

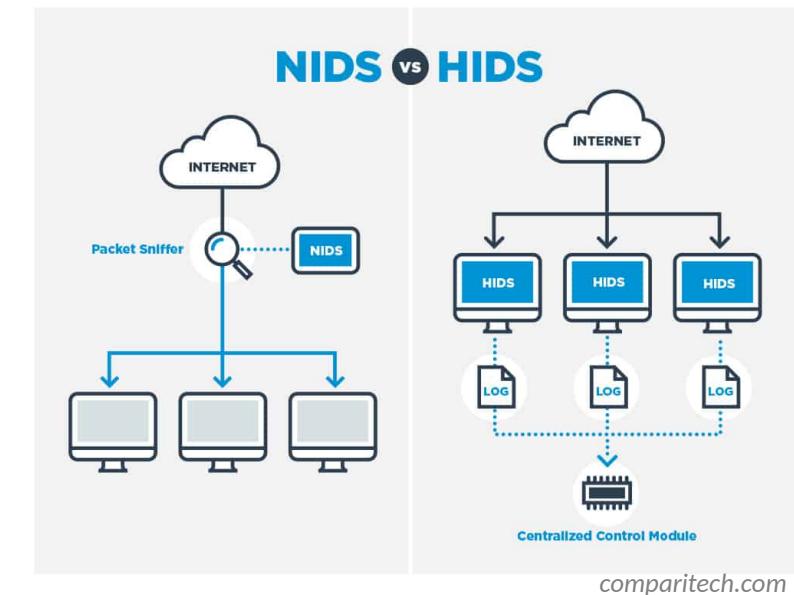
# Network Traffic Analysis with **Malcolm**

A faint watermark of the Malcolm logo is visible behind the word 'Malcolm'. The logo consists of a stylized yellow 'M' shape containing a circular emblem with three interlocking rings.

Seth Grover, Malcolm developer • Cybersecurity R&D • Idaho National Lab

# 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

# Zeek Log Files

- Network Protocols
  - Files
  - Detection
  - Network Observations

**conn.log** | IP, TCP, UDP, ICMP connection details

FIELD	TYPE	DESCRIPTION
to	time	Timestamp of the first packet
sid	string	Unique ID of the connection
string_n	addr	Originating endpoint IP address:string
string_p	port	Originating endpoint PORT:TCP/UDP port for TCP/UDP
to_ip_n	addr	Responding endpoint IP address:string
to_ip_p	port	Responding endpoint PORT:TCP/UDP port for TCP/UDP
proto	proto	Transport layer protocol of connection
service	string	Detected application protocol, if any
duration	interval	Connection length
string_ipseq	vector	Orig payload bytes from sequence numbers of TCP
resp_ipseq	vector	Resp payload bytes from sequence numbers of TCP
conn.state	string	Connection status (one conn.log = one state)
first_orig	bool	Is Orig in this log, resp?
last_resp	bool	Is Resp in this log, resp?
missed_ipseqs	vector	Number of bytes missing due to current gaps
history	string	Connection state history (one conn.log = history!)
orig_pkts	vector	Number of Orig packets
orig_ip_pkts	vector	Number of Orig IP packets (one IP header, length header field)
resp_pkts	vector	Number of Resp packets
resp_ip_pkts	vector	Number of Resp IP packets (one IP header, length header field)
tcpd	set	If Turned on, connection I/O of encapsulating connection
orig_ip_addr	string	Low-layer address of the originator
resp_ip_addr	string	Low-layer address of the responder
view	int	The user VLAN for this connection
inner_vlans	int	The inner VLAN for this connection

**http.log** | HTTP request/reply details

FIELD	TYPE	DESCRIPTION
to	time	Timestamp of the HTTP request
req_id	string	Underlying connection info - See conn.log
trans_depth	vector	Protocol depth into the connection
method	string	HTTP Request verb (GET, POST, etc) as string
host	string	Name of the Host header
uri	string	URI used in the request
referer	string	Name of the Referer header
user_agent	string	Name of the User-Agent header
response_body_hex	vector	Uncompressed content size of the data
response_body_hex_size	vector	Uncompressed content size of the data
status_code	vector	Status code returned by the server
status_msg	string	Status message returned by the server
info_code	vector	Last seen / has reply message by server
info_msg	string	Last seen / has reply message by server
tags	set	Indication of various attributes discovered
username	string	Username of basic auth if performed
password	string	Password of basic auth if performed
proxied	set	Headers indicative of a proxied request
orig_host	vector	The unique Origin-Host
orig_header_name	vector	The names from Origin-Header
orig_header_value	vector	The types from Origin-Header
resp_header_name	vector	The names from Resp-Header
resp_header_value	vector	The types from Resp-Header
client_header_name	vector	The names of HTTP headers sent by client
server_header_name	vector	The names of HTTP headers sent by host
cookie_name	vector	Variable names extracted from cookie
set_cookie	vector	Variable names extracted from the URL
url_params	vector	URL parameters
url_query	vector	URL query parameters

files.log   File analysis results		
FIELD	TYPE	DESCRIPTION
id	int	Timestamp when the file was processed
file	string	Unique identifier for every file
is_header	set	Message that received the data
re_header	set	Message that received the data
conn_id	set	Connection ID(s) over which the transferred
resource	string	An identification of the resource of the file data
depths	count	Depth of the related to resource e.g., HTTP request depths
analysis	set	Set of analyses attached during the analysis
miner_type	string	The specific algorithm by the miner
filename	string	Filename, Hashvalue or Filepath
duration	interval	The duration that the file was analyzed
local_eng	bool	Did the file originate locally?
is_eng	bool	Was the file engine originated?
anon_ipaddr	count	Number of files processed by the analysis engine
total_types	count	Total number of types that should comprise one file
missing_types	count	Number of types in the stream missed
overflow_types	count	Number of types in the stream due to overflow
threshold	bool	If the file analysis timed out at least once
parent_file	string	Contains the ID of the file extracted from
modified	string	Modified hash of the file
extracted	string	Local filename of download files, if analysis
entropy	double	Information density of the file contents

pe.log   Portable Executable (PE)		
FIELD	TYPE	DESCRIPTION
is	bool	Current processing
file	string	The full file location relative to the current working directory.
machine	string	The target machine that the file was compiled for.
compile_id	bool	True if the file was created at the time of compilation.
os	string	The required operating system.
subsystems	string	The subsystems that are required to run the file.
is_executable	bool	Is the file executable, or just an object file?
is_dll	bool	Is the file a dynamic linked library?
is_com	bool	Does the file support COM (Component Object Model)?
is_msil	bool	Does the file support Base-Build (.NET Framework)?
is_coff	bool	Does the file have a COFF header?
is_pe32	bool	Does the file use 32-bit architecture?
has_import_table	bool	Does the file have an import table?
has_export_table	bool	Does the file have an export table?
has_nt_headers	bool	Does the file have an NT header?
has_stripping_table	bool	Does the file have a stripping table?
sections_names	vector	The names of the sections, in order.

[corelight.com](http://corelight.com)

# 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.zeek.org/en/stable/scripts/base/frameworks/notice/weird.zeek.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

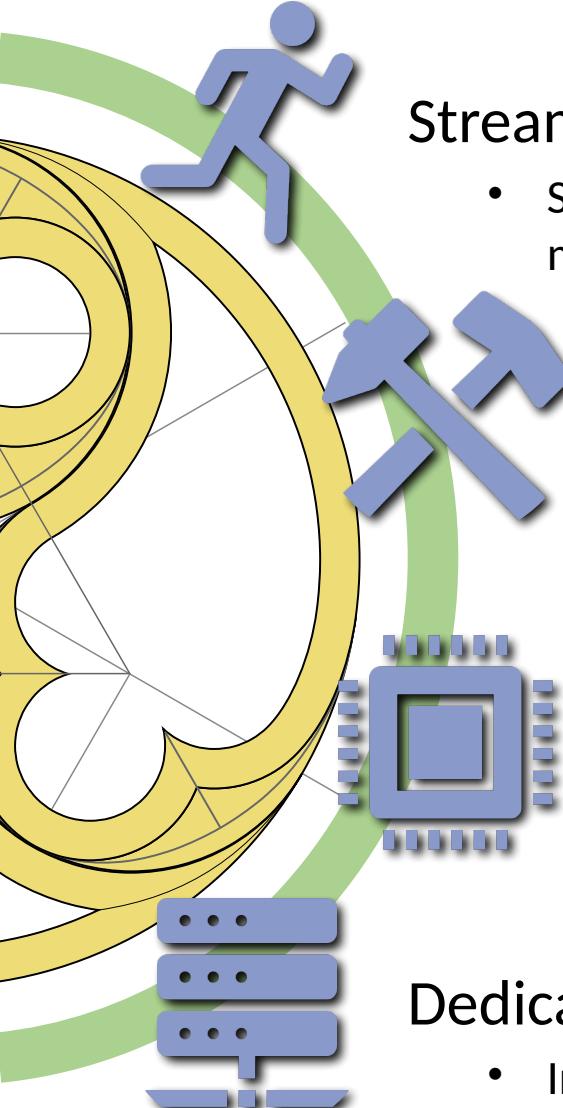
## Weaknesses

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



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>



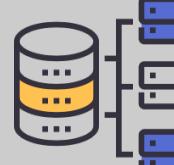
Capture &  
Analysis



File Scanning



Forwarding &  
Enrichment



Storage



Anomaly  
Detection



Anomaly  
Detection  
Plugin



Alerting



Alerting  
Plugin



Asset  
Management



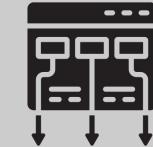
Visualization



OpenSearch  
Dashboards



Payload  
Analysis



Framework



TCPDUMP



OpenSearch



netbox



Arkime



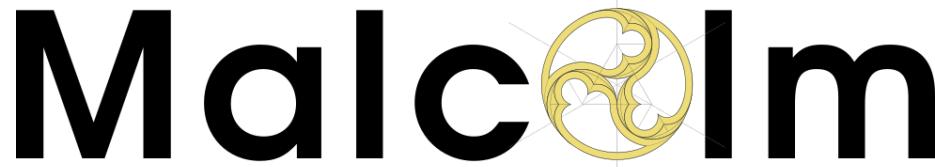
CyberChef



docker



Arkime  
session PCAP  
export to  
WIRESHARK



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

# Supported Protocols

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

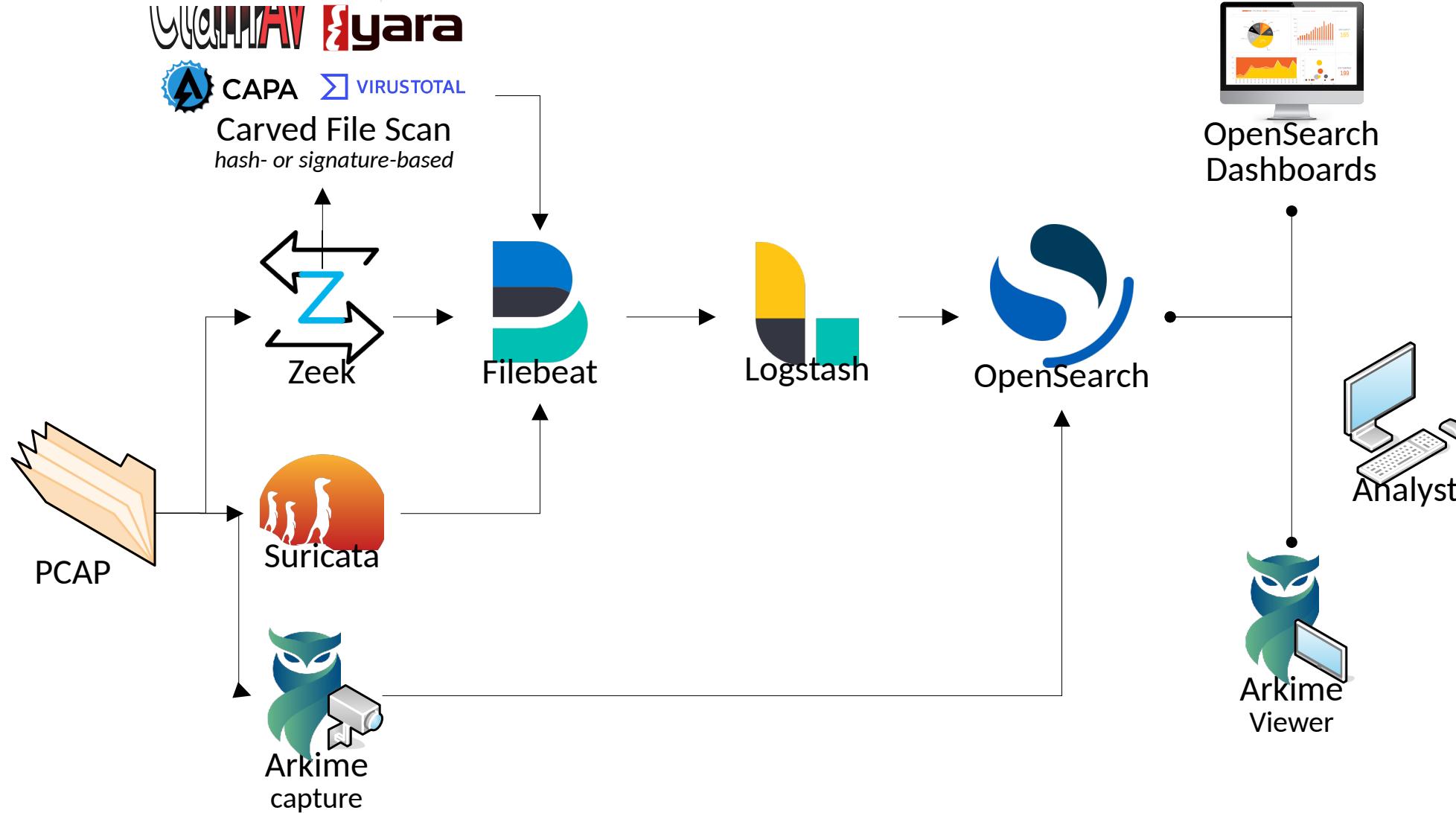
**Open Platform Communications Unified Architecture (OPC UA) Binary**  
Open Shortest Path First (OSPF)  
OpenVPN  
PostgreSQL  
**Process Field Net (PROFINET)**  
Remote Authentication Dial-In User Service (RADIUS)  
Remote Desktop Protocol (RDP)  
Remote Framebuffer / Virtual Network Computing (RFB/VNC)  
**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)  
**Synchrophasor (IEEE-C37.118)**  
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.)

# Malcolm



## Data Pipeline

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

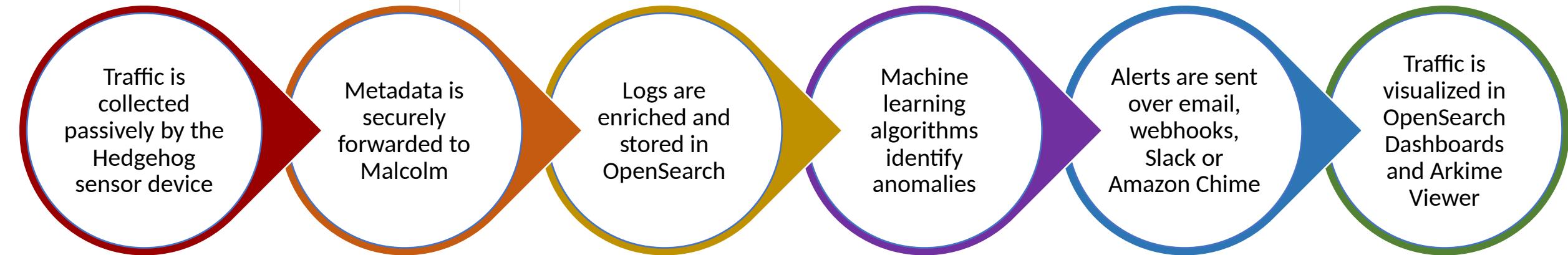


# Malcolm



## Data Pipeline

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



- 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 dedicated sensor

- 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

- 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-tiered Malcolm instance

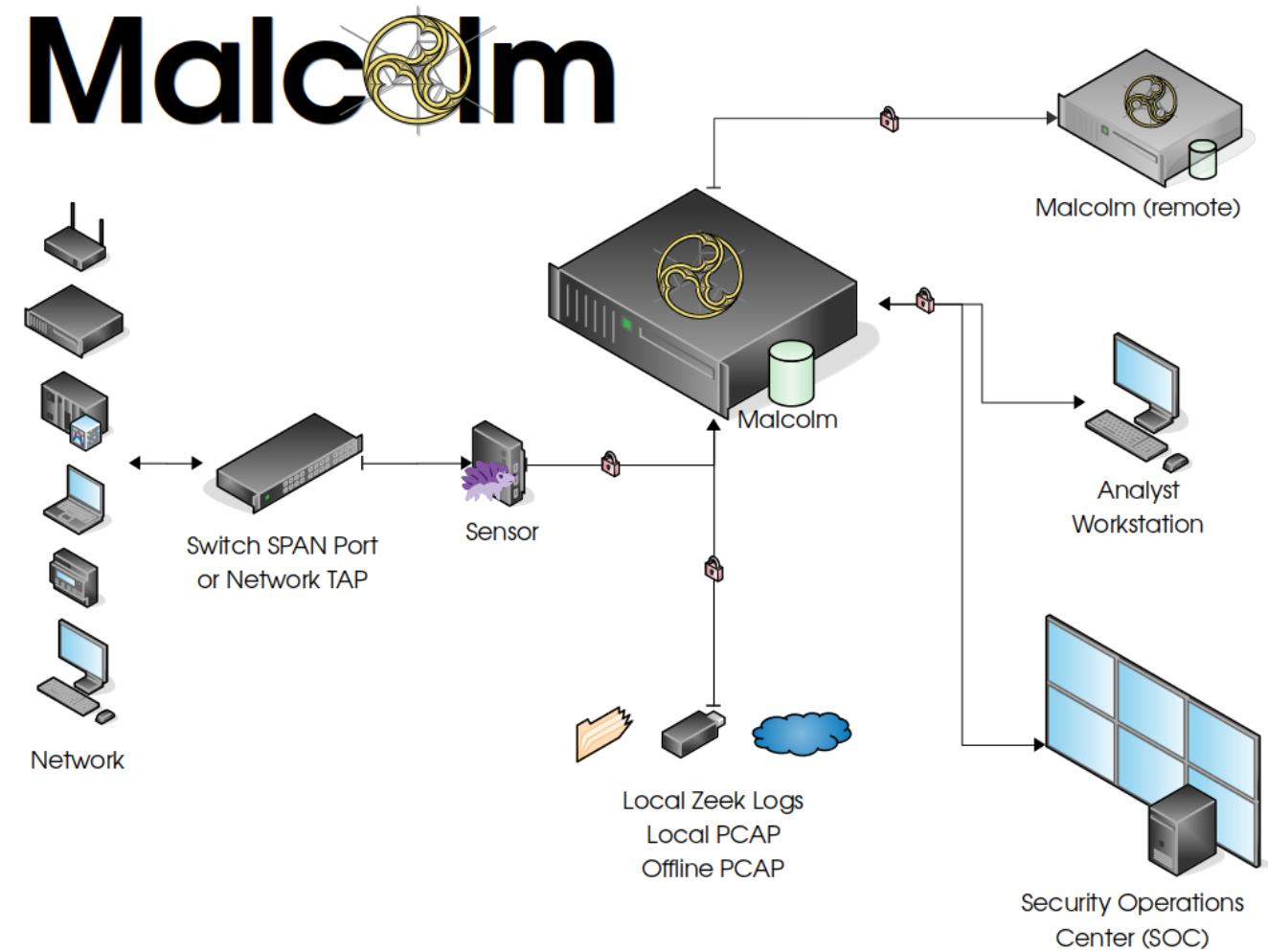
- Default detectors are provided for action and result, flow size and types of transferred files
- Custom detectors may be created for any aspect of any supported protocol

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

- 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 / LDAP

# 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](https://github.com/idaholab/Malcolm)
- Walkthroughs on [YouTube](#): search “Malcolm Network Traffic Analysis”



# Identifying Network Hosts and Subnets

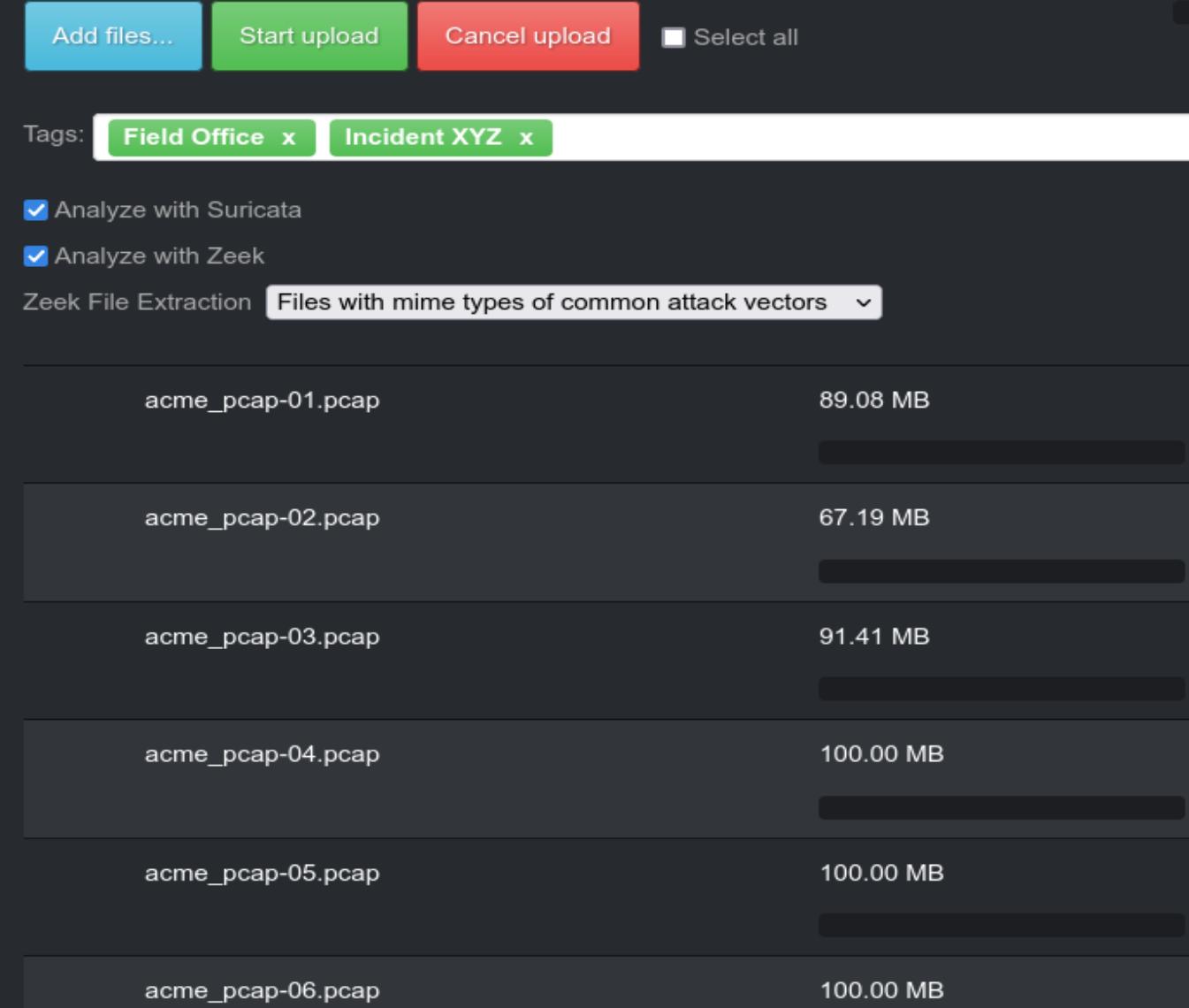
- Assign custom names to network devices and subnets prior to PCAP import
- Allows identification of cross-segment traffic and network log enrichment
- Define in NetBox: <https://localhost/netbox>

The screenshot displays the NetBox web interface, which is a network management tool. On the left, there is a sidebar with navigation links: Organization, Devices, Connections, Wireless, IPAM (selected), IP Addresses, IP Ranges, and PREFIXES (selected). The main content area has two tabs: 'Prefixes' and 'Devices'. The 'Devices' tab is active, showing a table of 32 devices. The table columns include Name, Status, Tenant, Site, Location, Rack, Role, Manufacturer, Type, and IP Address. The 'PREFIXES' tab shows a list of network prefixes with details like Status, Children, VRF, Utilization, Tenant, and Site. A central sidebar provides filters for Devices, Device Roles, Platforms, Virtual Chassis, and Virtual Device Contexts. The top right of the screen shows a search bar, user authentication (guest), and buttons for Add, Import, and Export.

Name	Status	Tenant	Site	Location	Rack	Role	Manufacturer	Type	IP Address
SolarHMI09	Active	—	Cyberville	—	—	HMI	Dell	Precision 3460	192.168.0.128/32
WINDHMI09	Active	—	Cyberville	—	—	HMI	Dell	Precision 3460	10.10.30.130/32
Battery HMI	Active	—	Cyberville	—	—	HMI	Schneider Electric	Unspecified	10.10.10.3/32
Combined Cycle BOP Historian	Active	—	Cyberville	—	—	Historian	Dell	PowerEdge	10.10.20.5/32
Substation Historian	Active	—	Cyberville	—	—	Historian	Unspecified	Unspecified	10.10.100.5/32
Battery Historian	Active	—	Cyberville	—	—	Historian	Dell	PowerEdge	10.10.10.5/32
Solar Panel Historian	Active	—	Cyberville	—	—	Historian	Dell	PowerEdge	192.168.0.5/32
Cellular Modem	Active	—	Cyberville	—	—	Modem	Digi	WR-21	10.10.10.11/32

# 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>



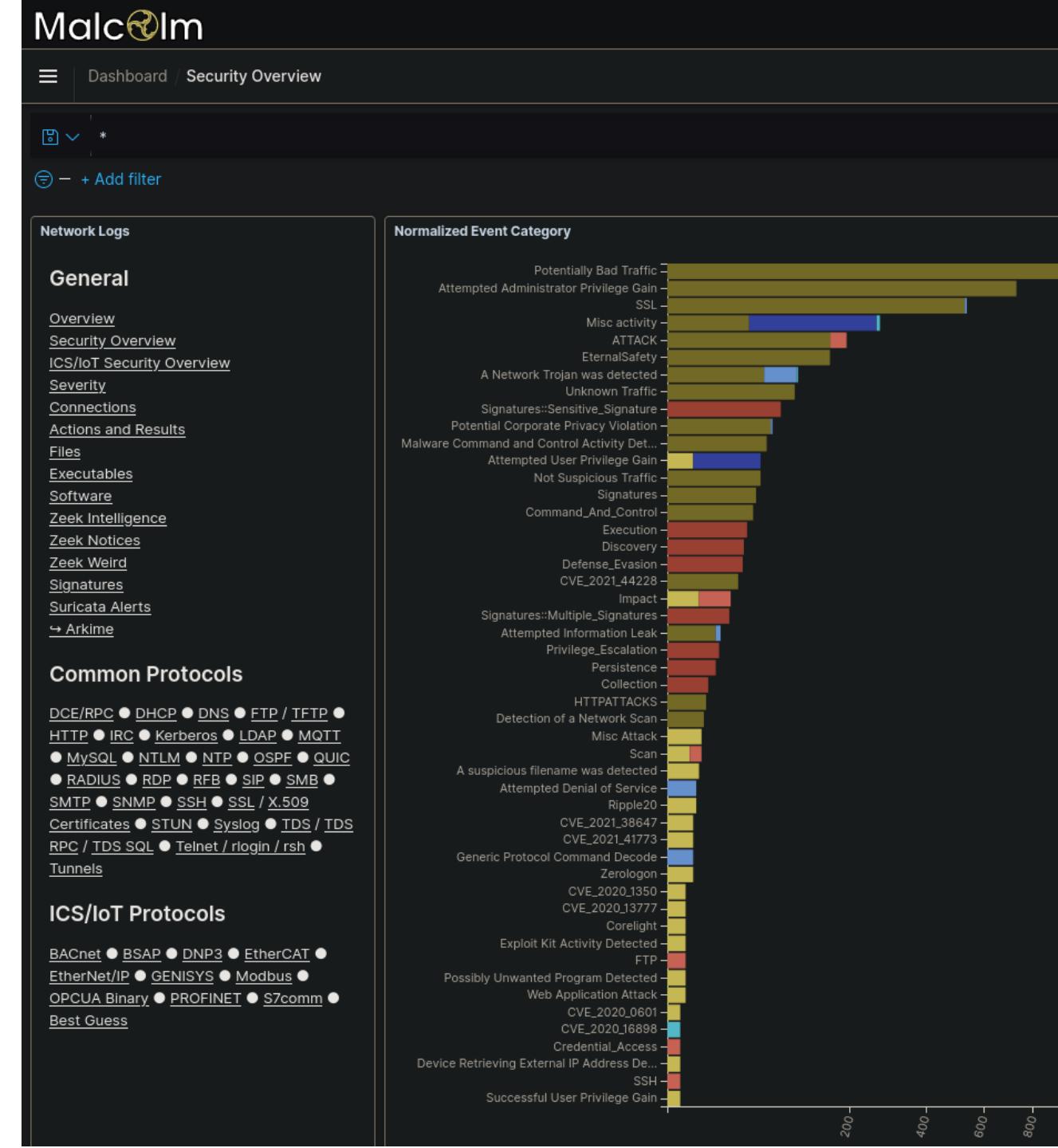
# Data Tagging and Enrichment



- 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 (JA3 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
  - `ics`,  
`ics_best_guess`,  
`cross_segment`,  
etc.

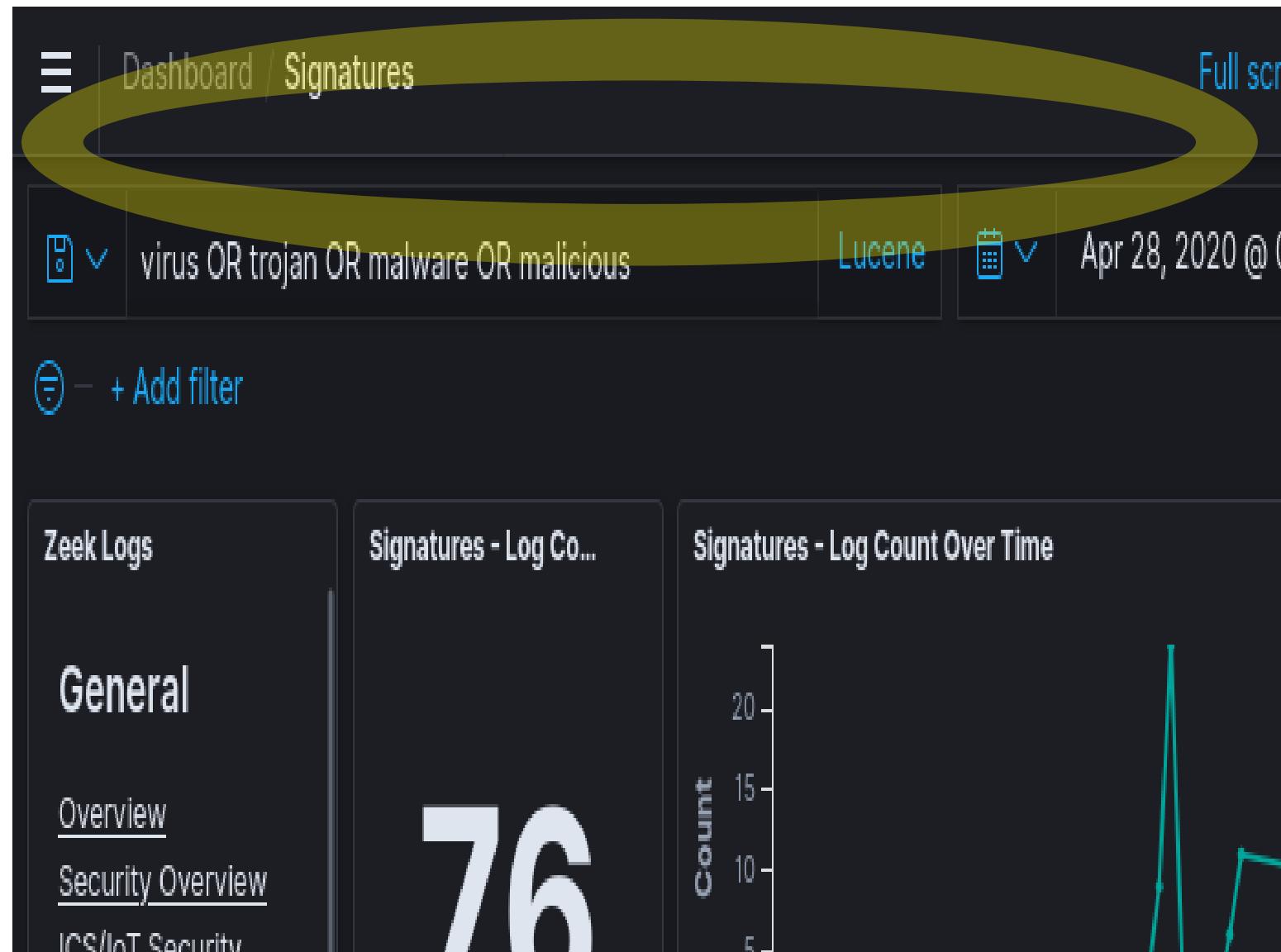
# OpenSearch Dashboards

- 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 / TDSX](#)

## Normalized Event Categories

Po

Attempted Adminis

A Network T

Signatures::

Potential Corpora

Malware Command and C

Attempted

No

Com

Signatures::

Attempted

Detection

A suspicious file

Attempted

# Zeek Notices

- 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

Malcolm

Dashboard / Zeek Notices

+ Add filter

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 RPC / TDS SQL
- Telnet / rlogin / rsh
- Tunnels

ICS/IoT Protocols

Notices - Log Count

749

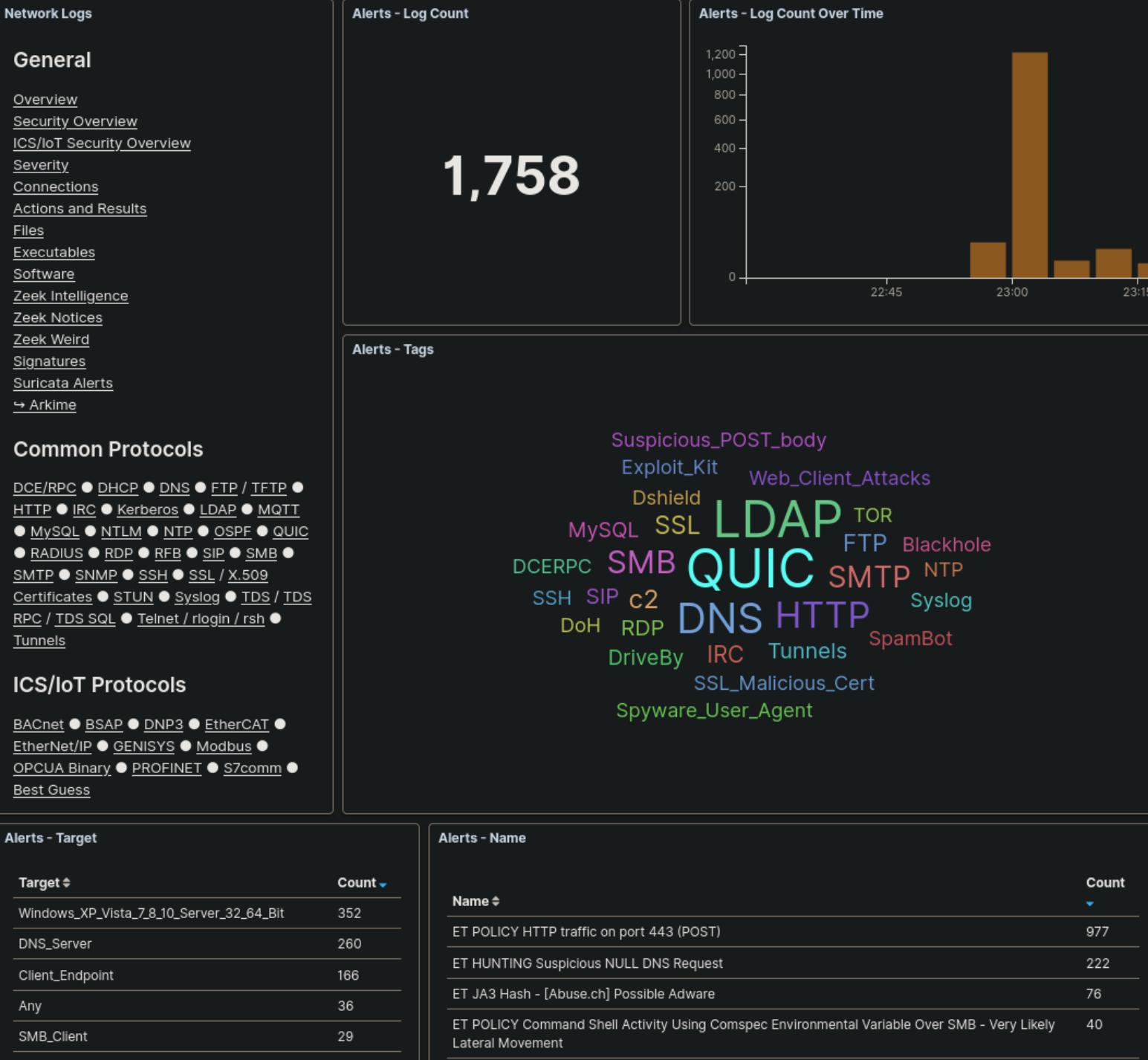
Notices - Log Count Over Time

Notices - Notice Type

Notice Category	Notice Subcategory	Count
SSL	Invalid_Server_Cert	512
ATTACK	Execution	60
ATTACK	Lateral_Movement	39
EternalSafety	ViolationTx2Cmd	28
Signatures	Sensitive_Signature	26
EternalSafety	ViolationNtRename	22
ATTACK	Discovery	15
EternalSafety	EternalBlue	13
EternalSafety	DoublePulsar	10
ATTACK	Lateral_Movement_Multiple_Attempts	6

# 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

**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 • REB • 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 • GENIUS • Modbus • OPCUA Binary • PROFINET • S7comm • Best Guess

**Outdated/Insecure Application Protocols**

Application Protocol	Protocol Version	Count
smb	1	124,835
ftp	-	3,099
tls	TLSv10	422
tls	TLSv11	253
tls	-	239
ntp	3	90
ttcp	-	84

**Normalized Event Category**

**Notice, Alert, Signature and Weird - Summary**

Provider	Dataset	Category	Name
suricata	alert	Potentially Bad Traffic	ET POLICY HTTP traffic on port 443 (POST)
zeek	notice	SSL	Invalid_Server_Cert
suricata	alert	Attempted Administrator Privilege Gain	ET EXPLOIT Possible Zerologon NetServerAuthenticate (CVE-2020-1472)
zeek	weird	-	line_terminated_with_single_CR
zeek	weird	-	NUL_in_line
zeek	weird	-	end-of-data reached before &until expression found (/op:/spicy-lisp/analyzer/lisp.spicy:165:18)
suricata	alert	Misc activity	ET HUNTING Suspicious NULL DNS Request
suricata	alert	Attempted Administrator Privilege Gain	ET EXPLOIT Possible Zerologon Phase 1/3 - NetServerChallenge (CVE-2020-1472)
zeek	weird	-	possible_split_routing
zeek	weird	-	data_before_established
zeek	weird	-	premature_connection_reuse
suricata	alert	Unknown Traffic	ET JA3 Hash - [Abuse.ch] Possible Adware
zeek	weird	-	-
zeek	notice	ATT	-
zeek	notice	Eternet	-
suricata	alert	Atten Gain	-
zeek	notice	Sign	-
zeek	weird	-	-
suricata	alert	Poter	-

**Malcolm**

**Dashboard | ICS/IoT Security Overview**

**Full screen Share Clone Reporting**

**Zeek Logs**

**ICS/IoT Log Counts**

- General**
- Overview
- Security Overview
- ICS/IoT Security Overview
- Severity
- Connections
- Actions and Results
- Files
- Executables
- Software
- Notices
- Weird
- Signatures
- Intel Feeds
- Arkime

**Common Protocols**

- DCE/RPC • DHCP • DNS • FTP / TFTP • HTTP • IRC • Kerberos • LDAP • MQTT • MySQL • NTLM • NTP • OSPF • QUIC • RADIUS • RDP • REB • SIP • SMB • SMTP • SNMP • SSH • SSL / X.509 Certificates • STUN • Syslog • TDS / TDS-RPC / TDS-SQL • Telnet / rlogin / rsh • Tunnels

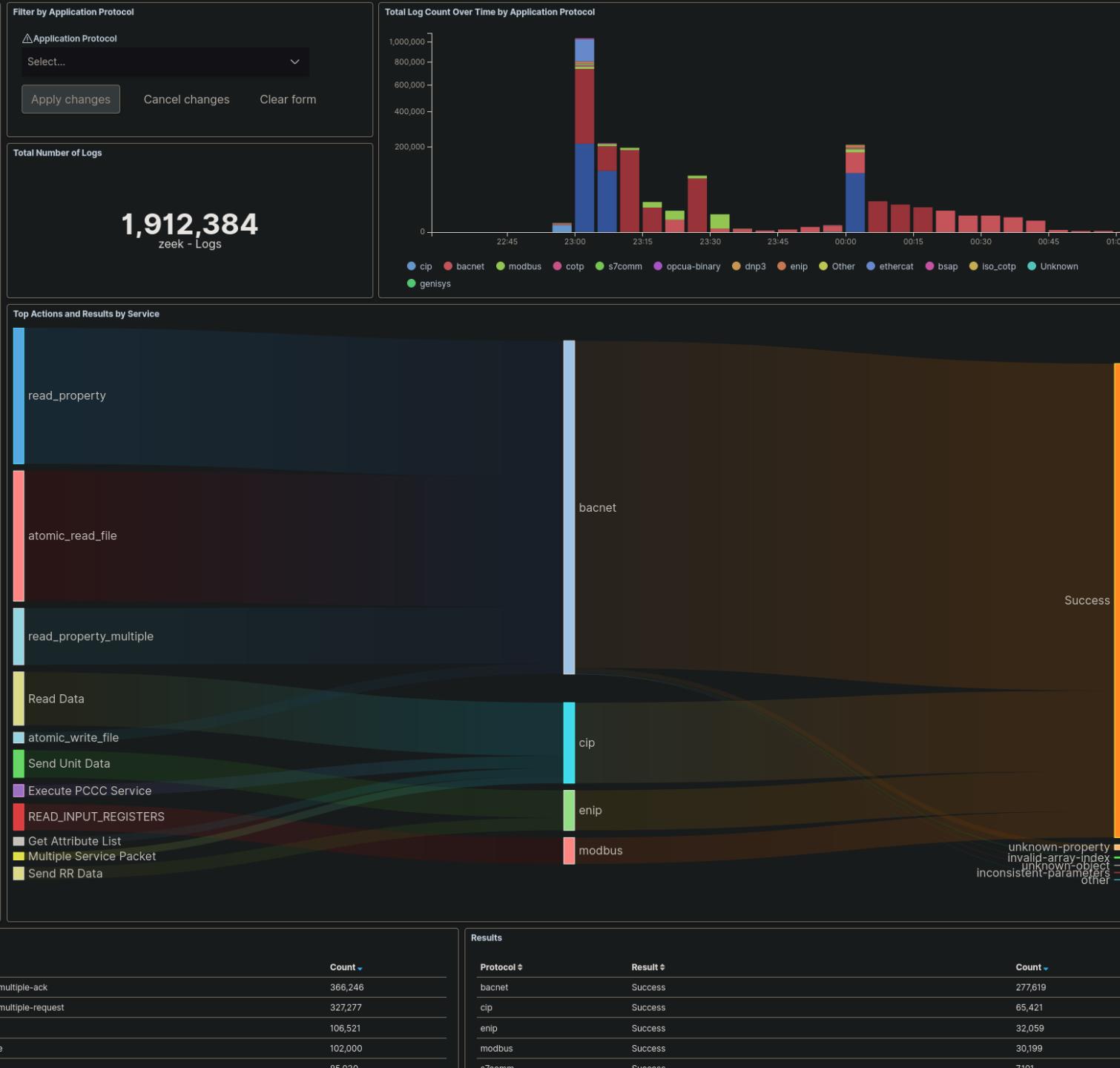
**ICS/IoT Protocols**

- BACnet • BSAP • DNP3 • EtherCAT • EtherNet/IP • Modbus • PROFINET • S7comm • Best Guess

**ICS/IoT Traffic Over Time**

**ICS/IoT External Traffic**

Protocol	Source IP	Source Country	Destination IP	Destination Country	Count
cpdp	134.249.62.202	Ukraine	134.249.61.182	Ukraine	679
s7comm	134.249.62.202	Ukraine	134.249.61.182	Ukraine	411
modbus	118.189.96.132	Singapore	118.189.96.132	Singapore	32
modbus	192.168.66.235	-	166.161.16.230	United States	15
s7comm	134.249.62.206	Ukraine	134.249.61.183	Ukraine	5



# 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
  - Synchrophasor (IEEE-C37.118)

[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

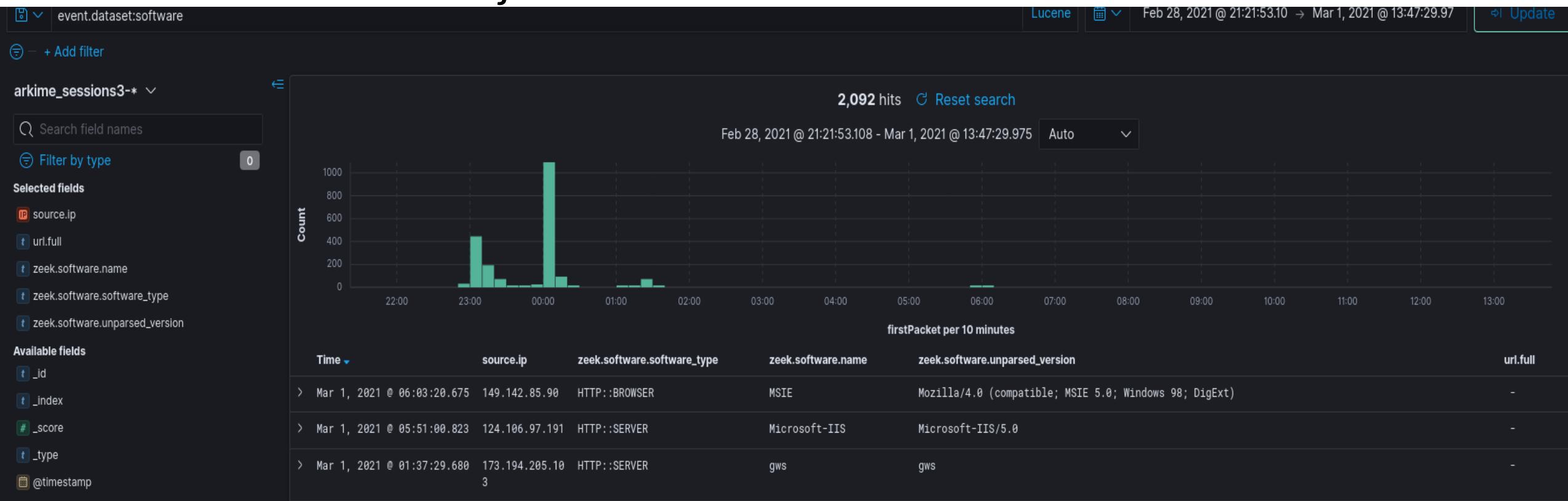
EternalSafety

ATTACK

Export: Raw  For

# 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

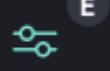


## New Visualization

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

	<b>Arkime</b>	<b>Dashboards (Lucene)</b>	<b>Dashboards (DQL)</b>
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 &gt;= 200 &amp;&amp; http.statuscode &lt;= 300</code>	<code>http.statuscode:[200 TO 300]</code>	<code>http.statuscode &gt;= 200 and http.statuscode &lt;= 300</code>

# Search Syntax Comparison (cont.)

	Arkime	Dashboards (Lucene)	Dashboards (DQL)
Field range (exclusive)	<code>http.statuscode &gt; 200 &amp;&amp; http.statuscode &lt; 300</code>	<code>http.statuscode:{200 TO 300}</code>	<code>http.statuscode &gt; 200 and http.statuscode &lt; 300</code>
Field range (mixed exclusivity)	<code>http.statuscode &gt;= 200 &amp;&amp; http.statuscode &lt; 300</code>	<code>http.statuscode:[200 TO 300}</code>	<code>http.statuscode &gt;= 200 and http.statuscode &lt; 300</code>
Match all search terms (AND)	<code>(tags == [external_source, external_destination]) &amp;&amp; (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)
GeolP information available	country == EXISTS!	_exists_:destination.geo OR _exists_:source.geo	destination.geo:* or source.geo:*
Log type	event.dataset == notice	event.dataset:notice	event.dataset:notice
IP CIDR Subnets	ip.src == 172.16.0.0/12	source.ip:"172.16.0.0/12"	source.ip:"172.16.0.0/12"
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
GeolP information available	country == EXISTS!	_exists_:destination.geo OR _exists_:source.geo	destination.geo:* or source.geo:*

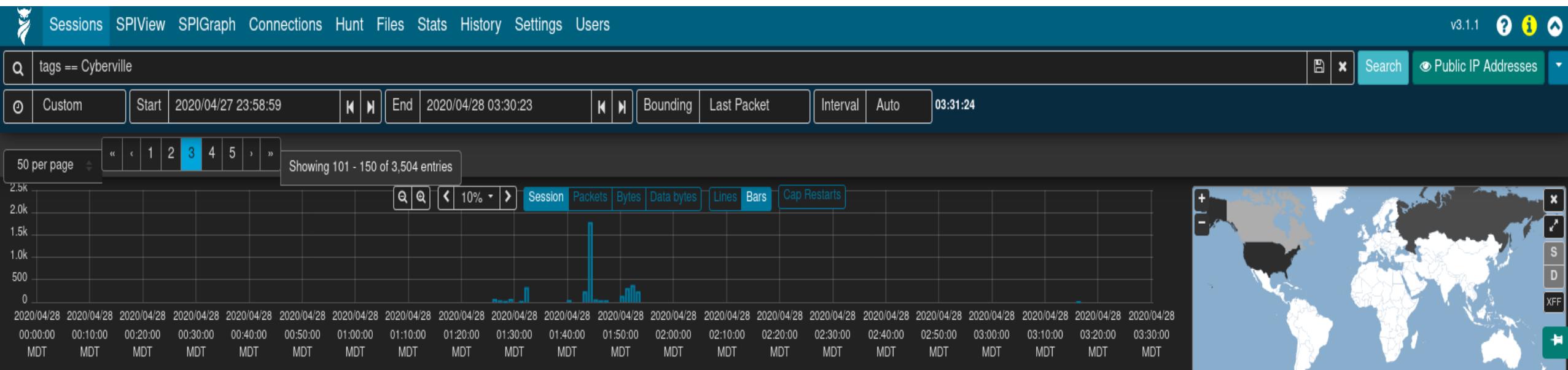
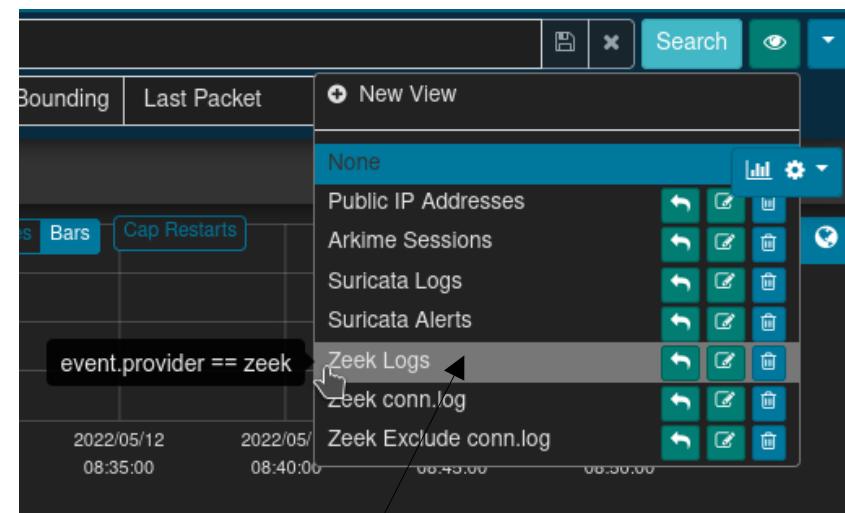


# Arkime

- 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>

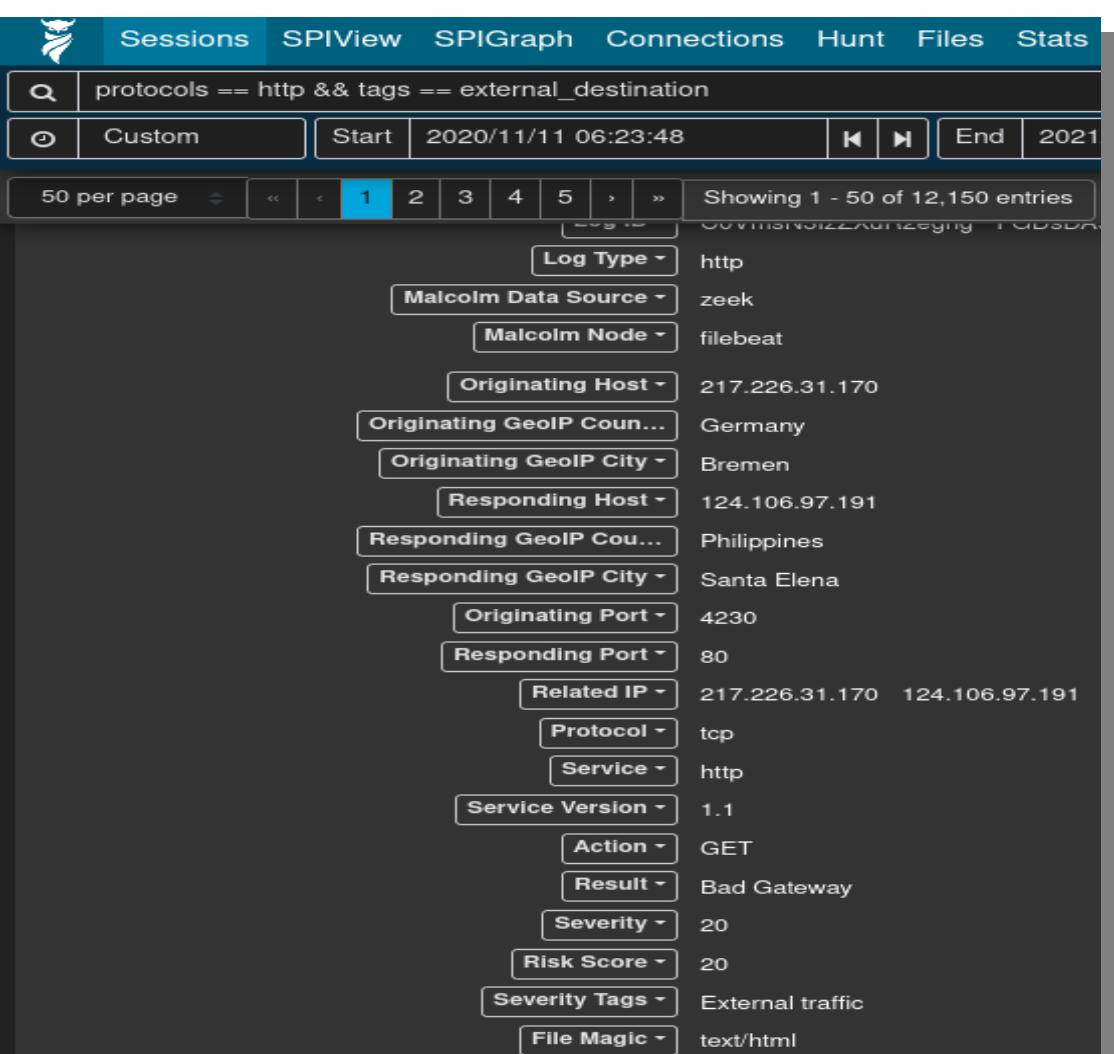
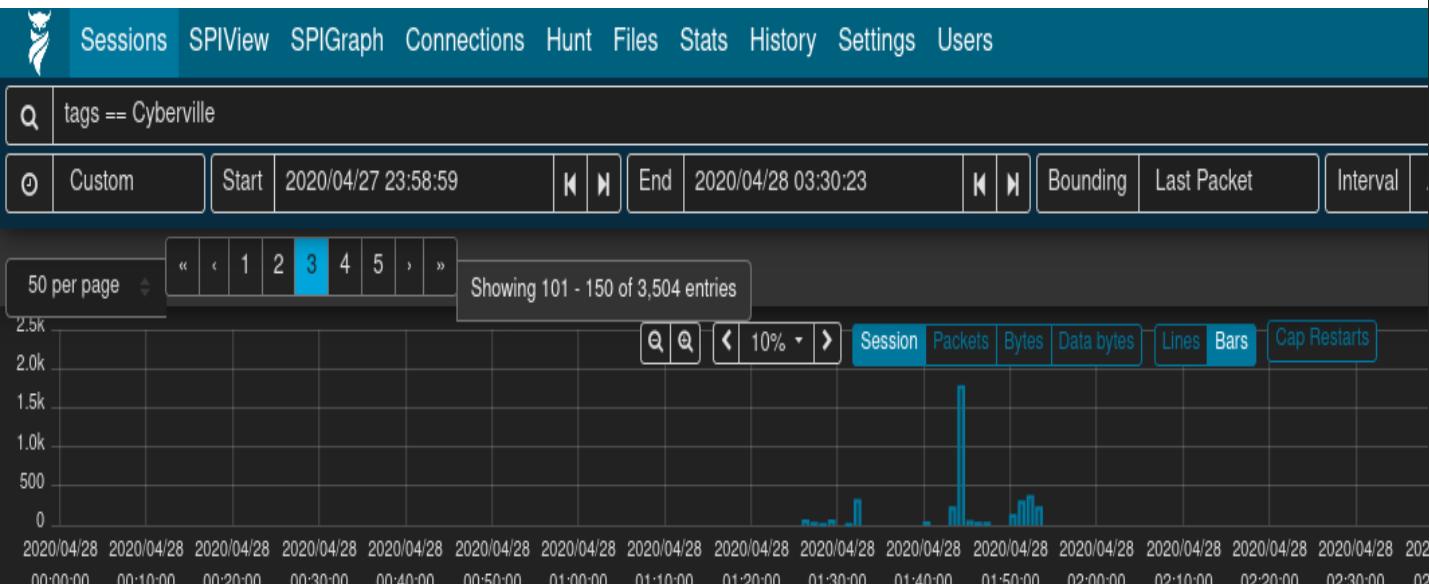
# Arkime Filters and Search

- 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



Zeek http.log

# 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

```
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
```

## Destination

```
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](#)

# 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 shows the Arkime application interface. At the top, there's a navigation bar with links: Sessions, SPIView, SPIGraph, Connections, Hunt, Files, Stats, History, Settings, and Users. On the far right of the bar are icons for help, information, and refresh. Below the navigation bar is a search bar containing the query "country != US & protocols == http". To the right of the search bar are buttons for "Search" and "Arkime Sessions". Underneath the search bar are filters for "Custom" start and end times (set to 2021/02/28 23:59:11 and 2021/03/01 00:28:26), "Bounding", "Last Packet", and "Interval" (set to "Auto"). The total duration shown is "00:29:15". Below these filters are three tabs: "Open Items", "Visible Items" (which is selected), and "Matching Items". To the right of these tabs are buttons for "Include same time period" and "linked segments (slow)". A "Filename" field is set to "US\_HTTP.pcap". On the far right of this section is a button labeled "Export PCAP" with a count of "0". At the bottom left, there's a pagination control showing "Showing 1 - 50 of 120 entries" and a "50 per page" dropdown. In the center, there's a timeline chart showing packet activity over time. At the top of the chart are search and zoom controls, followed by a legend for Session (selected), Packets, Bytes, Data bytes, Lines, Bars, and Cap Restarts. On the right side of the interface is a world map with various regions highlighted in different shades of gray.

# 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  $\leq 1$  week before using (it runs many queries)



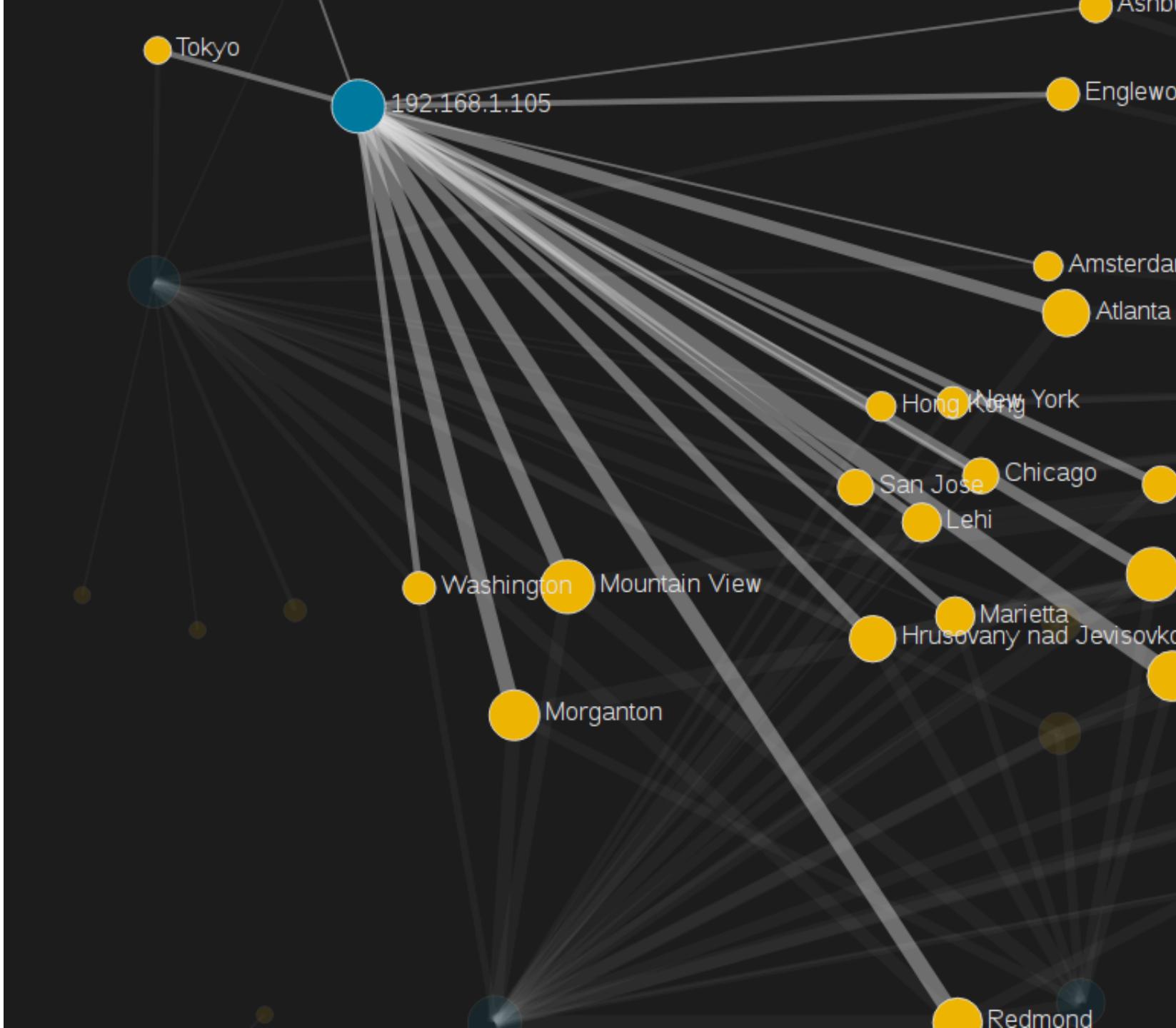
# SPIGraph

- 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

Sessions SPIView SPIGraph Connections Hunt Files Stats History Settings Users v3.1.1 ? ! 🔍

protocols == http Search Arkime Sessions

All (careful) Start 1969/12/31 17:00:00 End 2021/12/06 12:10:02 Bounding Last Packet

Creating a new packet search job will search the packets of 2,906 sessions. Create a packet search job

### Hunt Job History

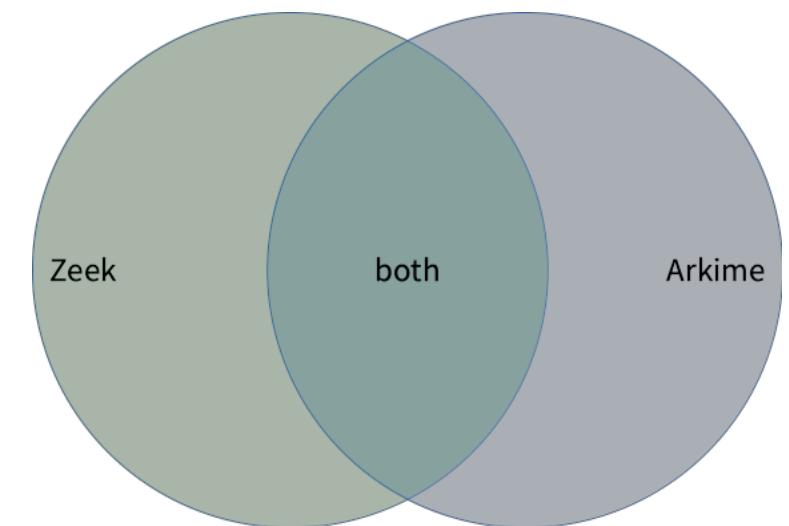
Search your packet search job history 50 per page 1 Showing 1 - 1 of 1 entries

Status	Matches	Name	User	Search text	Notify	Created	ID
<span style="color: green;">✓</span> 100%	141	HTTP with password		password (ascii)		2021/12/06 12:12:27 MST	s5YpkX0BTA40FhD4X7dA

This hunt is **finished**  
Found 141 sessions matching **password** (ascii) of 2,908 sessions searched  
Created: 2021/12/06 12:12:27 MST  
Last Updated: 2021/12/06 12:12:32 MST  
Examining 500 raw source and destination packets per session  
The sessions query expression was: **protocols == http**  
The sessions query view was: **Arkime Sessions**  
The sessions query time range was from 1969/12/31 17:00:00 MST to 2021/12/06 12:10:02 MST

# Data Source Correlation

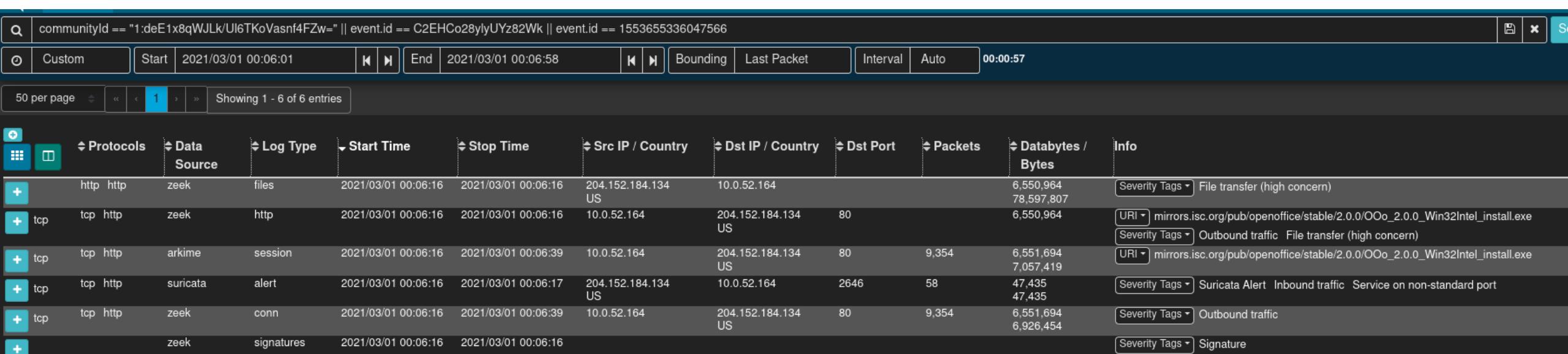
- Search syntax is different between Arkime and Dashboards (and in some cases, so are field names)
  - See search syntax comparison table, Malcolm and Arkime docs
- Despite considerable overlap, there are differences in protocol parser support among Zeek, Suricata and Arkime
  - Learning the strengths 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:r7tGG//fXP1P0+BXH3zXETCtEFI=" || event.id == "CQcoro2z6adgtGlk42"
```



The screenshot shows the Arkime interface with a search bar at the top containing the query: `communityId == "1:r7tGG//fXP1P0+BXH3zXETCtEFI=" || event.id == "CQcoro2z6adgtGlk42"`. Below the search bar are various filtering and timeline controls. The main area displays a table of log entries. The table has columns for Protocols, Data Source, Log Type, Start Time, Stop Time, Src IP / Country, Dst IP / Country, Dst Port, Packets, Databytes / Bytes, and Info. There are five rows of data:

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 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 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 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 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

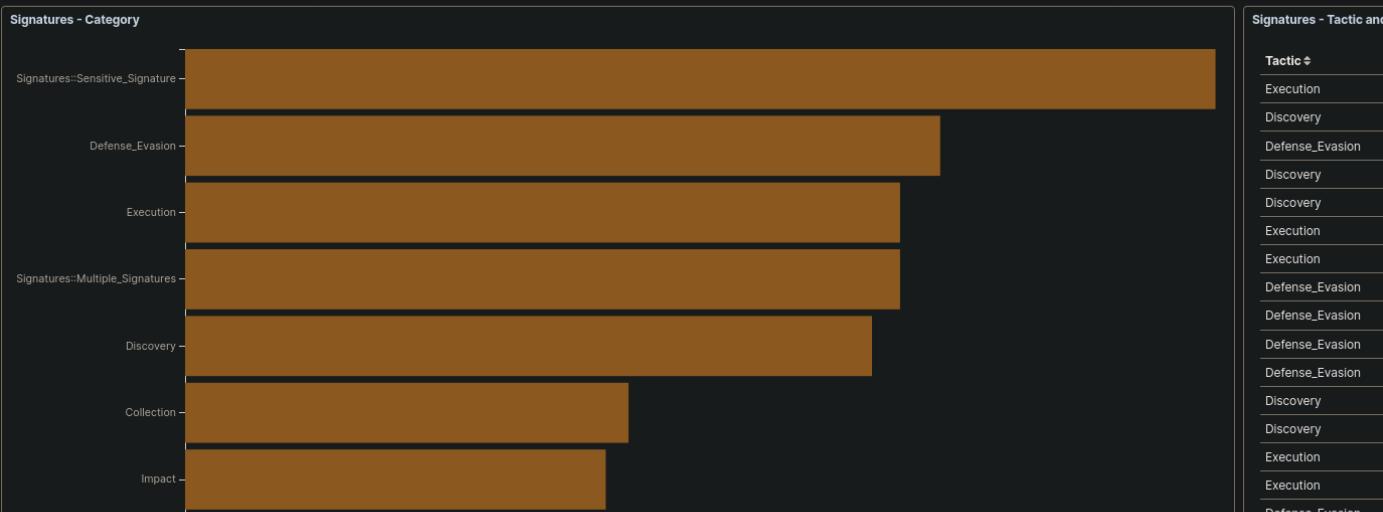
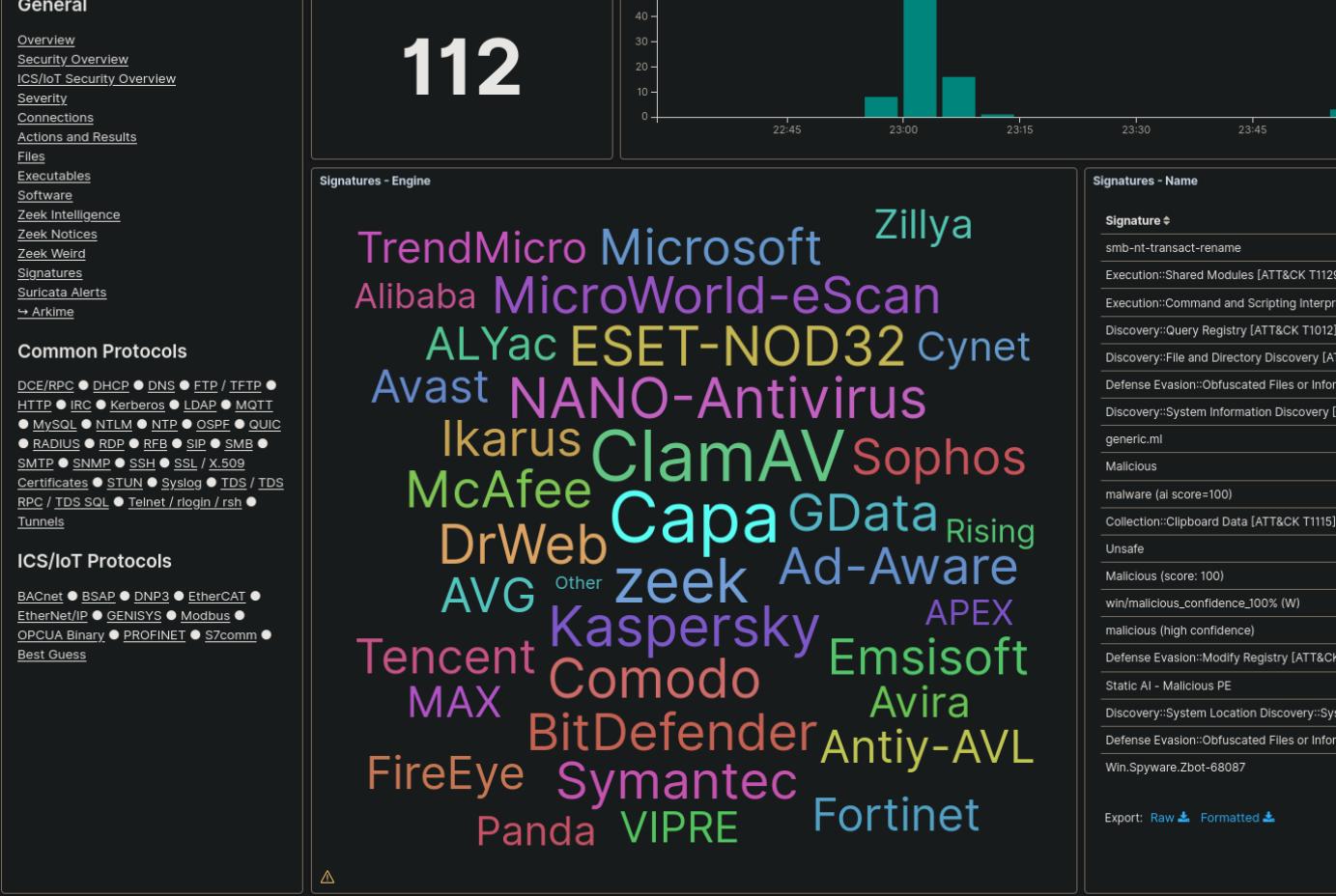


- Zeek can “carve” file transfers from common protocols
- Malcolm can examine carved files and flag hits
  - ClamAV - open source antivirus engine
  - YARA - pattern matching swiss army knife
  - Capa - portable executable capabilities analyzer
  - VirusTotal - online database of file hashes
    - requires 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

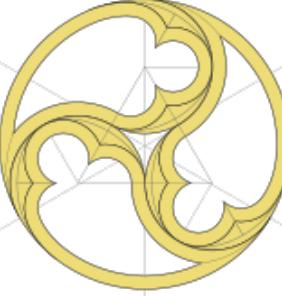
- Signatures dashboard in Dashboards shows scanned file hits
- Use `zeek.fuid` field in *Signatures - Logs* table to pivot to connection UID (`zeek.uid`) and other logs with pertinent session details



# 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

# Malcolm



## Thank you!

Visit [Malcolm on GitHub](#) to read the docs, make suggestions, report issues and st★r to show your support!

Malcolm is Copyright © 2022 Battelle Energy Alliance, LLC, and is developed and released as open-source software through the cooperation of the Cybersecurity and Infrastructure Security Agency of the US Department of Homeland Security.