Relationship	System Services (2)	Create Account (2)	Create Account (2)	File and Directory Permissions Modification (2)	Network Sniffing	Network Service Scanning	Taint Shared Content	Data from Local System	Multi-Stage Channels	Exfiltration Over Web Service (2)	Firmware Corruption
Search Open Websites/Domains (2)		Valid Accounts (4)	User Execution (2)	Create or Modify System Process (4)	Exploitation for Privilege Escalation	Group Policy Modification	OS Credential Dumping (2)	Network Share Discovery	Use Alternate Authentication Material (4)	Data from Network Shared Drive	Non-Application Layer Protocol	Scheduled Transfer	Inhibit System Recovery
Search Victim-Owned Websites			Windows Management Instrumentation	Event Triggered Execution (13)	Group Policy Modification	Hide Artifacts (7)	Steal Application Access Token	Network Sniffing		Data from Removable Media	Non-Standard Port	Transfer Data to Cloud Account	Resource Hijacking
				External Remote Services	Hijack Execution Flow (11)	Hijack Execution Flow (11)	Steal or Forge Kerberos Tickets (4)	Peripheral Device Discovery		Data Staged (2)	Protocol Tunneling		Service Stop
				Hijack Execution Flow (11)	Process Injection (11)	Indicator Removal on Host (3)	Steal Web Session Cookie	Process Discovery		Email Collection (2)	Proxy (4)		System Shutdown/Reboot
				Implant Container Image	Scheduled Task/Job (4)	Indirect Command Execution	Two-Factor Authentication Interception	Query Registry		Input Capture (4)	Remote Access Software		
				Office Application Startup (4)	Valid Accounts (4)	Masquerading (3)	Unsecured Credentials (4)	Remote System Discovery		Man in the Browser	Traffic Signaling (1)		
				Pre-OS Boot (3)		Modify Authentication Process (4)		System Information Discovery		Man-in-the-Middle (2)	Web Service (2)		
				Scheduled Task/Job (4)		Modify Cloud Compute Infrastructure (4)		System Network Configuration Discovery		Screen Capture			
				Server Software Component (2)		Modify Registry		System Owner/User Discovery		Video Capture			
				Traffic Signaling (1)		Modify System Image (2)		System Service Discovery					
				Valid Accounts (4)		Network Boundary Bridging (1)		System Time Discovery					
						Obfuscated Files or Information (3)		Virtualization/Sandbox Evasion (2)					
						Pre-OS Boot (3)		Weaken Encryption (2)					
						Process Injection (11)		XSL Script Processing					
						Rogue Domain Controller							
						Rootkit							
						Signed Binary Proxy Execution (11)							
						Signed Script Proxy Execution (1)							
						Subvert Trust Controls (4)							
						Template Injection							
						Traffic Signaling (1)							
						Trusted Developer Utilities Proxy Execution (1)							
						Unused/Unsupported Cloud Regions							
						Use Alternate Authentication Material (4)							
						Valid Accounts (4)							
						Virtualization/Sandbox Evasion (2)							
						Weaken Encryption (2)							
						XSL Script Processing							

Sources : <https://attack.mitre.org/matrices/enterprise/>

MITRE ATT&CK Matrix

How to Read It?

- ❖ **Tactics** across the top
- ✓ What technique accomplish

Reconnaissance	Resource Development	Initial Access	Execution
10 techniques	6 techniques	9 techniques	10 techniques
Active Scanning (2)	Acquire Infrastructure (6)	Drive-by Compromise	Command and Scripting Interpreter (8)
Gather Victim Host Information (4)	Compromise Accounts (2)	Exploit Public-Facing Application	Exploitation for Client Execution
Gather Victim Identity Information (3)	Compromise Infrastructure (6)	External Remote Services	Inter-Process Communication (2)
Gather Victim Network Information (6)	Develop Capabilities (4)	Hardware Additions	Native API
Gather Victim Org Information (4)	Establish Accounts (2)	Phishing (3)	Scheduled Task/Job (6)
Phishing for Information (3)	Obtain Capabilities (6)	Replication Through Removable Media	Shared Modules
Search Closed Sources (2)		Supply Chain Compromise (3)	Software Deployment Tools
Search Open Technical Databases (5)		Trusted Relationship	System Services (2)
Search Open Websites/Domains (2)		Valid Accounts (4)	User Execution (2)
Search Victim-Owned Websites			Windows Management Instrumentation



MITRE ATT&CK Matrix

How to Read It?

- ❖ **Technique** for each column
 - ✓ The way adversaries accomplishing the tactics
 - ✓ Same Technique can be in different Tactics

Persistence	Privilege Escalation	Defense Evasion
18 techniques	12 techniques	37 techniques
Account Manipulation (4)	Abuse Elevation Control Mechanism (4)	Abuse Elevation Control Mechanism (4)
BITS Jobs	Access Token Manipulation (5)	Access Token Manipulation (5)
Boot or Logon Autostart Execution (12)	Boot or Logon Autostart Execution (12)	BITS Jobs
Boot or Logon Initialization Scripts (5)	Boot or Logon Initialization Scripts (5)	Deobfuscate/Decode Files or Information
Browser Extensions	Create or Modify System Process (4)	Direct Volume Access
Compromise Client Software Binary	Event Triggered Execution (15)	Execution Guardrails (1)
Create Account (3)		Exploitation for Defense Evasion
		File and Directory Permissions Modification (2)



Tactic Vs Technique

Tactic : The What”	Technique : The How”
Reconnaissance	Active Scanning
Resource Development	Compromise Account
Initial Access	Drive by Compromise
Execution	Command and Scripting Interpreter



MITRE ATT&CK Use Case

- **ATT&CK can help you create a threat-informed defense**
- **Do what you can, with what you have, where you are:**
 - Detection
 - Assessment and Engineering
 - Threat Intelligence
 - Adversary Emulation
 - Threat Hunting
- **Choose a starting point that works for your team**



Detection Engineering

Detection engineering is a set of practices and systems to deliver modern and effective threat detection.

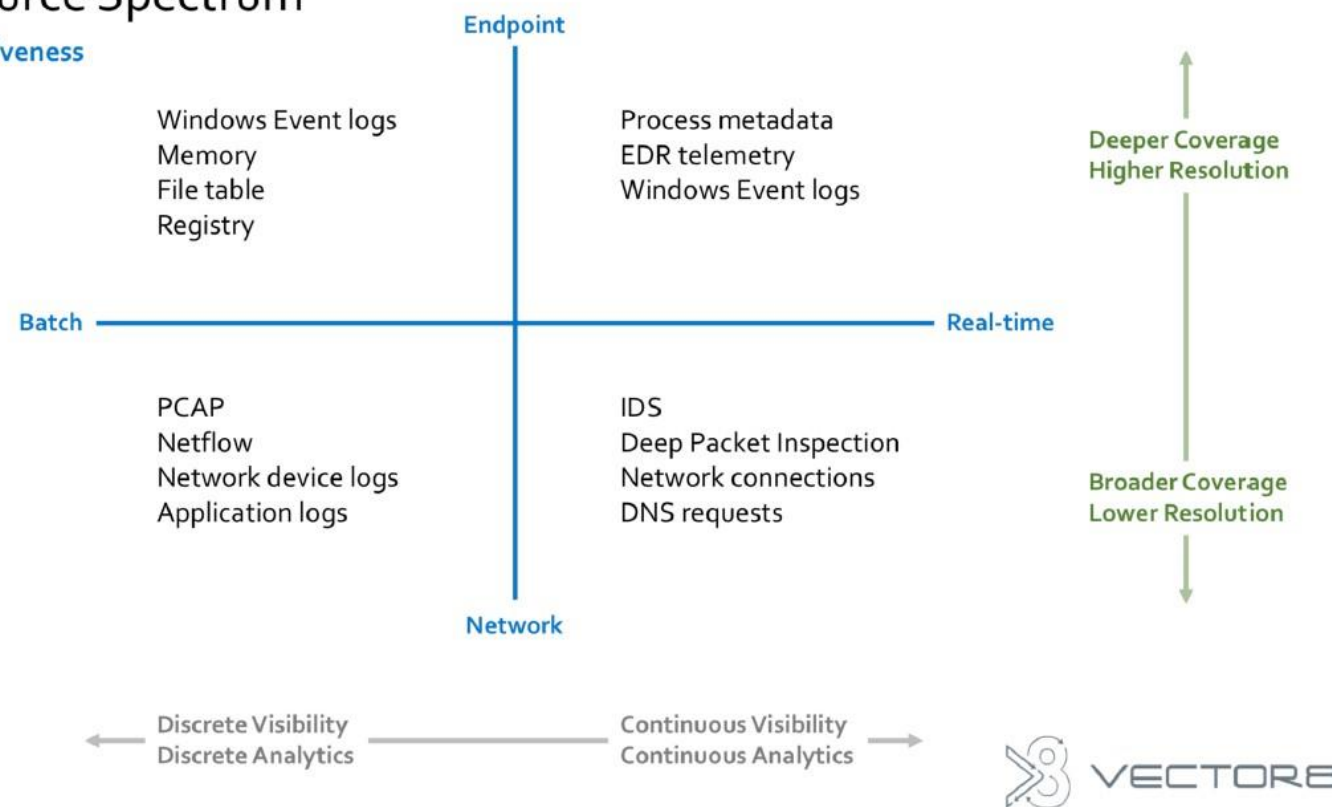
When building a solid detection engineering, the main goal is to catch malicious things and to not catch too many not malicious things. If the detection system interrupt an analyst's activities because calling attention to things that are not malicious, then you're creating more work for the analysts.

Detection products only create value by detecting things that are truly bad, and most detection products lean towards detecting more activity so as to not miss anything.

Detection Engineering

Data Source Spectrum

Visibility vs Liveness



©2018 Vector8, Inc.

Source : Fidelis Cyber Security and Vector8 About Data Source Spectrum

Example of Data Sources from Endpoint

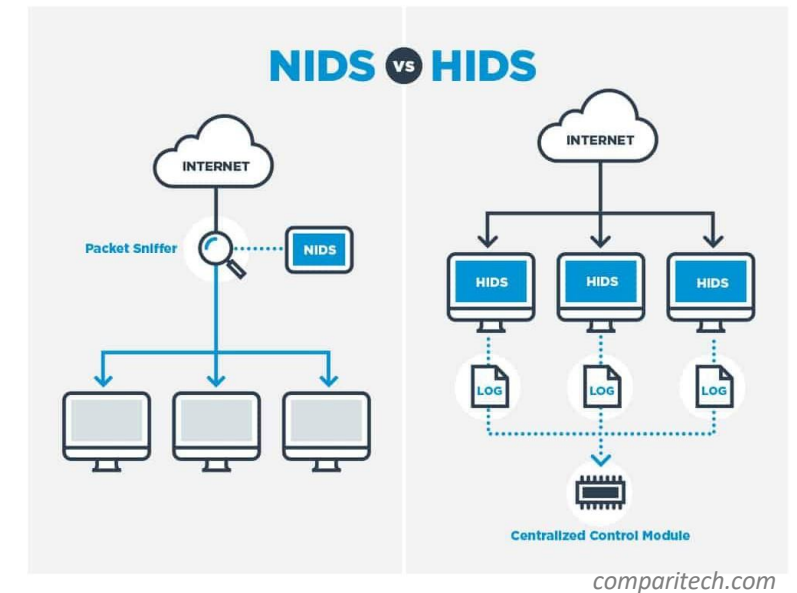
Type of Data	Description	Tools
Operating System logs	Useful Data sources. By Default capabilities for each OS.	Built in Function from OS
Process Activity	Process start, DLL libraries loading, Process install driver, Process perform code injection, Process open port for incoming network connections, connections, Process initiate network connection, Process create/change file, Process create/change registry key/value	Sysmon (Windows) Auditd (Linux) Osquery Endpoint Detection Response Operating System Logs
Volatile Artifacts	Temporary artifact collected from endpoint data sources for the purposes of hunting that might not touch the disk on the host Data Collection : Memory, Network Conn, Process Conn,	Winpmem, Comae, (for Collecton) EDR Volatility Google Rapid Response (GRR) Velociraptor
Non-Volatile Artifacts	Artifacts that resides on the endpoint / host disk. Data Collection : Prefetch, Amcache, Shimcache, MFT, Registry, bash_history, Task Scheduler	Brimorlabs KAPE Kansa FastIR Collector

Example of Data Sources from Network

Type of Event	Description	Tools
Netflow	Network traffic flow metadata. NetFlow data is analyzed to create a picture of network traffic flow and volume. used as a network traffic analyzer to determine its point of origin, destination, volume and paths on the network	Silk Nfsen & Nfdump
Packet Capture	Packet Capture is a networking term for intercepting a data packet that is crossing a specific point in a data network. Once a packet is captured in real-time, it is stored for a period of time so that it can be analyzed, and then either be downloaded, archived or discarded.	Moloch, Tcpdump, Wireshark, tshark
Network IDS	A network-based intrusion detection system (NIDS) detects malicious traffic on a network. NIDS usually require promiscuous network access in order to analyze all traffic	Snort, Suricata Bro Commercial NIDS Product
Proxy Log	Proxy server logs contain the requests made by users and applications on your network. This does not only include the most obvious part : web site request by users but also application or service requests made to the internet (for example application updates).	Squid Commercial Proxy Product
DNS Log	One of the constantly re-occurring techniques is DNS-based activities like exfiltration via DNS (<i>Domain Name System</i>) or C2 (<i>Command and Control</i>) communication via DNS. Still, a lot of companies are lacking in DNS logging, missing DNS-based detection rules, or not aware of their own blindspots. Data collected : DNS Server, DNS Collected from Network, Host Based (Sysmon 10),	Passive DNS Log DNS Server

Intrusion Detection Systems

- HIDS: Host Intrusion Detection Systems
 - Agents run on individual hosts or devices on a network
 - Not what we're talking about today
- NIDS: Network Intrusion Detection Systems
 - Monitor and analyze network traffic for anomalies: suspicious activity, policy violations, etc.
 - Generally passive/out-of-band; otherwise it's an Intrusion Prevention System
 - Detection methods
 - Signature-based detection (e.g., Suricata)
 - Statistical anomaly-based detection (e.g., Random Cut Forest)
 - Stateful protocol analysis detection (e.g., Zeek)



IDS: Types of Attacks

- Scanning Attack
 - Determine network topology
 - IDS highlights connections from one host to many other hosts in the network, or connection attempts to sequential IP addresses and/or ports
- Denial of Service Attack
 - Interrupt service by flooding requests or flaws in protocol implementations
 - IDS identifies large volume of traffic from or to a particular host or invalid connection states (e.g., TCP SYN/ACK with no ACK)
- Penetration Attack
 - Gain access to system resources by exploiting a software or configuration flaw
 - Trickier, but IDS may detect vulnerable software versions or simply alert on unusual operations (e.g., a “write” operation in an already-configured environment with mostly “read” operations)





- Extensible, open-source passive network analysis framework
- More than just an Intrusion Detection System:
 - Packet capture (like  **TCPDUMP**)
 - Traffic inspection (like  Wireshark)
 - Intrusion detection (like **SNORT**)
 - Log recording (like NetFlow and syslog)
 - Scripting framework (like  python™)



Strengths

- Analyzes both link-layer and application-layer behavior
- Content extraction
- Behavioral analysis
- Session correlation
- Can add support for uncommon protocols through scripts/plugins

Weaknesses

- Session metadata only (not full payload)
- Setup and configuration can be complicated
- Produces flat textual log files which can be unwieldy for in-depth analysis

- Network
Protocols
- Files
- Detection
- Network
Observations

[illegible]

FILE	TYPE	DESCRIPTION
0	none	Current directory
1	string	The name of the current window object
location	string	The origin, protocol, host, and file name of the current page
location.href	string	The full URL of the current page
location.protocol	string	The protocol of the current page
location.host	string	The host and port of the current page
location.pathname	string	The path of the current page
location.search	string	The search string of the current page
location.hash	string	The hash of the current page
location.reload()	void	Reloads the current page
location.assign(url)	void	Assigns the current page to the specified URL
location.replace(url)	void	Replaces the current page with the specified URL
location.back()	void	Go back to the previous page
location.forward()	void	Go forward to the next page
location.stop()	void	Stop the current page from loading
location.open(url, name, replace)	void	Open a new window with the specified URL, name, and replace flag
location.close()	void	Close the current window

Network Protocols

- `conn` – Network session tracking
 - Identified by session 4-tuple (originating IP:port, responding IP:port)
 - One session (line in a log file) for every IP connection
 - Unique identifier (UID) ties lines from other logs to a session
- `http`, `modbus`, `ftp`, `dns`, etc.
 - Protocol-specific log files created as traffic is seen
 - Contain application-layer metadata about network activities

Files

- `files` – File analysis results
 - Each transferred file identified with FUID
 - Associated with connection UID(s) over which file was transferred
 - File name, mime type, file size, etc. provided when available
- `pe` – Analysis of Portable Executable (PE) files
 - Target platform, architecture, OS, etc. for executables transferred across the network
- `x509` – Analysis of X.509 public key certificates

Detection

- `notice` – Zeek concept of “alarms,” notices draw extra attention to an event
 - `Conn::Content_Gap`, `DNS::External_Name`,
`FTP::Bruteforcing`, `Heartbleed::SSL_Heartbeat_Attack`,
`HTTP::SQL_Injection_Attacker`, `Scan::Address_Scan`,
`Scan::Port_Scan`, `Software::Vulnerable_Version`,
`SSH::Password_Guessing`, `SSL::Certificate_Expired`,
`Weird::Activity`, ...
 - <https://docs.zeek.org/en/stable/zeek-noticeindex.html>

Detection (cont.)

- `weird` – Unexpected network-level activity
 - > 150 weirdness indicators across many protocols
 - <https://docs.zeeb.org/en/stable/scripts/base/frameworks/notice/weird.zeeb.html#id1>
- `signatures` – Signature matches, including hits from enabled carved file scanners like ClamAV, YARA and capa

Network Observations

- Periodic dump of entities seen over the last day
 - `known_certs` – SSL certificates
 - `known_devices` – MAC addresses
 - `known_hosts` – Hosts with TCP handshakes
 - `known_modbus` – Modbus masters and slaves
 - `known_services` – Services (TCP “servers”)
 - `software` – Software being used on the network (e.g., Apache, OpenSSH, etc.)
 - Could be used for identifying vulnerable versions of software or firmware



Arkime

Strengths

- Large scale index packet capture and search tool
- Packet analysis engine with support for many common IT protocols
- Web interface for browsing, searching, analysis and PCAP carving for exporting
- PCAP payloads (not just session header/metadata) are viewable and searchable

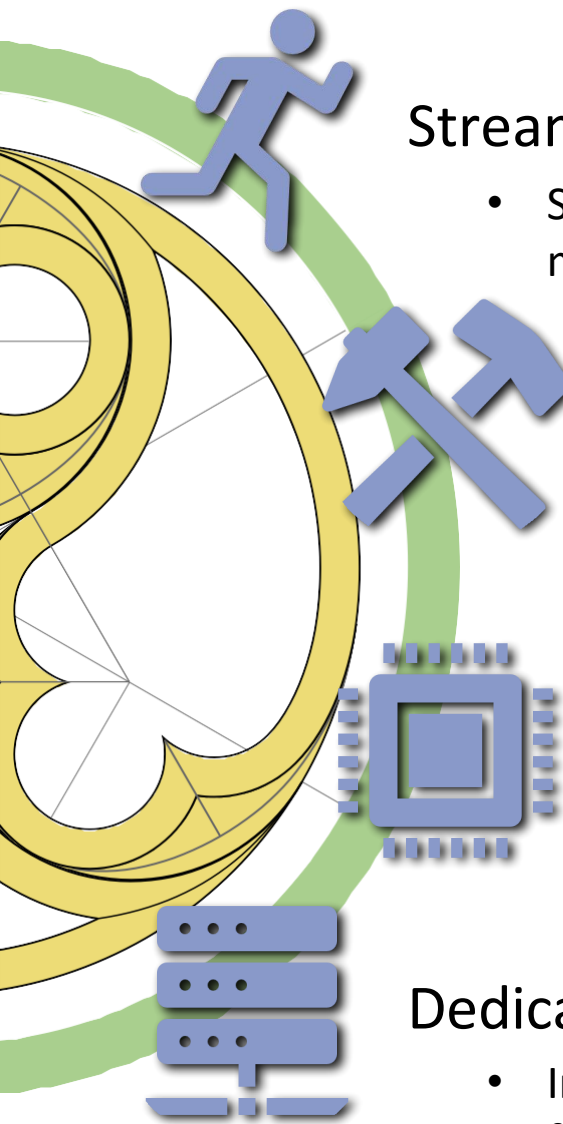
Weaknesses

- No OT protocol support
- Adding new protocol parsers requires C programming

Malcolm

A powerful open-source network traffic analysis tool suite.

<https://github.com/idaholab/Malcolm>



Streamlined deployment

- Suitable for field use (hunt or incident response) or SOC deployment. Runs in Docker on Linux, macOS and Windows platforms. Provides easy-to-use web-based user interfaces.

Industry-standard tools

- Uses Arkime and Zeek for network traffic capture, Logstash for parsing and enrichment, OpenSearch for indexing and Dashboards and Arkime Viewer for visualization. Also leverages OpenSearch Anomaly Detection, Suricata IDS, YARA, capa, ClamAV, CyberChef and other proven tools for analysis of traffic and artifacts.

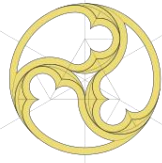
Expanding control systems visibility

- Analyzes more protocols used in operational technology (OT) networks than other open-source or paid solutions. Ongoing development is focused on increasing the quantity and quality of industrial control systems (ICS) traffic.

Dedicated sensor appliance

- Includes Hedgehog Linux, a hardened Linux distribution for capturing network traffic and forwarding its metadata to Malcolm.

Malcolm



Components

<https://github.com/idaholab/Malcolm/#Components>



Capture &
Analysis



File Scanning



Forwarding &
Enrichment



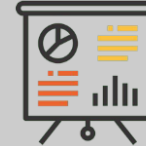
Storage



Anomaly
Detection



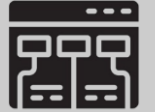
Alerting



Visualization



Payload
Analysis



Framework



beats



OpenSearch



OpenSearch
Anomaly
Detection
Plugin



OpenSearch
Alerting
Plugin



OpenSearch
Dashboards



CyberChef



docker



SURICATA



ClamAV



CAPA



netsniff-ng



TCPDUMP



VIRUSTOTAL



logstash



Arkime

Arkime
session PCAP
export to

WIRESHARK



NGINX



Supported Protocols

<https://github.com/idaholab/Malcolm/#Protocols>

Internet layer

Border Gateway Protocol (BGP)

Building Automation and Control (BACnet)

Bristol Standard Asynchronous Protocol (BSAP)

Distributed Computing Environment / Remote Procedure Calls
(DCE/RPC)

Dynamic Host Configuration Protocol (DHCP)

Distributed Network Protocol 3 (DNP3)

Domain Name System (DNS)

EtherCAT

EtherNet/IP / Common Industrial Protocol (CIP)

FTP (File Transfer Protocol)

GENISYS

Google Quick UDP Internet Connections (gQUIC)

Hypertext Transfer Protocol (HTTP)

IPsec

Internet Relay Chat (IRC)

Lightweight Directory Access Protocol (LDAP)

Kerberos

Modbus

MQ Telemetry Transport (MQTT)

MySQL

NT Lan Manager (NTLM)

Network Time Protocol (NTP)

Oracle

**Open Platform Communications Unified Architecture
(OPC UA) Binary**

Open Shortest Path First (OSPF)

OpenVPN

PostgreSQL

Process Field Net (PROFINET)

Remote Authentication Dial-In User Service (RADIUS)

Remote Desktop Protocol (RDP)

Remote Framebuffer / Virtual Network Computing (RFB/VNC)

S7comm / Connection Oriented Transport Protocol (COTP)

Secure Shell (SSH)

Secure Sockets Layer (SSL) / Transport Layer Security (TLS)

Session Initiation Protocol (SIP)

Server Message Block (SMB) / Common Internet File System (CIFS)

Simple Mail Transfer Protocol (SMTP)

Simple Network Management Protocol (SNMP)

SOCKS

STUN (Session Traversal Utilities for NAT)

Syslog

Tabular Data Stream (TDS)

Telnet / remote shell (rsh) / remote login (rlogin)

TFTP (Trivial File Transfer Protocol)

WireGuard

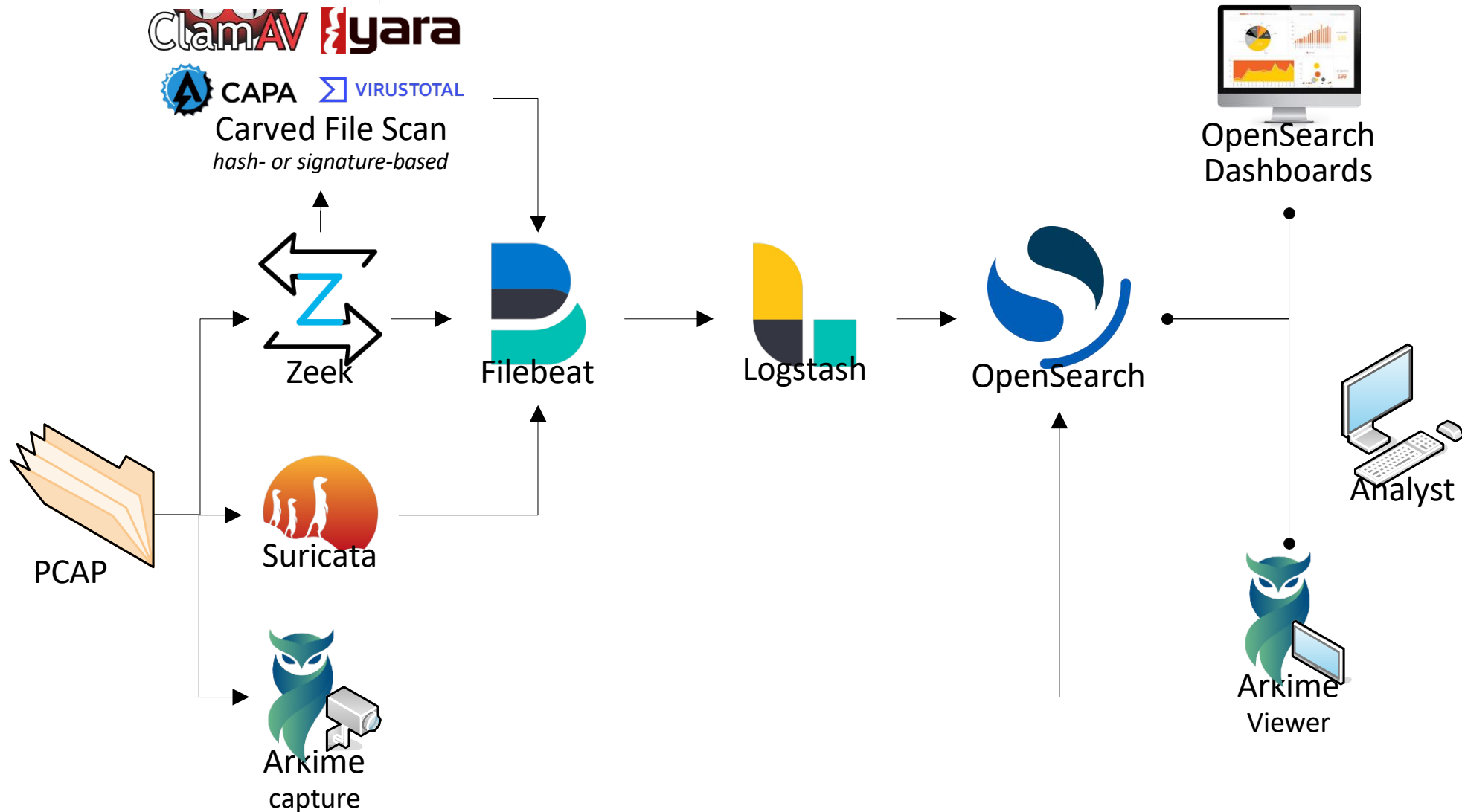
various tunnel protocols (e.g., GTP, GRE, Teredo, AYIYA, IP-in-IP, etc.)

** Industrial control systems protocols indicated with **bold***

Malcolm

Data Pipeline

<https://github.com/idaholab/Malcolm>



Malcolm

Data Pipeline

<https://github.com/idaholab/Malcolm>

Traffic is collected passively by the Hedgehog sensor device

- Zeek, Arkime Capture and Suricata generate metadata about network communications
- Full PCAP may be stored locally on the sensor
- Files transfers are detected and the files scanned for threats
- PCAP may also be uploaded to or captured by Malcolm without requiring a

Metadata is securely forwarded to Malcolm

- All communications between the sensor and aggregator are TLS-encrypted
- Sensor data including resource utilization, syslog, audit logs, temperatures and more may also be forwarded

Logs are enriched and stored in OpenSearch

- Lookups are performed for GeoIP, ASN, MAC-to-vendor, community ID, domain name entropy, etc.
- Network events normalized across protocols and data sources
- Best-guess techniques applied for identifying obscure ICS traffic
- Enriched metadata may be forwarded to higher-

Machine learning algorithms identify anomalies

- Default detectors are provided for action and result, flow size and types of transferred files
- Custom detectors may be created for any aspect of any supported protocol

Alerts are sent over email, webhooks, Slack or Amazon Chime

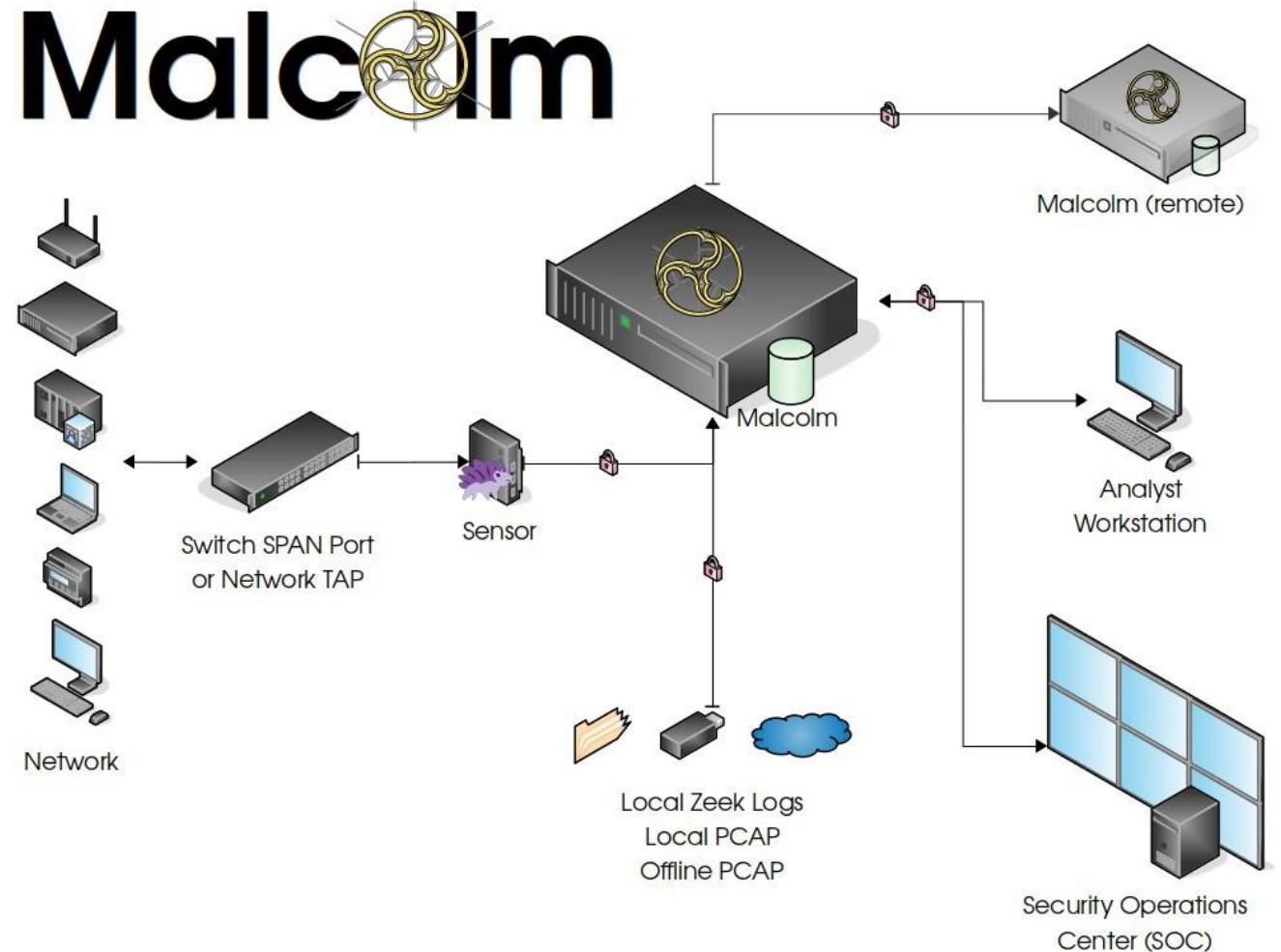
- Alerts may be triggered by exceeded thresholds, anomalies detected, custom queries, etc.

Traffic is visualized in OpenSearch Dashboards and Arkime Viewer

- Dozens of custom dashboards are provided for all supported protocols
- PCAP payloads are retrieved from sensor automatically on demand
- Custom visualizations may be created via drag-and-drop interface
- Malcolm can authenticate users from its own list or via Active Directory /

Configuring and Running Malcolm

- Runs natively in Docker or in a Virtual Machine
- 16+GB RAM, 4+ cores, “enough” disk for PCAP and logs suggested
- Documentation and source code on GitHub:
github.com/idaholab/Malcolm



Identifying Network Hosts and Subnets

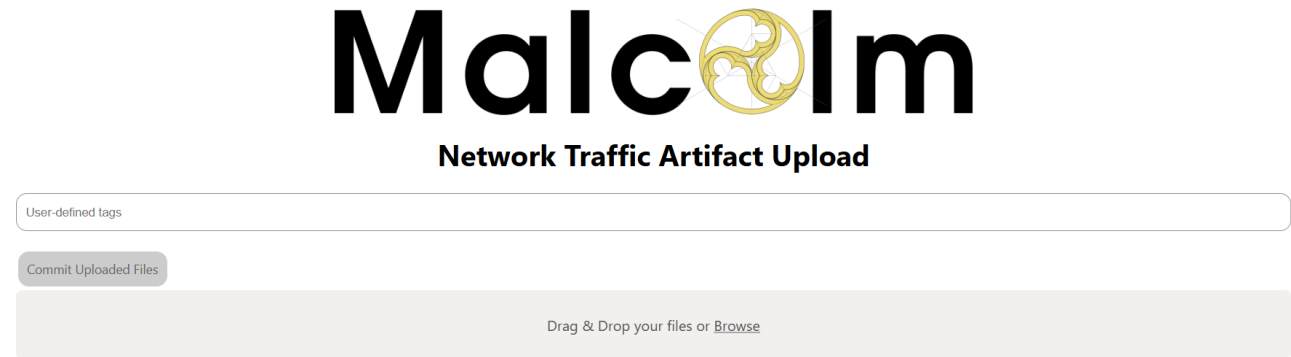
- Assign custom names to network hosts and subnets prior to PCAP import
- Allows identification of cross-segment traffic and name-based search and filter
- Define in text file(s) or via web interface
- <https://localhost/netbox>

	Address	Name	Tag	Search mappings	
	06:46:0b:a6:16:bf	serial-host.intranet.lan	testbed		
ent	10.0.0.0/8	corporate			
	127.0.0.1	localhost			
	127.0.1.1	localhost			
ent	172.16.0.0/12	virtualized	testbed		
	192.168.10.10	office-laptop.intranet.lan			
ent	192.168.40.0/24	corporate			
ent	192.168.50.0/24	corporate			
ent	192.168.100.0/24	control			
ent	192.168.200.0/24	dmz			
	:::1	localhost			
▼	Address	Name	Tag (optional)		



Importing Traffic Captures for Analysis

- Specify tags for search and filter
- Enable Suricata and/or Zeek analysis and file extraction
 - Or configure as global defaults
- Upload PCAP files or archived Zeek logs
 - pcapng not supported yet
- `https://localhost/upload`



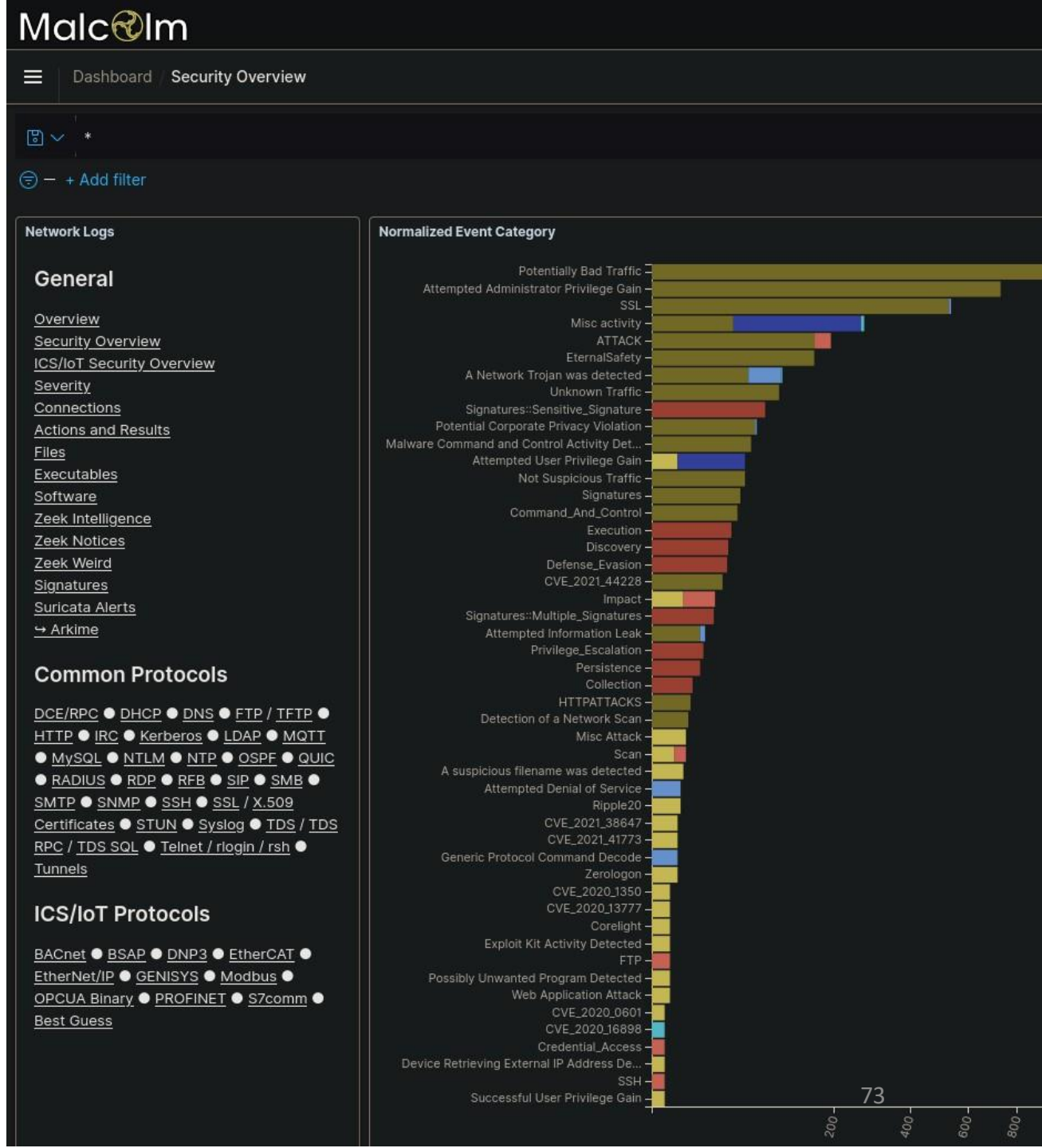
Data Tagging and Enrichment



- Logstash enriches Zeek and Suricata log metadata
 - MAC addresses to hardware vendor
 - GeoIP and ASN lookups
 - Internal/external traffic based on IP ranges
 - Reverse DNS lookups
 - DNS query and hostname entropy analysis
 - Connection fingerprinting (JA4+ for TLS, HASSH for SSH, Community ID for flows)
- `tags` field
 - Populated for Arkime sessions, Zeek logs and Arkime alerts with tags provided on upload and words extracted from PCAP filenames
 - `internal_source`, `internal_destination`, `external_source`, `external_destination`, `cross_segment`

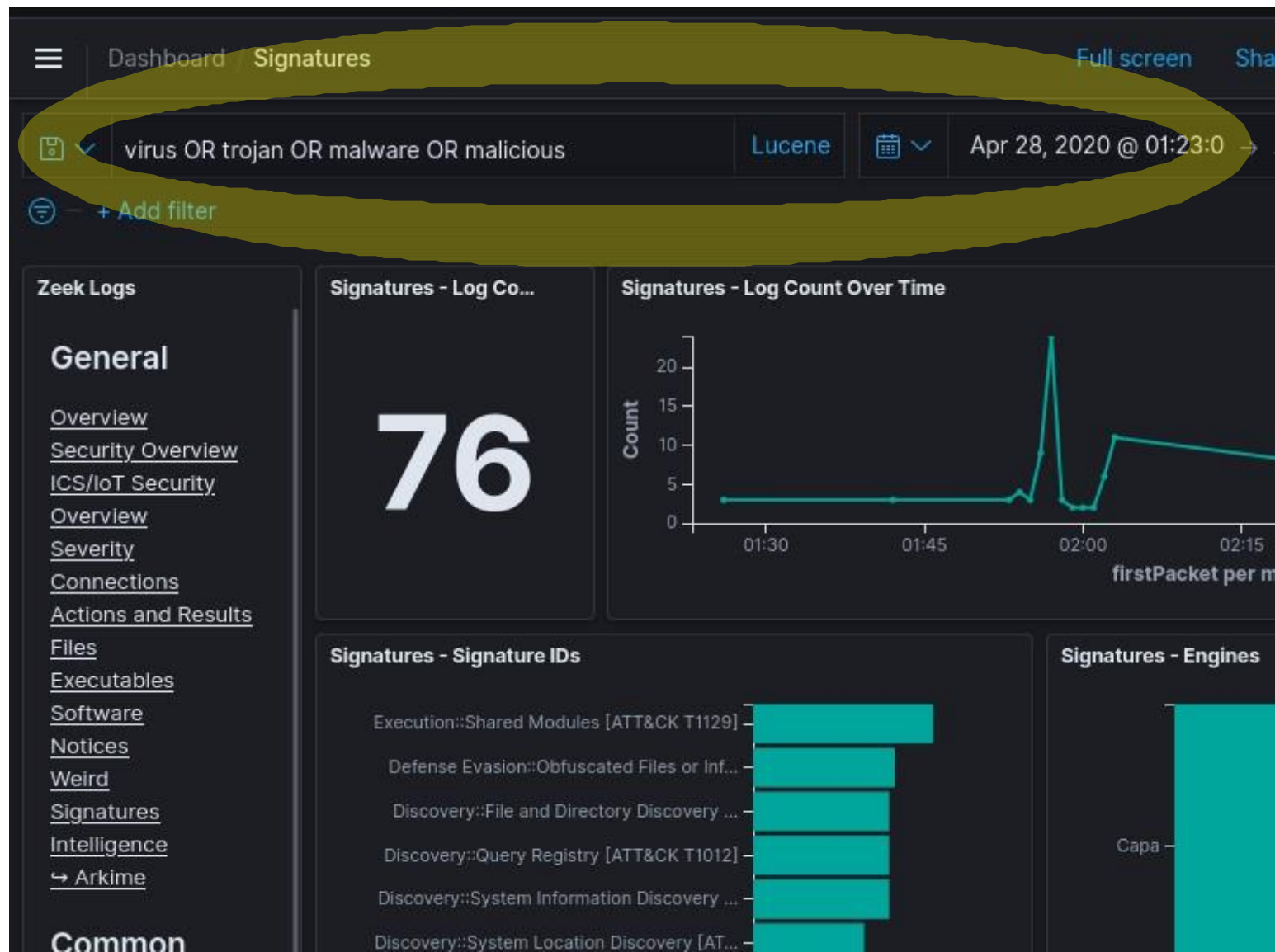


- Front end for Zeek logs and Suricata alerts
- Prebuilt visualizations for all protocols Malcolm parses
- WYSIWYG editors to create custom visualizations and dashboards
- Drill down from high-level trends to specific items of interest
- <https://localhost/dashboards>



Dashboards Filters and Search

- Time filter: define search time frame
- Query bar: write queries in Lucene syntax or DQL (Dashboards Query Language)
- Filter bar: define filters using a UI
 - Pin filters as you move across dashboards
- Save queries and filters for reuse



Overview Dashboards

- High-level view of trends, sessions and events
- Populated from logs across all protocols
- Good jumping-off place for investigation

Network Logs

General

[Overview](#)

[Security Overview](#)

[ICS/IoT Security Overview](#)

[Severity](#)

[Connections](#)

[Actions and Results](#)

[Files](#)

[Executables](#)

[Software](#)

[Zeek Intelligence](#)

[Zeek Notices](#)

[Zeek Weird](#)

[Signatures](#)

[Suricata Alerts](#)

[↔ Arkime](#)

Common Protocols

[DCE/RPC](#) ● [DHCP](#) ● [DNS](#) ● [FTP](#) / [TFTP](#) ●

[HTTP](#) ● [IRC](#) ● [Kerberos](#) ● [LDAP](#) ● [MQTT](#)

● [MySQL](#) ● [NTLM](#) ● [NTP](#) ● [OSPF](#) ● [QUIC](#)

● [RADIUS](#) ● [RDP](#) ● [RFB](#) ● [SIP](#) ● [SMB](#) ●

[SMTP](#) ● [SNMP](#) ● [SSH](#) ● [SSL](#) / [X.509](#)

[Certificates](#) ● [STUN](#) ● [Syslog](#) ● [TDS](#) / [TDS](#)

Normalized Event Categories

Potential
Attempted Administration

A Network T

Signatures:
Potential Corpora
Malware Command and Co
Attempted
No

Com

Signatures:
Attempted

Detection

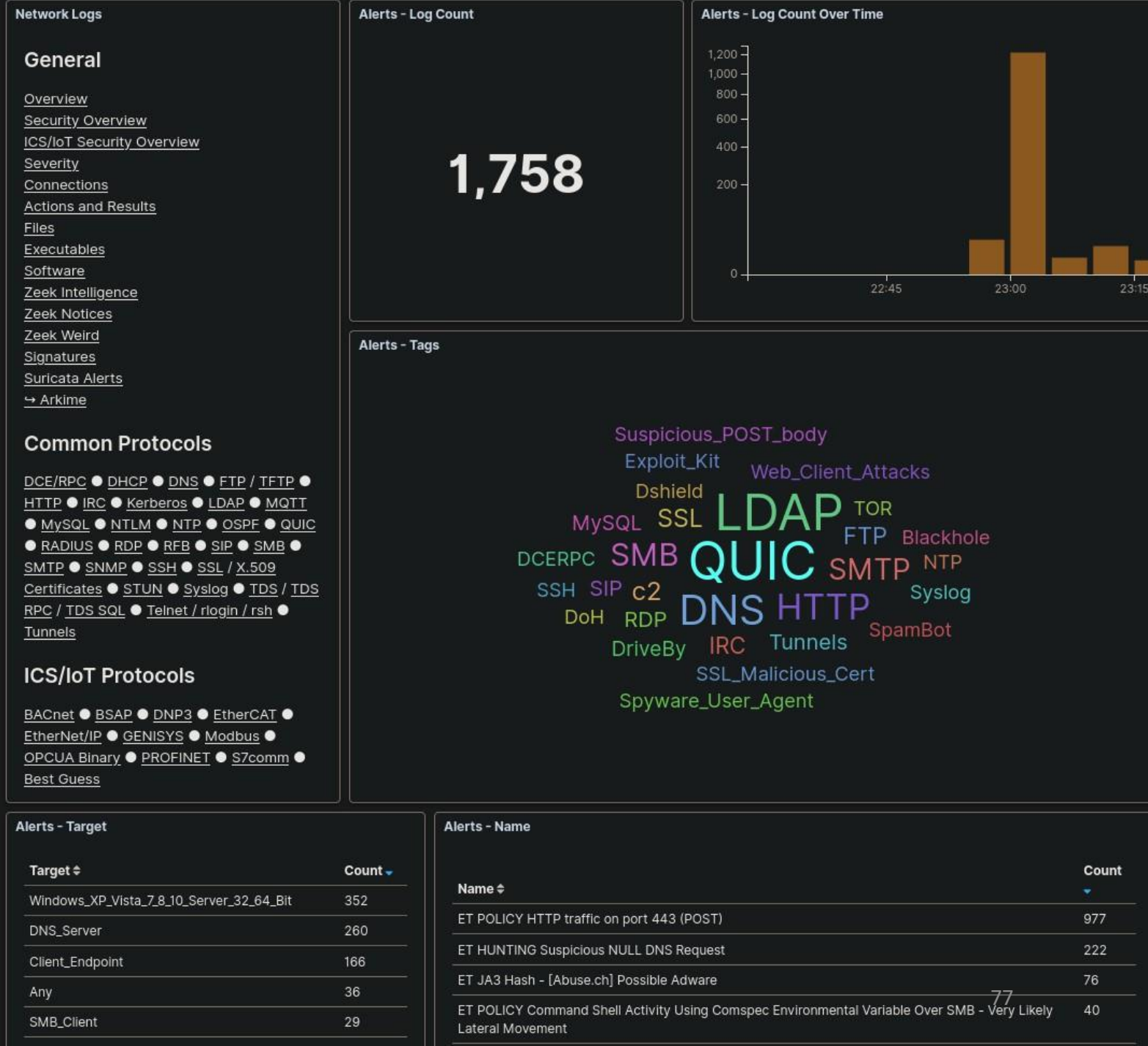
A suspicious file
Attempted

- Zeek notices are things that are odd or potentially bad
- In addition to Zeek's defaults, Malcolm raises notices for recent critical vulnerabilities and attack techniques

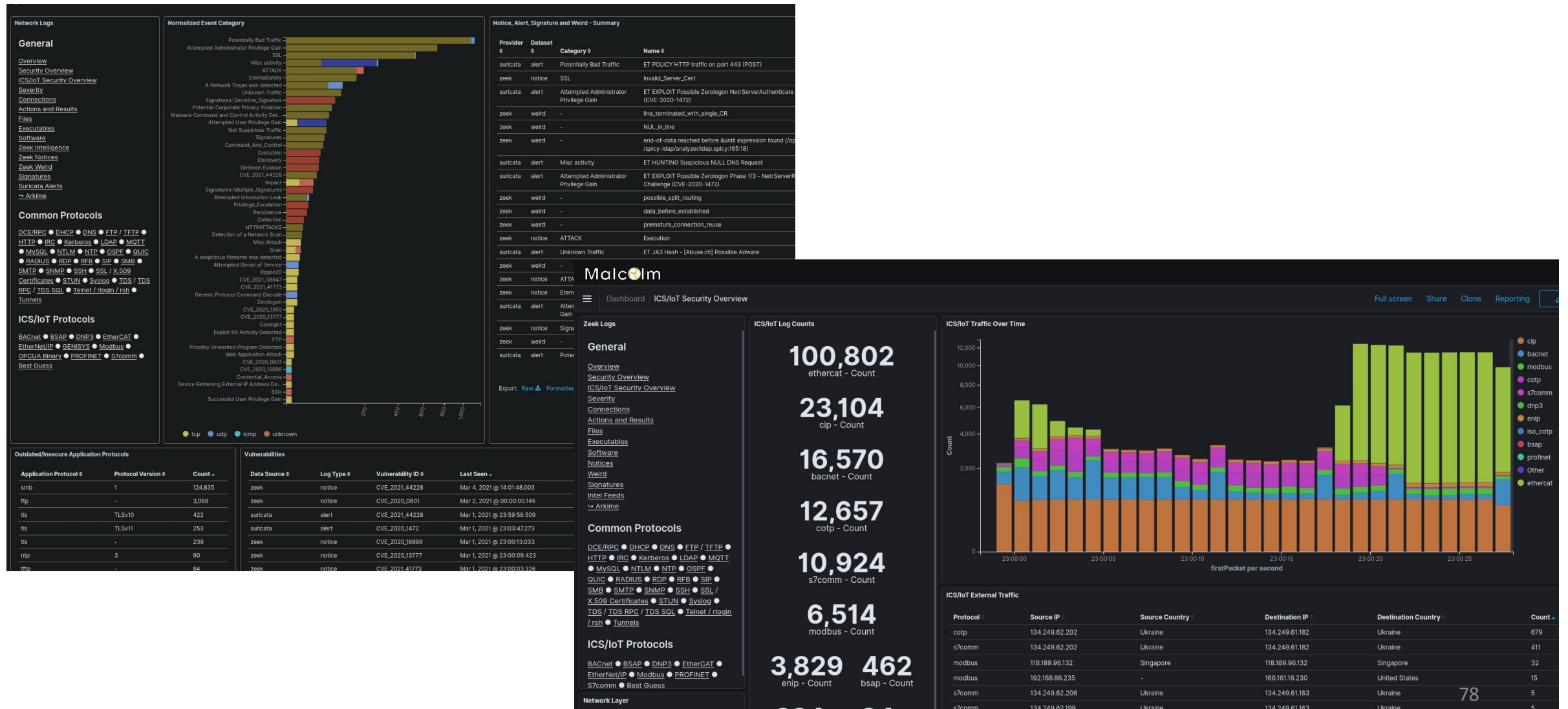
[illegible]

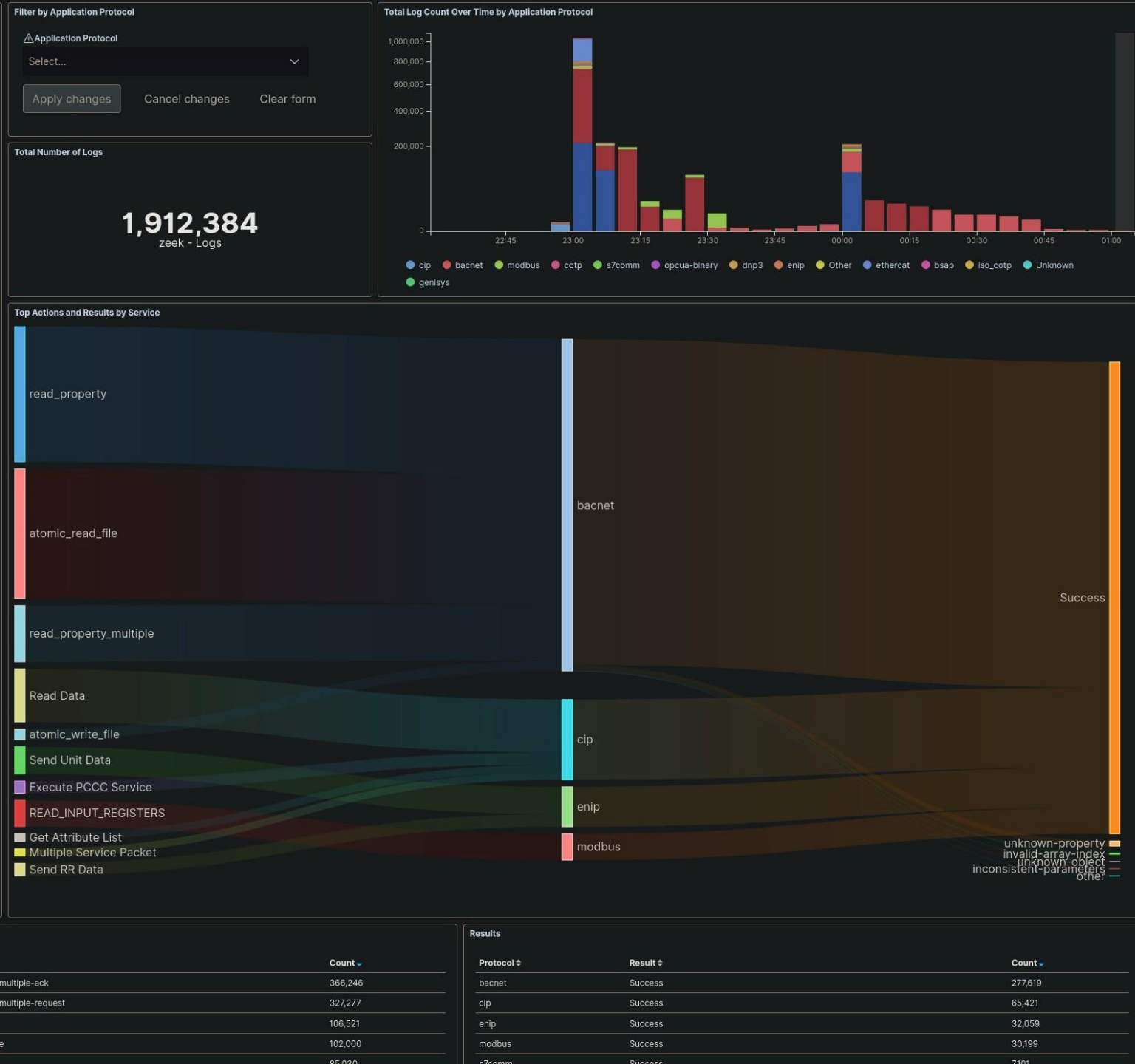
Suricata Alerts

- Protocol-aware Suricata signatures generate alerts for suspect traffic
- Use the default Emerging Threats Open ruleset or custom signatures from other sources



Security & ICS/IoT Security Overviews





Actions and Results

- Malcolm normalizes “action” (e.g., write, read, create file, logon, logoff, etc.) and “result” (e.g., success, failure, access denied, not found) across protocols

Protocol Dashboards

- Highlight application-specific fields of interest
- Grouped by common IT protocols and ICS/IoT protocols
- ICS protocols
 - BACnet
 - BSAP
 - DNP3
 - EtherCAT
 - EtherNet/IP
 - GENISYS
 - Modbus
 - OPCUA Binary
 - PROFINET
 - S7comm

[Zeek Intelligence](#)
[Zeek Notices](#)
[Zeek Weird](#)
[Signatures](#)
[Suricata Alerts](#)
[↔ Arkime](#)

Common Protocols

[DCE/RPC](#) ● [DHCP](#) ● [DNS](#) ● [FTP / TFTP](#) ●
[HTTP](#) ● [IRC](#) ● [Kerberos](#) ● [LDAP](#) ● [MQTT](#)
● [MySQL](#) ● [NTLM](#) ● [NTP](#) ● [OSPF](#) ● [QUIC](#)
● [RADIUS](#) ● [RDP](#) ● [RFB](#) ● [SIP](#) ● [SMB](#) ●
[SMTP](#) ● [SNMP](#) ● [SSH](#) ● [SSL / X.509](#)
[Certificates](#) ● [STUN](#) ● [Syslog](#) ● [TDS / TDS](#)
[RPC / TDS SQL](#) ● [Telnet / rlogin / rsh](#) ●
[Tunnels](#)

ICS/IoT Protocols

[BACnet](#) ● [BSAP](#) ● [DNP3](#) ● [EtherCAT](#) ●
[EtherNet/IP](#) ● [GENISYS](#) ● [Modbus](#) ●
[OPCUA Binary](#) ● [PROFINET](#) ● [S7comm](#) ●
[Best Guess](#)

Notices - Notice Type

Notice Category ▾

SSL

ATTACK

ATTACK

EternalSafety

Signatures

EternalSafety

ATTACK

EternalSafety

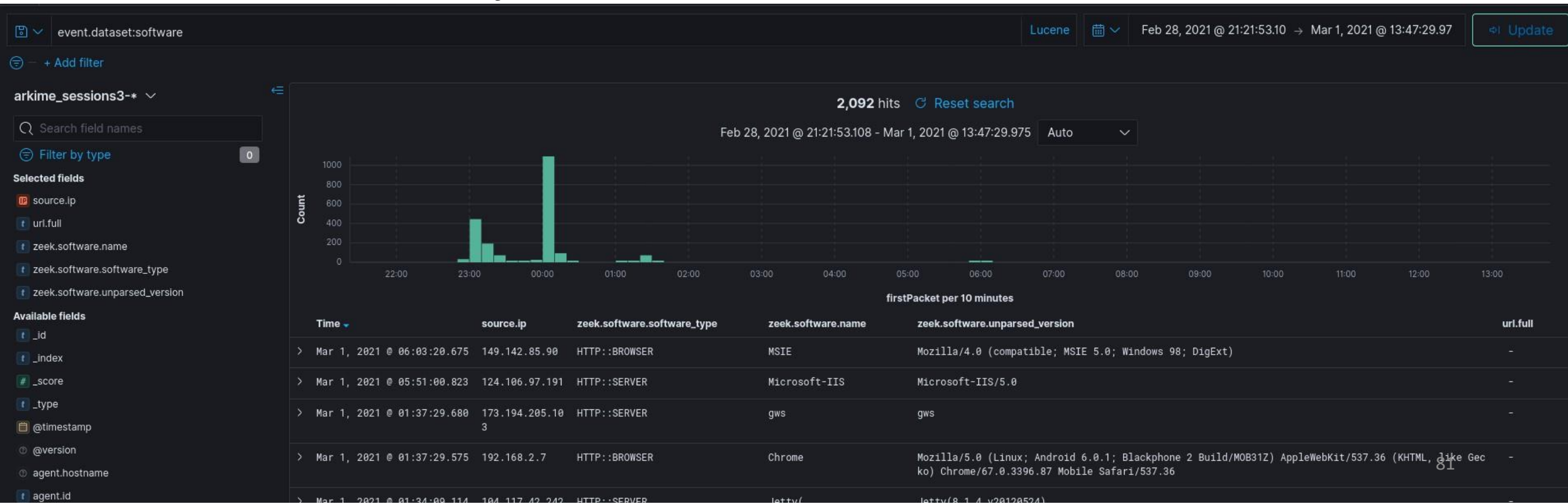
EternalSafety

ATTACK

Export: [Raw](#) [Download](#) [Format](#)

Discover

- Field-level details of logs matching filter criteria
- Create and view saved searches and column configurations
- View other events just before and after an event



New Visualization

Q Filter



Area



Controls



Coordinate
Map



Data Table



Gantt Chart



Gauge



Goal



Heat Map



Horizontal Bar



Line



Markdown



Metric



Pie



Region Map



Sankey
Diagram



TSVB



Tag Cloud



Timeline



Vega



Vertical Bar

Custom Visualizations

- Create new visualizations from scratch or based on existing charts or dashboards

Search Syntax Comparison

	Arkime	Dashboards (Lucene)	Dashboards (DQL)
Field exists	<code>event.dataset == EXISTS!</code>	<code>_exists_:event.dataset</code>	<code>event.dataset:*</code>
Field does not exist	<code>event.dataset != EXISTS!</code>	<code>NOT _exists_:event.dataset</code>	<code>NOT event.dataset:*</code>
Field matches a value	<code>port.dst == 22</code>	<code>destination.port:22</code>	<code>destination.port:22</code>
Field does not match a value	<code>port.dst != 22</code>	<code>NOT destination.port:22</code>	<code>NOT destination.port:22</code>
Field matches at least one of a list of values	<code>tags == [external_source, external_destination]</code>	<code>tags:(external_source OR external_destination)</code>	<code>tags:(external_source or external_destination)</code>
Field range (inclusive)	<code>http.statuscode >= 200 && http.statuscode <= 300</code>	<code>http.statuscode:[200 TO 300]</code>	<code>http.statuscode >= 200 and http.statuscode <= 300</code>

Search Syntax Comparison (cont.)

	Arkime	Dashboards (Lucene)	Dashboards (DQL)
Field range (exclusive)	<code>http.statuscode > 200 && http.statuscode < 300</code>	<code>http.statuscode:{200 TO 300}</code>	<code>http.statuscode > 200 and http.statuscode < 300</code>
Field range (mixed exclusivity)	<code>http.statuscode >= 200 && http.statuscode < 300</code>	<code>http.statuscode:[200 TO 300}</code>	<code>http.statuscode >= 200 and http.statuscode < 300</code>
Match all search terms (AND)	<code>(tags == [external_source, external_destination]) && (http.statuscode == 401)</code>	<code>tags:(external_source OR external_destination) AND http.statuscode:401</code>	<code>tags:(external_source or external_destination) and http.statuscode:401</code>
Match any search terms (OR)	<code>(zeek_ftp.password == EXISTS!) (zeek_http.password == EXISTS!) (zeek.user == "anonymous")</code>	<code>_exists_:zeek_ftp.password OR _exists_:zeek_http.password OR zeek.user:"anonymous"</code>	<code>zeek_ftp.password:* or zeek_http.password:* or zeek.user:"anonymous"</code>

Search Syntax Comparison (cont.)

	Arkime	Dashboards (Lucene)	Dashboards (DQL)
Global string search (anywhere in the document)	all Arkime search expressions are field-based	microsoft	microsoft
Wildcards	host.dns == "*micro?oft*" (? for single character, * for any characters)	dns.host:*micro?oft* (? for single character, * for any characters)	dns.host:*micro*ft* (* for any characters)
Regex	host.http == /. *www\ .f.*k\ .com.* /	zeek_http.host: /. *www\ .f.*k\ .com.* /	Dashboards Query Language does not currently support regex
IPv4 values	ip == 0.0.0.0/0	source.ip:"0.0.0.0/0" OR destination.ip:"0.0.0.0/0"	source.ip:"0.0.0.0/0" OR destination.ip:"0.0.0.0/0"
IPv6 values	(ip.src == EXISTS! ip.dst == EXISTS!) && (ip != 0.0.0.0/0)	(_exists_:source.ip AND NOT source.ip:"0.0.0.0/0") OR (_exists_:destination.ip AND NOT destination.ip:"0.0.0.0/0")	(source.ip:* and not source.ip:"0.0.0.0/0") or (destination.ip:* and not destination.ip:"0.0.0.0/0")

Search Syntax Comparison (cont.)

	Arkime	Dashboards (Lucene)	Dashboards (DQL)
GeoIP information available	<code>country == EXISTS!</code>	<code>_exists_:destination.geo OR _exists_:source.geo</code>	<code>destination.geo:* or source.geo:*</code>
Log type	<code>event.dataset == notice</code>	<code>event.dataset:notice</code>	<code>event.dataset:notice</code>
IP CIDR Subnets	<code>ip.src == 172.16.0.0/12</code>	<code>source.ip:"172.16.0.0/12"</code>	<code>source.ip:"172.16.0.0/12"</code>
Search time frame	Use Arkime time bounding controls under the search bar	Use Dashboards time range controls in the upper right-hand corner	Use Dashboards time range controls in the upper right-hand corner
GeoIP information available	<code>country == EXISTS!</code>	<code>_exists_:destination.geo OR _exists_:source.geo</code>	<code>destination.geo:* or source.geo:*</code>

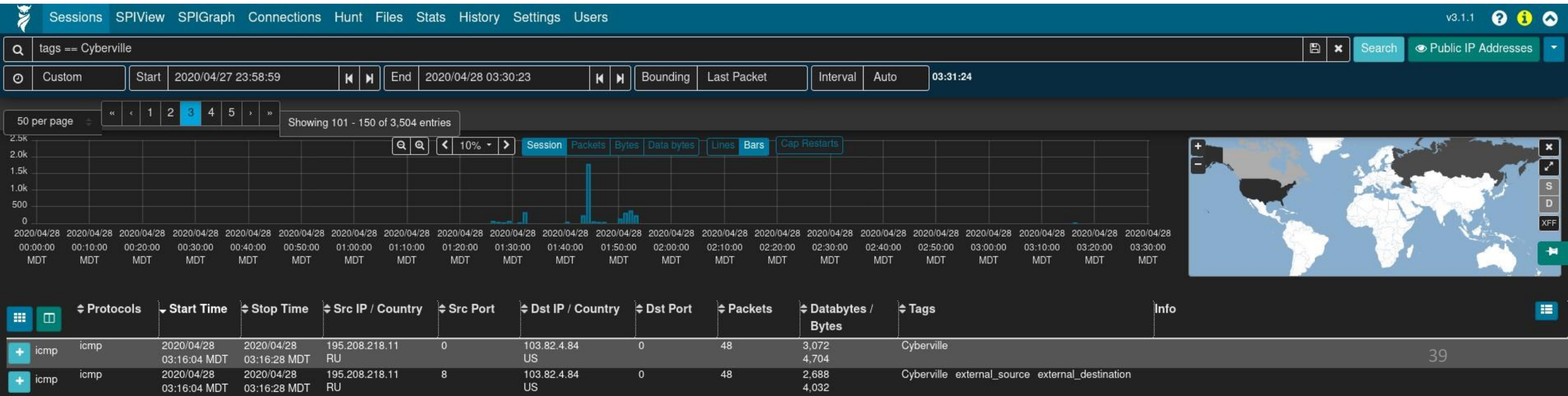
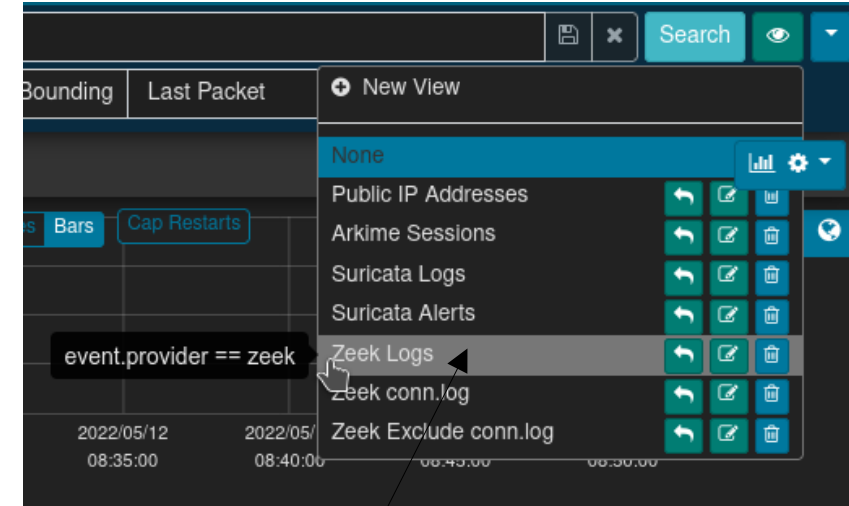


Arkime

- Front end for **both** enriched Zeek logs, Suricata alerts and Arkime sessions
 - Malcolm's custom Arkime Zeek data source adds full support for Zeek logs to Arkime, including ICS protocols
- Filter by data source (Zeek, Suricata or Arkime); or, view together
- “Wireshark at scale”: full PCAP availability for
 - viewing packet payload
 - exporting filtered and joined PCAP sessions
 - running deep-packet searches
- <https://localhost>

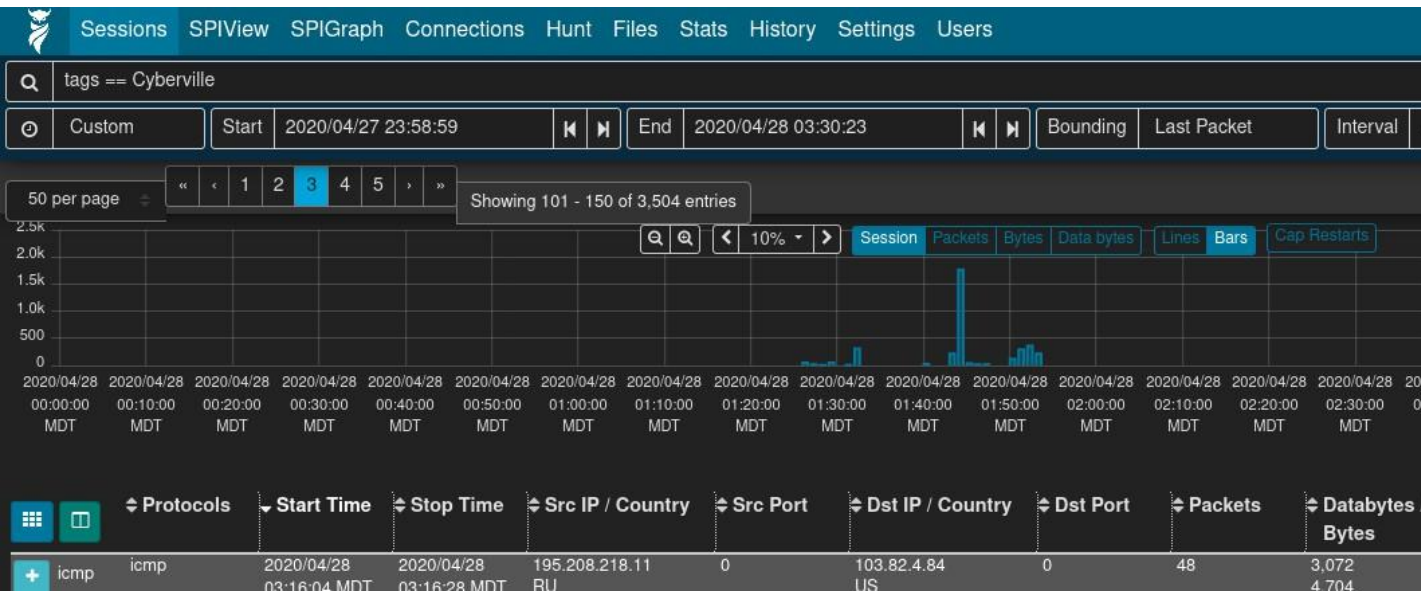
Arkime Filters and Search

- Time filter: define search time frame
- Map filter: restrict results to geolocation
- Query bar: write queries in Arkime syntax
- Views: overlay previously-specified filters on current search



Sessions

- Field-level details of sessions/logs matching filters
- Similar to Dashboards' Discover



The screenshot shows the 'Sessions' tab in a dashboard. The search bar contains 'protocols == http && tags == external_destination'. The time range is from 2020/11/11 06:23:48 to 2021/05/30 06:00:53. The interface displays a list of session details. The first entry is highlighted, showing the following information:

- Log Type: http
- Malcolm Data Source: zeek
- Malcolm Node: filebeat
- Originating Host: 217.226.31.170
- Originating GeoIP Country: Germany
- Originating GeoIP City: Bremen
- Responding Host: 124.106.97.191
- Responding GeoIP Country: Philippines
- Responding GeoIP City: Santa Elena
- Originating Port: 4230
- Responding Port: 80
- Related IP: 217.226.31.170 124.106.97.191
- Protocol: tcp
- Service: http
- Service Version: 1.1
- Action: GET
- Result: Bad Gateway
- Severity: 20
- Risk Score: 20
- Severity Tags: External traffic
- File Magic: text/html

Below the session details, there is a section titled 'Zeek http.log' showing the following information:

- Pipeline Depth: 1
- Request Method: GET
- URI: /_vti_bin/.../winnt/system32/cmd.exe?/c+dir+x:\\ c+dir+x:\\ c+dir+x:\\
- Version: 1.1

Packet Payloads

- Displayed for Arkime sessions with full PCAP (i.e., not Zeek logs)
- File carving on the fly
- Download session PCAP
- Examine payload with CyberChef

Source	Destination
GET /PostExploitation/PCAnyPass.exe HTTP/1.1 Accept: text/html, application/xhtml+xml, */* Referer: http://10.10.10.11/PostExploitation/ Accept-Language: en-US User-Agent: Mozilla/5.0 (compatible; MSIE 9.0; Windows NT 6.1; Trident/5.0) Accept-Encoding: gzip, deflate Host: 10.10.10.11 Connection: Keep-Alive	HTTP/1.0 200 OK Server: SimpleHTTP/0.6 Python/2.7.17 Date: Fri, 17 Apr 2020 19:21:32 GMT Content-type: application/x-msdos-program Content-Length: 49152 Last-Modified: Fri, 16 Apr 2010 19:09:50 GMT PCAnyPass.exe

Packets200naturalasciitf8hexShow PacketsLine NumbersUncompressShow Image & FilesShow Info

File Bytes:

base64CyberChef

Export PCAP

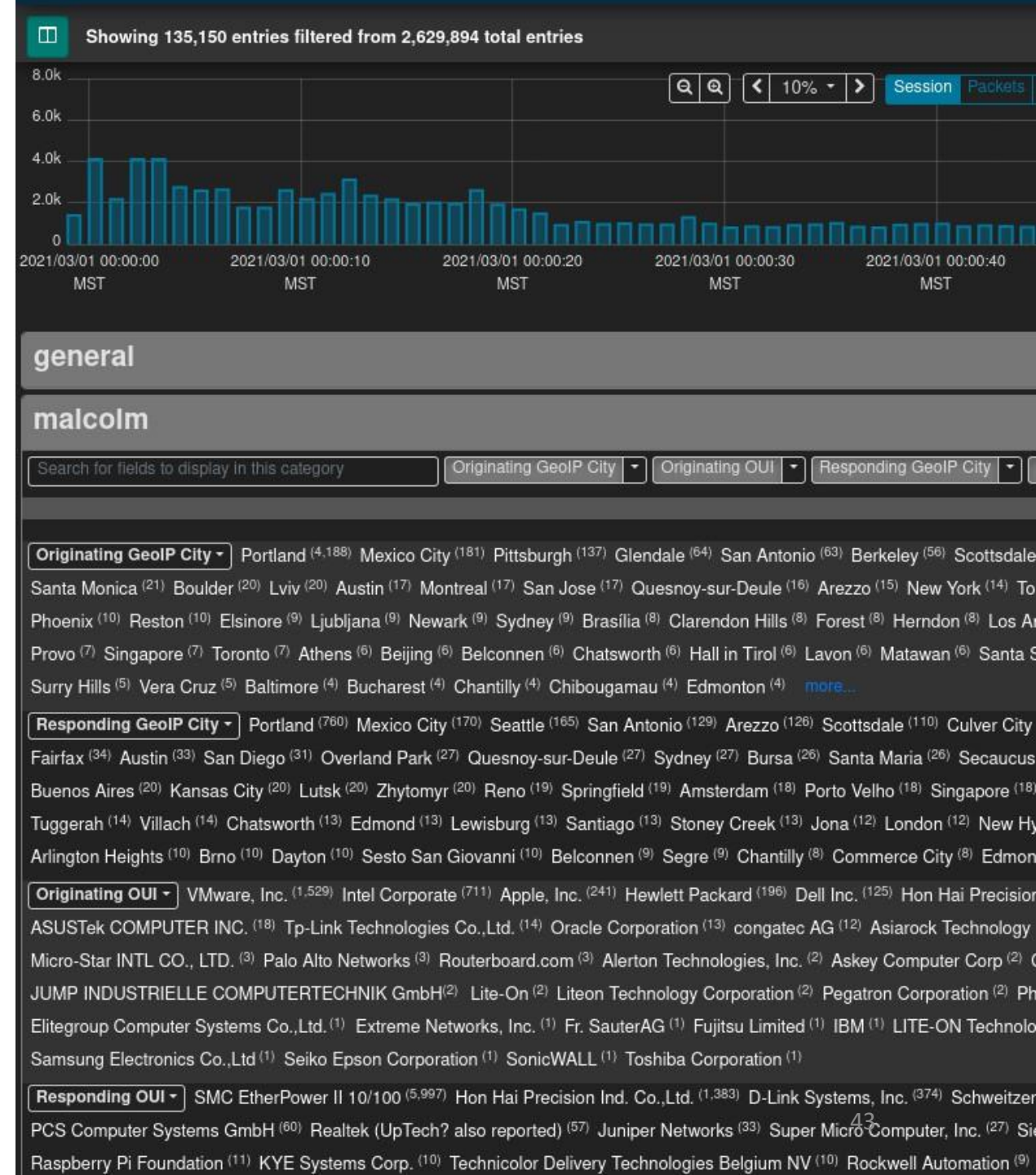
- Creates a new PCAP file from filtered sessions
- Include open, visible or all matching sessions
- Apply “Arkime Sessions” view to sessions first
- Narrow as much as possible prior to exporting (huge PCAP files are a pain)

The screenshot displays the Arkime Sessions web interface. At the top, a navigation bar includes links for Sessions, SPIView, SPIGraph, Connections, Hunt, Files, Stats, History, Settings, and Users. A search bar contains the query 'country != US && protocols == http'. Below the search bar, a filter section shows 'Custom' filters for Start (2021/02/28 23:59:11) and End (2021/03/01 00:28:26), with a duration of 00:29:15. The 'Include' section is set to 'same time period' and 'linked segments (slow)'. The 'Filename' field is 'US_HTTP.pcap'. An 'Export PCAP' button is visible on the right. The main area shows a bar chart of session activity over time, with a table below it. The table has columns for Protocols, Start Time, Stop Time, Src IP / Country, Src Port, Dst IP / Country, Dst Port, Packets, Databytes / Bytes, Tags, and Info. The first row shows a session for 'tcp http' on 2021/03/01 from 10.0.52.164 to 61.8.0.17 on port 2550, with 7,195 packets and 5,160,414 bytes. A world map is visible in the bottom right corner.

Protocols	Start Time	Stop Time	Src IP / Country	Src Port	Dst IP / Country	Dst Port	Packets	Databytes / Bytes	Tags	Info
tcp http	2021/03/01	2021/03/01	10.0.52.164	2550	61.8.0.17	80	7,195	5,160,414	HTTP out-of-order-dst	URI mirror.pacific.net.au/openoffice/stable/2.0.0/OOo_2.0.0_Win32Intel_install.exe

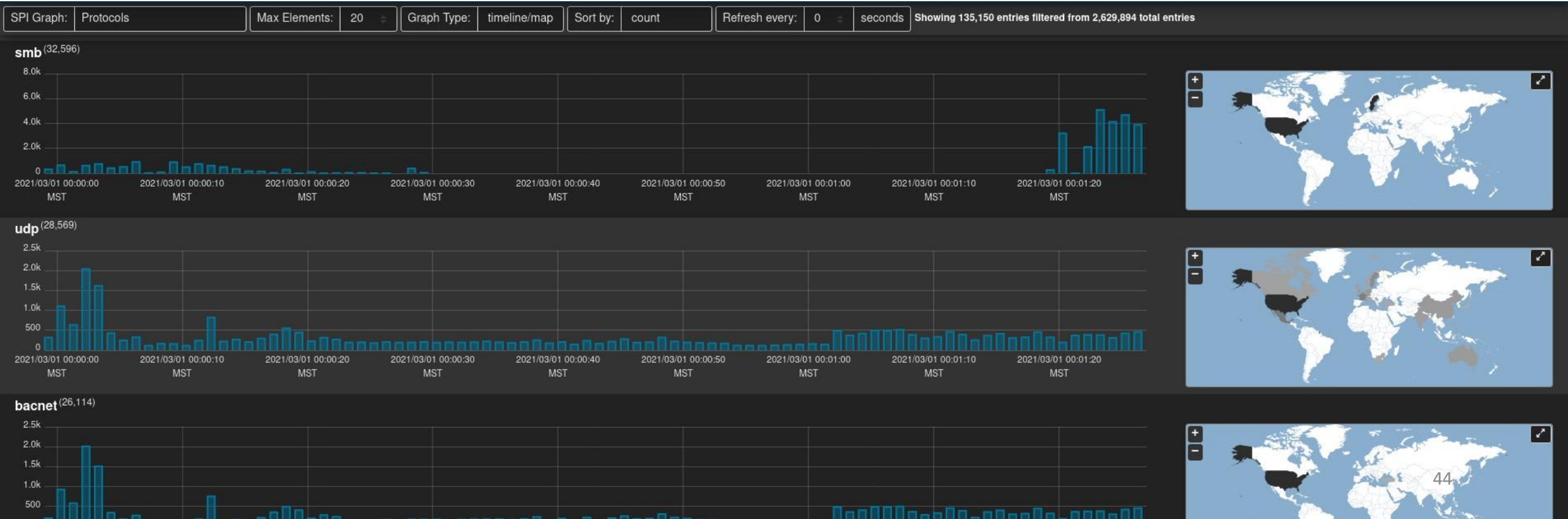
SPIView

- Explore “top n ” and field cardinality for all fields of both Arkime sessions and Zeek logs
- Apply filters or pivot to Sessions or SPIGraph view for field values of interest
- Limit search to ≤ 1 week before using (it runs many queries)



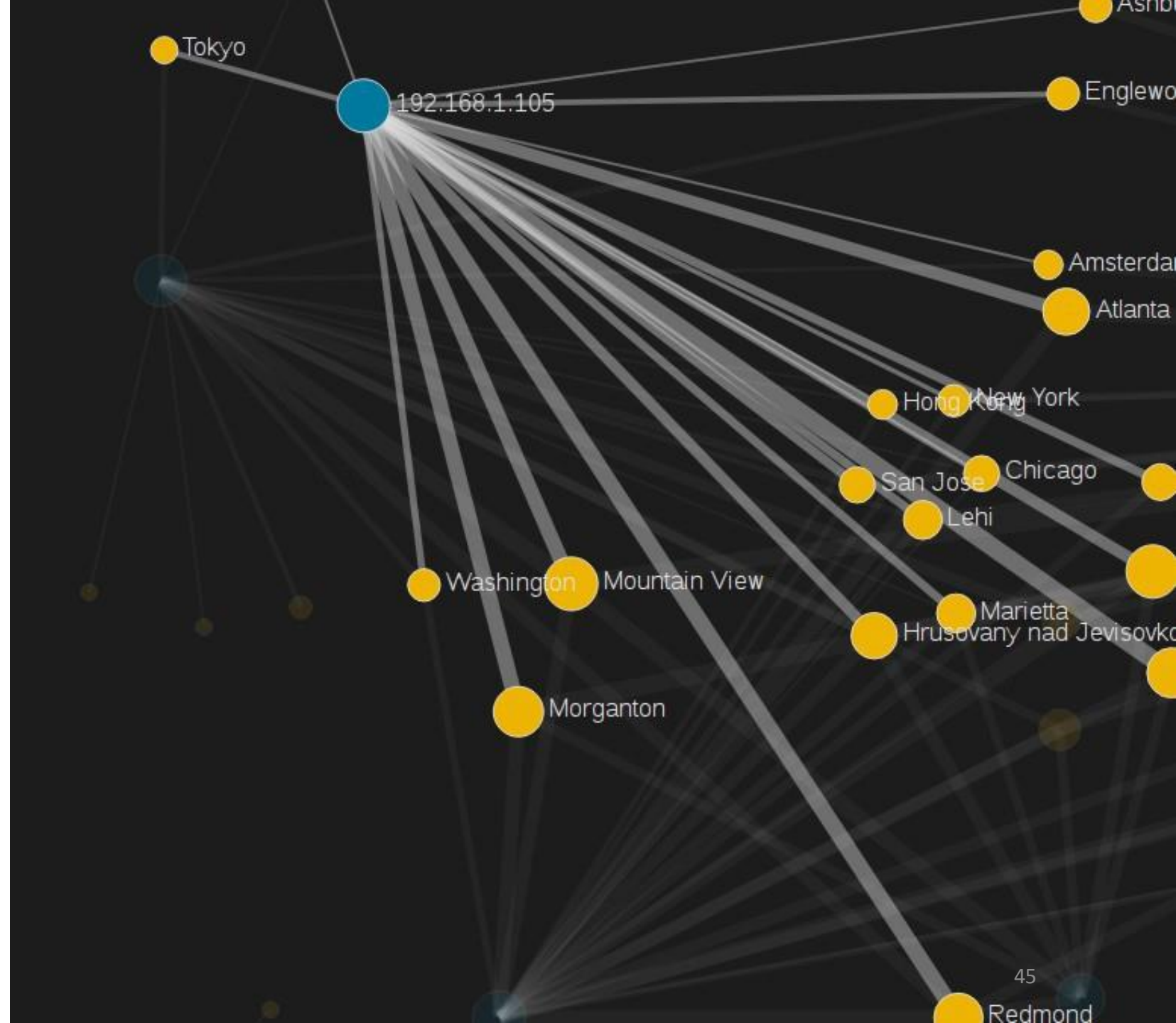
SPIGraph

- View “top n ” field values chronologically and geographically
- Identify trends and patterns in network traffic



Connections

- Visualize logical relationship between hosts
- Use any combination of fields for source and destination nodes
- Compare current vs. previous (baseline) traffic



Packet Search (“Hunt”)

- Deep-packet search (“PCAP grep”) of session payloads
- Search for ASCII, hex codes or regular expression matches
- Apply “Arkime Sessions” view to sessions first

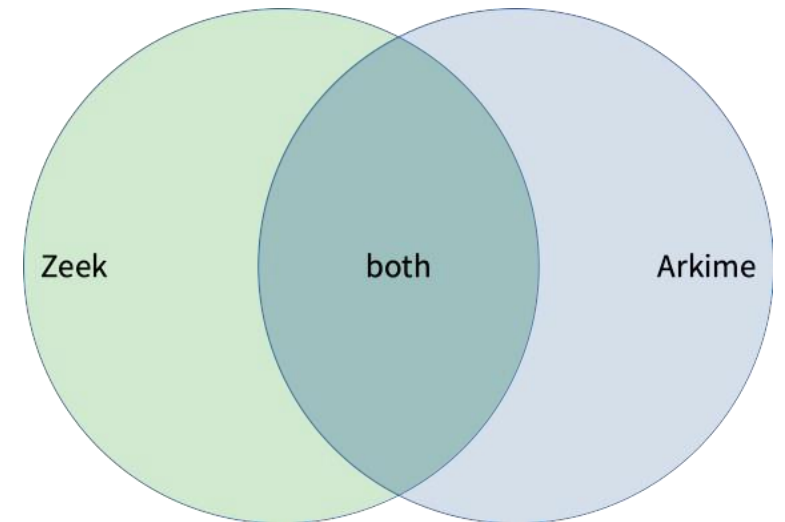
The screenshot displays the Arkime Hunt interface. At the top, a navigation bar includes links for Sessions, SPIView, SPIGraph, Connections, Hunt (active), Files, Stats, History, Settings, and Users. A search bar contains the query 'protocols == http'. Below this, a status bar indicates 'All (careful)' and provides start/end time filters (1969/12/31 17:00:00 to 2021/12/06 12:10:02) and bounding options. A message states: 'Creating a new packet search job will search the packets of 2,906 sessions.' The main section is titled 'Hunt Job History' and features a search bar for the job history. A table lists the search results:

Status	Matches	Name	User	Search text	Notify	Created	ID
100%	141	HTTP with password		password (ascii)		2021/12/06 12:12:27 MST	s5YpkX0BTA40FhD4X7dA

Below the table, a summary indicates the hunt is finished, found 141 sessions matching 'password (ascii)' out of 2,908 searched. It also lists the creation time (2021/12/06 12:12:27 MST), last update time (2021/12/06 12:12:32 MST), and the search criteria: 'protocols == http' and 'Arkime Sessions' view.

Data Source Correlation

- Search syntax is different between Arkime and Dashboards (and in some cases, so are field names)
 - See search syntax comparison table, Malcolm and Arkime docs
- Despite considerable overlap, there are differences in protocol parser support among Zeek, Suricata and Arkime
 - Learning the strengths of each will help you more effectively find the good stuff



Correlate Zeek or Suricata Logs and Packet Payloads

- Correlate Zeek or Suricata logs and Arkime sessions using common fields
- `communityId` fingerprints flows to bridge data sources
- `rootId/event.id` filters logs for the same session
- Filter community ID OR'ed with `event.id` to see all Arkime sessions and Zeek or Suricata logs for the same traffic

```
communityId == "1:deE1x8qWJLk/UI6TKoVasnf4FZw=" || event.id == "C2EHCo28ylyUYz82Wk || event.id == 1553655336047566"
```

communityId == "1:deE1x8qWJLk/UI6TKoVasnf4FZw=" event.id == C2EHCo28ylyUYz82Wk event.id == 1553655336047566											
Custom Start 2021/03/01 00:06:01 End 2021/03/01 00:06:58 Bounding Last Packet Interval Auto 00:00:57											
50 per page Showing 1 - 6 of 6 entries											
	Protocols	Data Source	Log Type	Start Time	Stop Time	Src IP / Country	Dst IP / Country	Dst Port	Packets	Databytes / Bytes	Info
+	http http	zeek	files	2021/03/01 00:06:16	2021/03/01 00:06:16	204.152.184.134 US	10.0.52.164			6,550,964 78,597,807	Severity Tags File transfer (high concern)
+	tcp tcp http	zeek	http	2021/03/01 00:06:16	2021/03/01 00:06:16	10.0.52.164	204.152.184.134 US	80		6,550,964	URI mirrors.isc.org/pub/openoffice/stable/2.0.0/OOo_2.0.0_Win32Intel_install.exe Severity Tags Outbound traffic File transfer (high concern)
+	tcp tcp http	arkime	session	2021/03/01 00:06:16	2021/03/01 00:06:39	10.0.52.164	204.152.184.134 US	80	9,354	6,551,694 7,057,419	URI mirrors.isc.org/pub/openoffice/stable/2.0.0/OOo_2.0.0_Win32Intel_install.exe
+	tcp tcp http	suricata	alert	2021/03/01 00:06:16	2021/03/01 00:06:17	204.152.184.134 US	10.0.52.164	2646	58	47,435 47,435	Severity Tags Suricata Alert Inbound traffic Service on non-standard port
+	tcp tcp http	zeek	conn	2021/03/01 00:06:16	2021/03/01 00:06:39	10.0.52.164	204.152.184.134 US	80	9,354	6,551,694 6,926,454	Severity Tags Outbound traffic
+		zeek	signatures	2021/03/01 00:06:16	2021/03/01 00:06:16						Severity Tags Signature

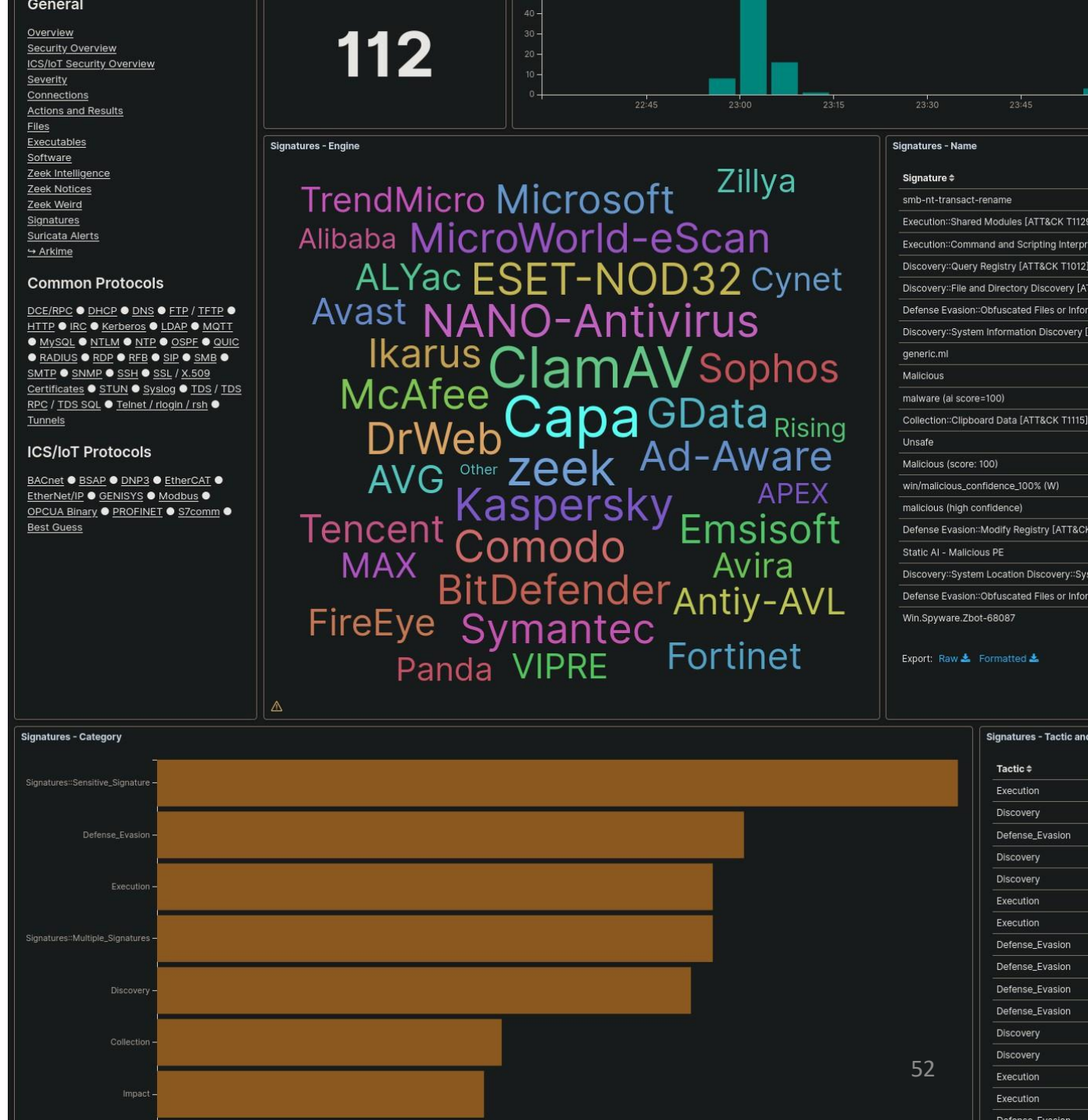
File Analysis

- Zeek can “carve” file transfers from common protocols
- Malcolm can examine carved files and flag hits
 - ClamAV – open source antivirus engine
 - YARA – pattern matching swiss army knife
 - Capa – portable executable capabilities analyzer
 - VirusTotal – online database of file hashes
 - requires API token and internet connection
- Triggering files can be saved to `zeek-logs/extract_files` under Malcolm directory for further analysis
 - Be careful! Carved files may contain live malware!



Signatures

- Signatures dashboard in Dashboards shows scanned file hits
- Use `zeek.fuid` field in *Signatures – Logs* table to pivot to connection UID (`zeek.uid`) and other logs with pertinent session details



Search Tips

- Always check your search time frame
- “Zoom in” (apply filters) for a particular field value, pivot to another field then “zoom out” (remove filters)
- Most UI controls can work with any data field (2000+)
- Filter on `event.dataset` (e.g., `conn` to see `conn.log`)
- Filter on protocol regardless of data source (e.g., `protocol:http` in Dashboards and `protocols == http` in Arkime)
- Use tags