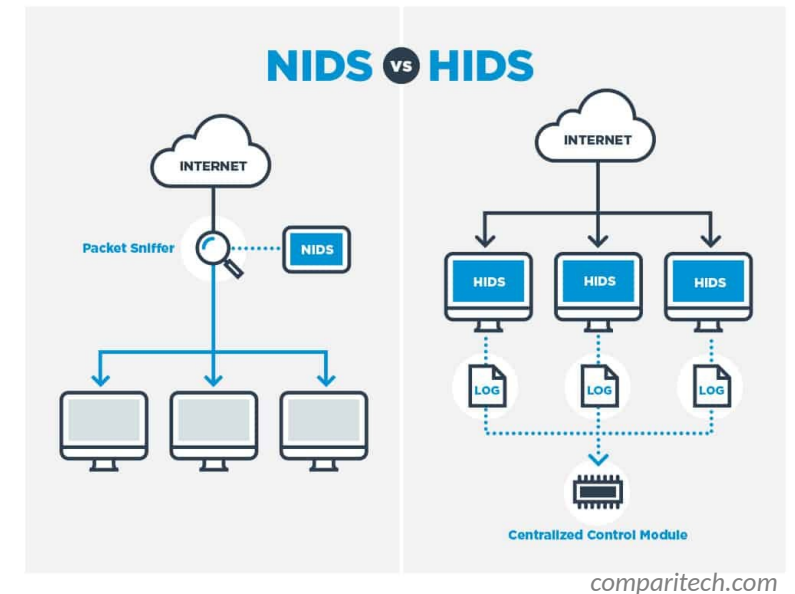


# Network Traffic Analysis with Malcolm

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	line	Destination of the first packet
uid	string	Unique ID of the connection
src_ip.h	addr	Originating endpoint IP address (big)
src_ip.p	port	Originating endpoint TCP/UDP port (or ICMP code)
dst_ip.h	addr	Receiving endpoint IP address (host)
dst_ip.p	port	Receiving endpoint TCP/UDP port (or ICMP code)
proto	proto	Transport layer protocol of connection
service	string	Detected application protocol, if any
duration	interval	Connection origin
req_bytes	count	Req. payload bytes from requests (includes HTTP)
resp_bytes	count	Resp. payload bytes from responses (includes HTTP)
conn_state	string	Connection state (see connlog, conn_state)
last_req	time	Last Req in this conn. (req?)
last_resp	time	Last Resp in this conn. (resp?)
closed_bytes	count	Number of bytes closed due to connection gen.
history	string	Connection state history (see connlog, "history")
req_pkts	count	Number of Req packets
req_ip_bytes	count	Number of Req IP bytes (see IP stat, req_ip_bytes)
resp_pkts	count	Number of Resp packets
resp_ip_bytes	count	Number of Resp IP bytes (see IP stat, resp_ip_bytes)
failed_packets	int	# of corrupted connection UDP or incoming connection
req_L2_addr	string	Link layer address of the originator
resp_L2_addr	string	Link layer address of the responder
link	int	The value L2_AK for this connection
peer_addr	int	The peer L2_AK for this connection

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

FIELD	TYPE	DESCRIPTION
to	line	Destination of the HTTP request
url & url	string	Underlying connection url, see connlog
trans_depth	count	Expanded depth into the connection
method	string	HTTP Request verb (GET, POST, HEAD, etc)
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
request_body_len	count	Uncompressed content size of Req data
response_body_len	count	Uncompressed content size of Resp data
status_code	count	Status code returned by the server
status_msg	string	Status message returned by the server
info_code	count	Last seen flow info reply code by server
info_msg	string	Last seen flow info message by server
flag	int	Indicators of various attributes discovered
streamsize	string	User name if gzip auth is performed
password	string	Password if gzip auth is performed
proceed	int	Headers indicator of a proceed request
req_host	vector	The unique Host:Req string
req_host_bytes	vector	The bytes from Req
req_host	vector	The unique Host:Req bytes
resp_host_bytes	vector	The bytes from Resp
resp_host	vector	The unique Host:Resp bytes
client_header	vector	The values of HTTP headers sent by Req
server_header	vector	The values of HTTP headers sent by Resp
cookie_name	vector	Variable names extracted from cookies
set_cookie	vector	Variable names extracted from the URI

*# of ports in req\_header, resp\_header is loaded  
# of ports in req\_header, resp\_header is loaded*

### files.log | File analysis results

FIELD	TYPE	DESCRIPTION
to	line	Destination where the file originated
file	string	Unique identifier for original file
to_host	addr	Host that received the data
to_host_ip	addr	IP address of the host that received the data
conn_addr	addr	Connection address where the transfered source
conn_src	string	An identification of the source of the file data
depth	count	Depth of the request to source (e.g. HTTP request depth)
analysis	int	# of analysis attached during file analysis
mime_type	string	The type determined by file signatures
signature	string	File name, filename, extension, etc
duration	interval	The duration that the file was analyzed
local_size	count	Size of the data in original body?
is_analyzed	bool	Was the file sent to the engine?
req_bytes	count	Number of bytes provided to the analysis engine
total_bytes	count	Total number of bytes that should comprise the file
missing_bytes	count	Number of bytes in the stream missed
overflow_bytes	count	Out of response bytes in the stream due to overflow
truncated	bool	If the file analysis timed out at user error
parent_file	string	Container the ID this was extracted from
url_fragment	string	URL/URI path of the file
extension	string	Last filename of extracted file, if filename
entropy	float	Information density of the file contents

### pe.log | Portable Executable (PE)

FIELD	TYPE	DESCRIPTION
to	line	Current connection
file	string	The full file path name including the filename
hashcode	string	The larger hex code that the file was assigned to
is_analyzed	bool	The file that the file was analyzed at
is_analyzed_time	string	The required analyzing system
signature	string	The signature that is required to run the file
is_dll	bool	Is the file a DLL or not?
is_exe	bool	Is the file an EXE or not?
is_dll_exe	bool	Is the file a DLL or EXE or not?
is_dll_exe	bool	Does the file support Address Space Layout Randomization?
is_dll_exe	bool	Does the file support Data Execution Prevention?
is_dll_exe	bool	Does the file have code integrity metadata?
is_dll_exe	bool	Does the file have an optional header?
is_dll_exe	bool	Does the file have a resource table?
is_dll_exe	bool	Does the file have an import table?
is_dll_exe	bool	Does the file have an export table?
is_dll_exe	bool	Does the file have an embedded resource table?
is_dll_exe	bool	Does the file have a debug info table?
is_dll_exe	bool	Does the file have a section?

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.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.

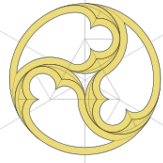
## Expanding control systems visibility

- 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>



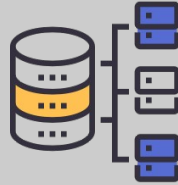
Capture & Analysis



File Scanning



Forwarding & Enrichment



Storage



Anomaly Detection



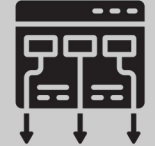
Alerting



Visualization



Payload Analysis



Framework



zeek

yara



beats



OpenSearch



OpenSearch Anomaly Detection Plugin



OpenSearch Alerting Plugin



OpenSearch Dashboards



CyberChef



docker



SURICATA

CAPA



logstash



netsniff-ng

VIRUSTOTAL



Arkime

Arkime session PCAP export to

WIRESHARK

TCPDUMP



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

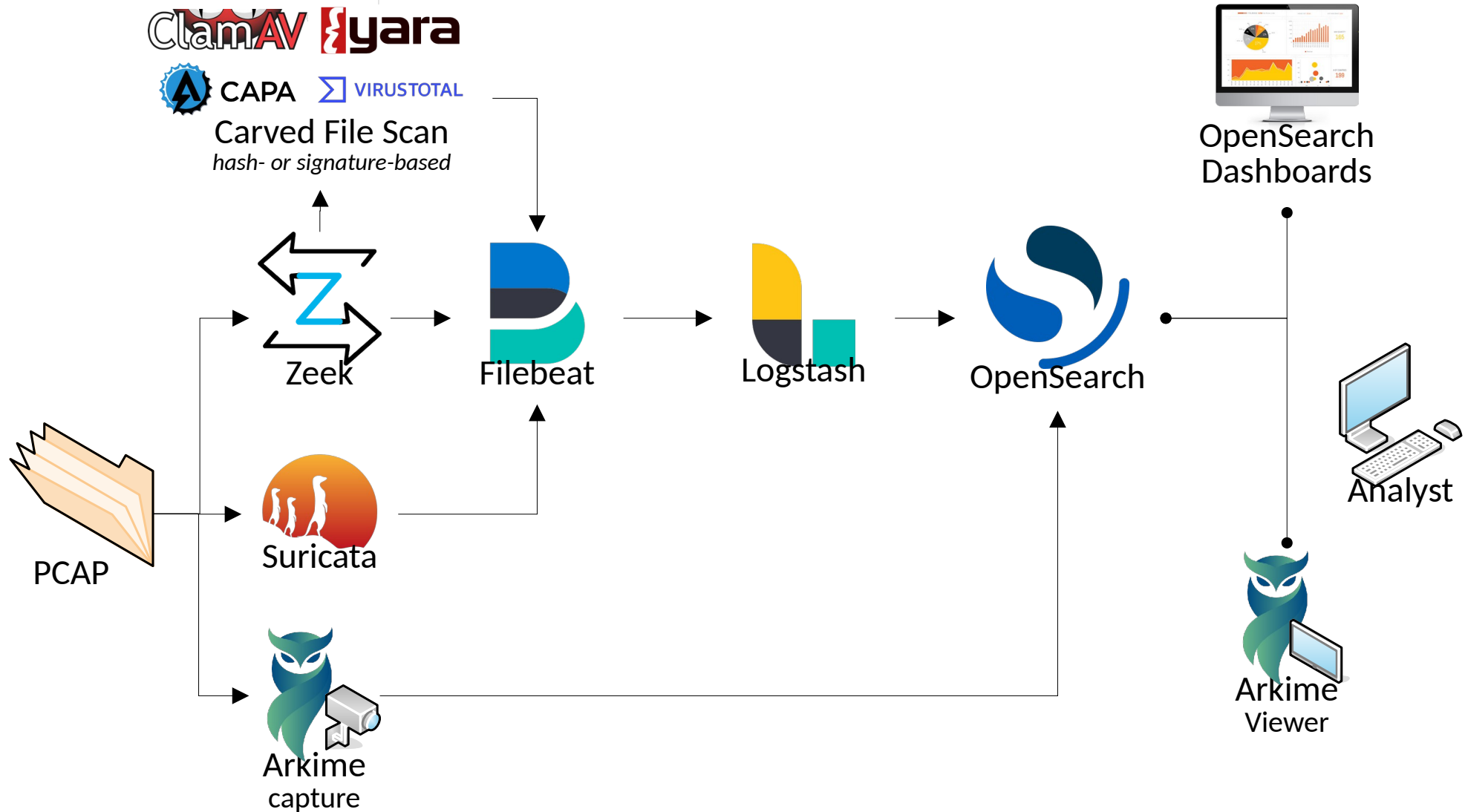
various tunnel protocols (e.g., GTP, GRE, Teredo, AYIYA, IP-in-IP, etc.)

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

# Malcolm

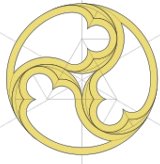
## Data Pipeline

<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 dedicated sensor

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-tiered Malcolm instance

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

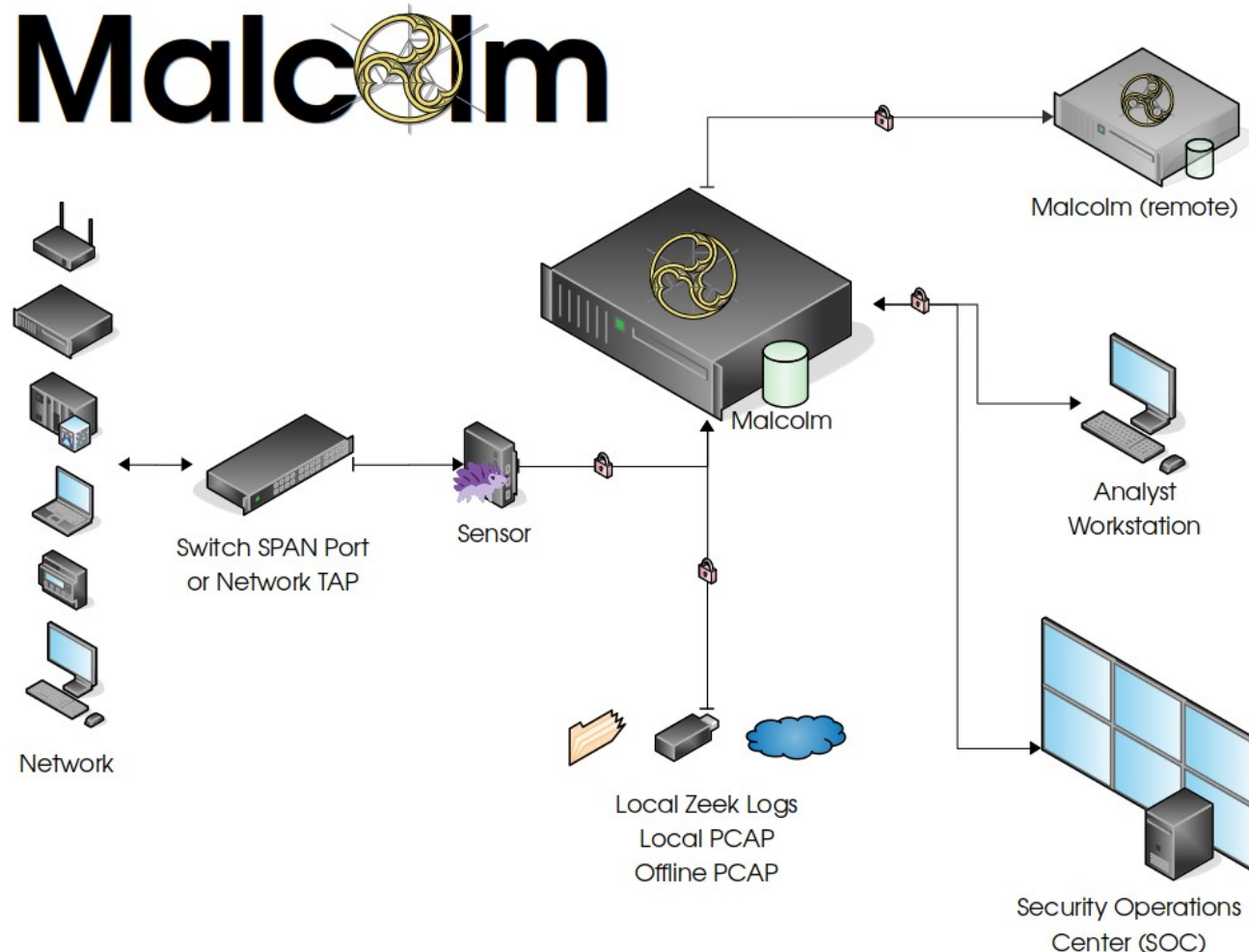
- Alerts may be triggered by exceeded thresholds, anomalies detected, custom queries, etc.

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 / 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 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/name-map-ui>



The screenshot shows a web interface for mapping network addresses to names and tags. The interface includes a search bar at the top right labeled 'Search mappings'. Below it is a table with columns for 'Address', 'Name', and 'Tag'. Each row contains a network address, a human-readable name, and an optional tag. To the right of each row are two icons: a pencil for editing and a red circle with a slash for deleting. At the bottom of the table, there are input fields for 'Address', 'Name', and 'Tag (optional)', along with a save icon.

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

# 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>

The screenshot shows the MalcolM web interface for uploading traffic captures. The interface is dark-themed and features the MalcolM logo at the top right, which includes a globe icon. Below the logo, the text "Capture File and Log Archive Upload" is displayed. The main area contains several interactive elements: a row of buttons for "Add files...", "Start upload", and "Cancel upload", along with a "Select all" checkbox. Below this, there is a "Tags" section with two active tags: "Field Office" and "Incident XYZ". There are also two checked checkboxes for "Analyze with Suricata" and "Analyze with Zeek". A dropdown menu for "Zeek File Extraction" is set to "Files with mime types of common attack vectors". The bottom section of the interface displays a list of six PCAP files, each with its name, size, and a "Start" button. The files are: acme\_pcap-01.pcap (89.08 MB), acme\_pcap-02.pcap (67.19 MB), acme\_pcap-03.pcap (91.41 MB), acme\_pcap-04.pcap (100.00 MB), acme\_pcap-05.pcap (100.00 MB), and acme\_pcap-06.pcap (100.00 MB).

File Name	Size	Action
acme_pcap-01.pcap	89.08 MB	Start
acme_pcap-02.pcap	67.19 MB	Start
acme_pcap-03.pcap	91.41 MB	Start
acme_pcap-04.pcap	100.00 MB	Start
acme_pcap-05.pcap	100.00 MB	Start
acme_pcap-06.pcap	100.00 MB	Start

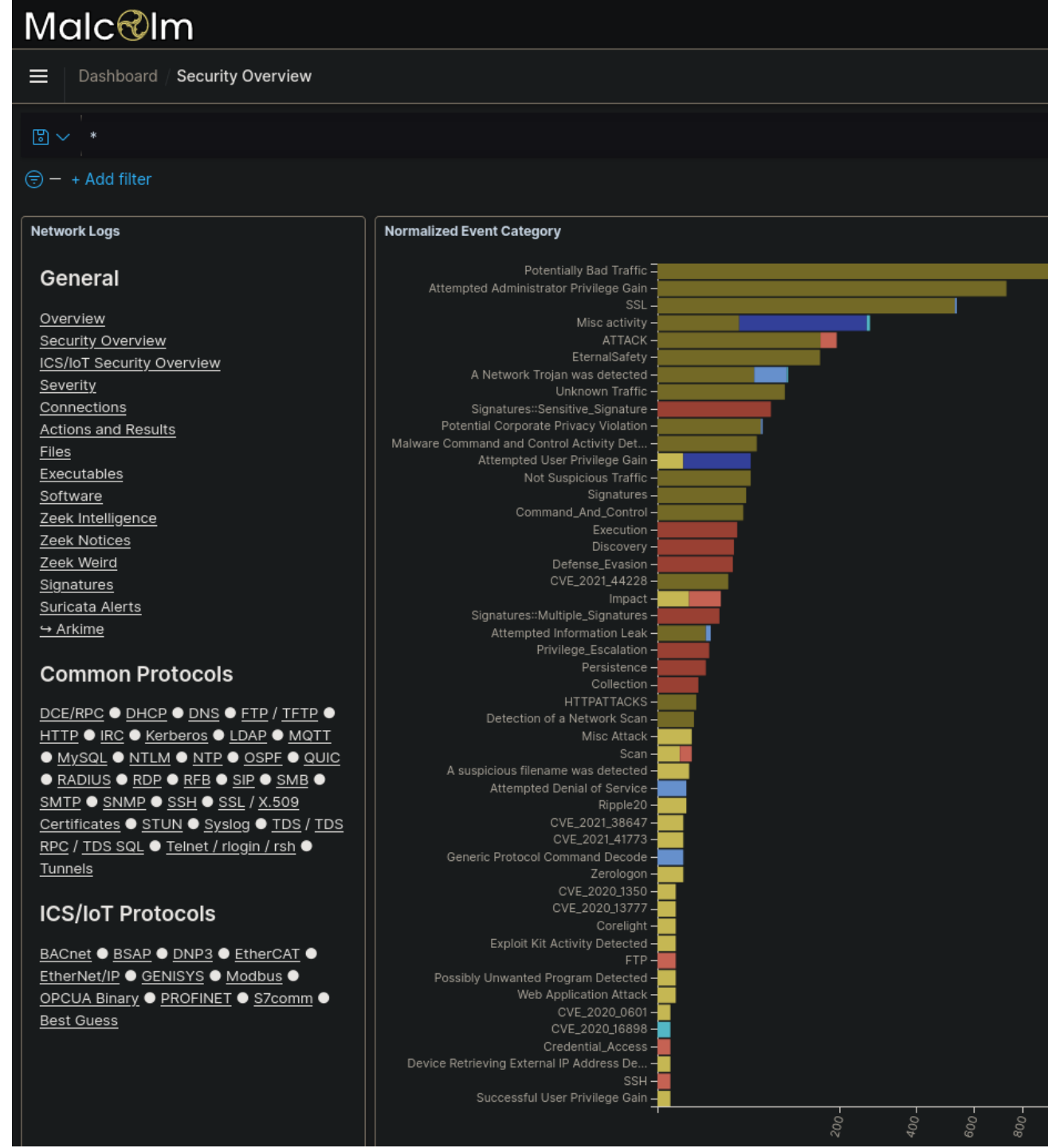
# Data Tagging and Enrichment



- 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
  - `internal_source`,  
`internal_destination`,  
`external_source`,  
`external_destination`,  
`cross_segment`

# 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 / TDS](#)

## Normalized Event Categories

Po  
Attempted Administr

A Network T

Signatures::  
Potential Corpora  
Malware Command and C  
Attempted  
No

Com

Signatures::  
Attempte

Detection

A suspicious file  
Attempte



# 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

BACnet ● BACnet/IP ● BACnet/Modbus ● EtherCAT ●

### Notices - Log Count

749

### Notices - Log Count Over Time

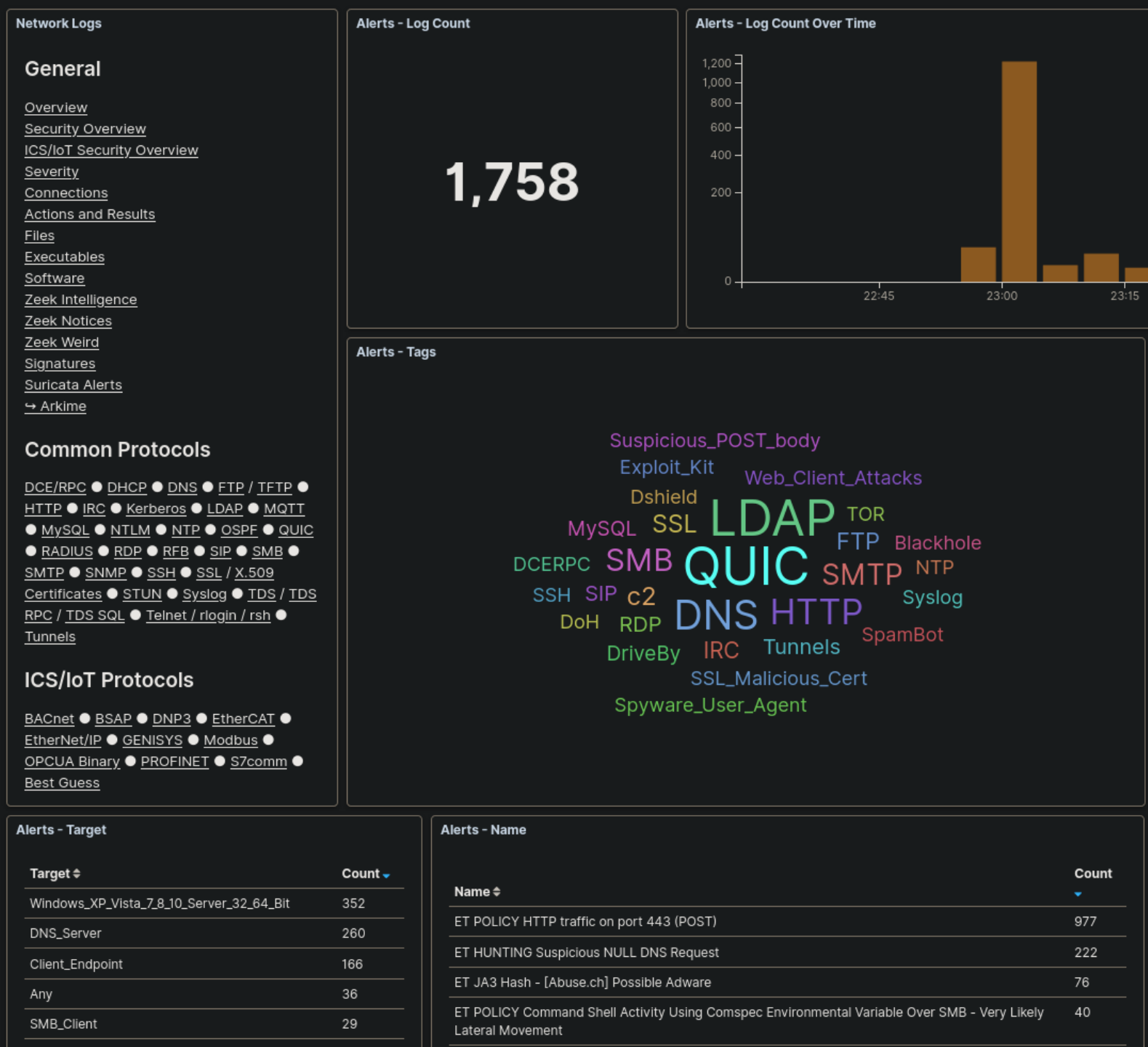
Time	Count
22:45	15
23:00	120

### 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
- Arkim

**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 ● IDS / IDS RPC / IDS SQL ● Telnet / rlogin / rsh ● Tunnels

**ICS/IoT Protocols**

BACnet ● BSAP ● DNP3 ● EtherCAT ● EtherNet/IP ● GENISYS ● Modbus ● OPCUA Binary ● PROFINET ● S7comm ● Best Guess

**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 (/spicy-ldap/analyzer/ldap.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_spill_routing
zeek	weird	-	data_before_established
zeek	weird	-	premature_connection_reuse
zeek	notice	ATTACK	Execution
suricata	alert	Unknown Traffic	ET JA3 Hash - [Abuse.ch] Possible Adware
zeek	weird	-	-
zeek	notice	ATTA	-
zeek	notice	Etern	-
suricata	alert	Atten Gain	-
zeek	notice	Sign	-
zeek	weird	-	-
suricata	alert	Poter	-

**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

**Vulnerabilities**

Data Source	Log Type	Vulnerability ID	Last Seen
zeek	notice	CVE_2021_44228	Mar 4, 2021 @ 14:01:48.003
zeek	notice	CVE_2020_0601	Mar 2, 2021 @ 00:00:00.145
suricata	alert	CVE_2021_44228	Mar 1, 2021 @ 23:59:59.509
suricata	alert	CVE_2020_1472	Mar 1, 2021 @ 23:03:47.273
zeek	notice	CVE_2020_16898	Mar 1, 2021 @ 23:00:13.033
zeek	notice	CVE_2020_13777	Mar 1, 2021 @ 23:00:09.423
zeek	notice	CVE_2021_41773	Mar 1, 2021 @ 23:00:03.326

**Malcolm**

Dashboard | ICS/IoT Security Overview

**Zeek Logs**

**General**

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

**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 ● IDS / IDS RPC / IDS SQL ● Telnet / rlogin / rsh ● Tunnels

**ICS/IoT Protocols**

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

**ICS/IoT Log Counts**

100,802 ethercat - Count

23,104 ctp - Count

16,570 bacnet - Count

12,657 cotp - Count

10,924 s7comm - Count

6,514 modbus - Count

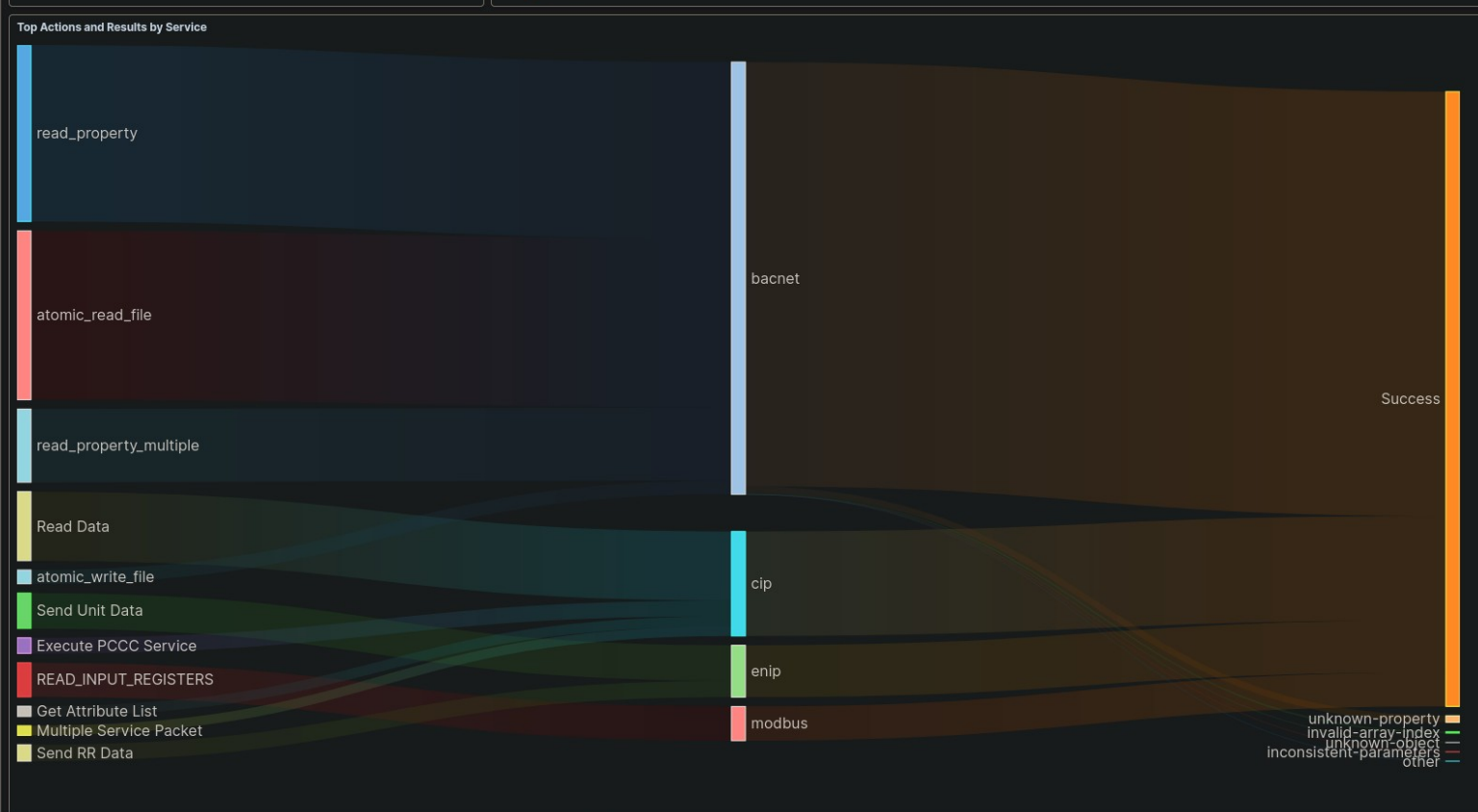
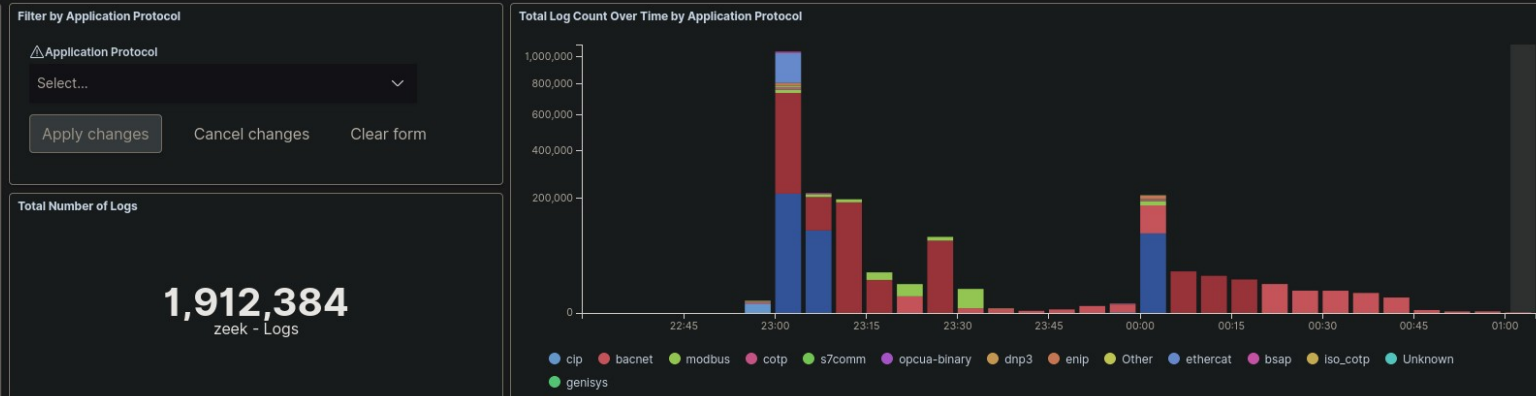
3,829 enip - Count

462 bsap - Count

**ICS/IoT Traffic Over Time**

**ICS/IoT External Traffic**

Protocol	Source IP	Source Country	Destination IP	Destination Country	Count
cotp	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	134.249.61.163	Ukraine	5
s7comm	134.249.62.199	Ukraine	134.249.61.163	Ukraine	5



	Count
multiple-ack	366,246
multiple-request	327,277
	106,521
	102,000
	95,000

Results

Protocol	Result	Count
bacnet	Success	277,619
cip	Success	65,421
enip	Success	32,059
modbus	Success	30,199
s7comm	Success	7,101

# 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/IIoT 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/IIoT 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

event.dataset:software Lucene Feb 28, 2021 @ 21:21:53.10 → Mar 1, 2021 @ 13:47:29.97 Update

+ Add filter

arkime\_sessions3-\*

Search field names

Filter by type 0

**Selected fields**

- source.ip
- url.full
- zeek.software.name
- zeek.software.software\_type
- zeek.software.unparsed\_version

**Available fields**

- \_id
- \_index
- \_score
- \_type
- @timestamp
- @version
- agent.hostname
- agent.id

2,092 hits Reset search

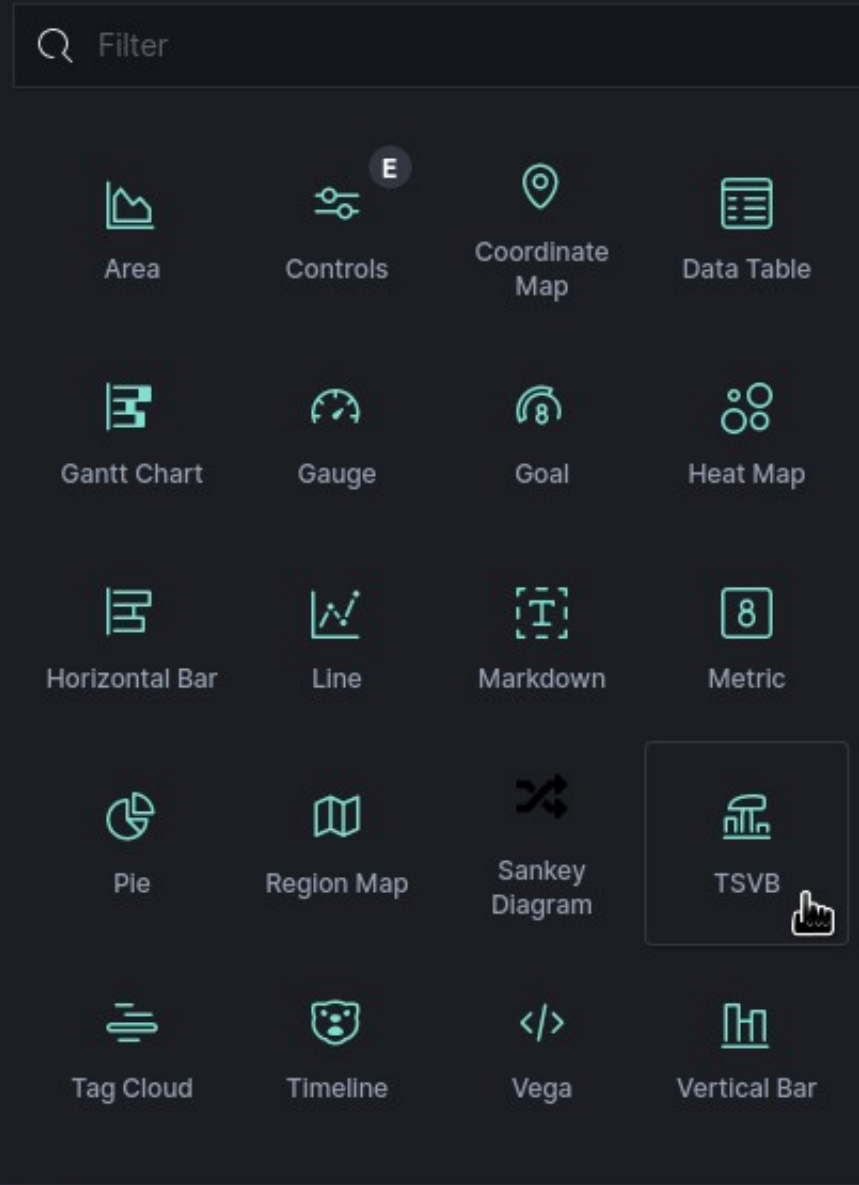
Feb 28, 2021 @ 21:21:53.108 - Mar 1, 2021 @ 13:47:29.975 Auto

Count

firstPacket per 10 minutes

Time	source.ip	zeek.software.software_type	zeek.software.name	zeek.software.unparsed_version	url.full
> Mar 1, 2021 @ 06:03:20.675	149.142.85.90	HTTP::BROWSER	MSIE	Mozilla/4.0 (compatible; MSIE 5.0; Windows 98; DigExt)	-
> Mar 1, 2021 @ 05:51:00.823	124.106.97.191	HTTP::SERVER	Microsoft-IIS	Microsoft-IIS/5.0	-
> Mar 1, 2021 @ 01:37:29.680	173.194.205.103	HTTP::SERVER	gws	gws	-
> Mar 1, 2021 @ 01:37:29.575	192.168.2.7	HTTP::BROWSER	Chrome	Mozilla/5.0 (Linux; Android 6.0.1; Blackphone 2 Build/MOB31Z) AppleWebKit/537.36 (KHTML, like Gec ko) Chrome/67.0.3396.87 Mobile Safari/537.36	-
> Mar 1, 2021 @ 01:34:00.114	104.117.42.243	HTTP::SERVER	letty/	letty(8.1.4.v20120524)	-

# New Visualization



# 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 &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.* /	<b>Dashboards Query Language does not currently support regex</b>
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.)

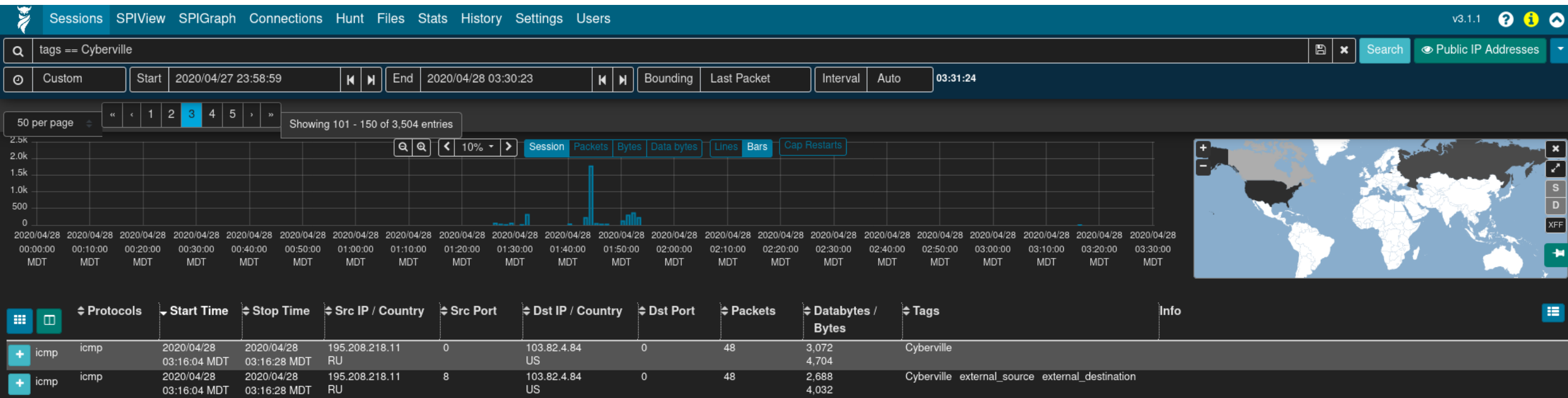
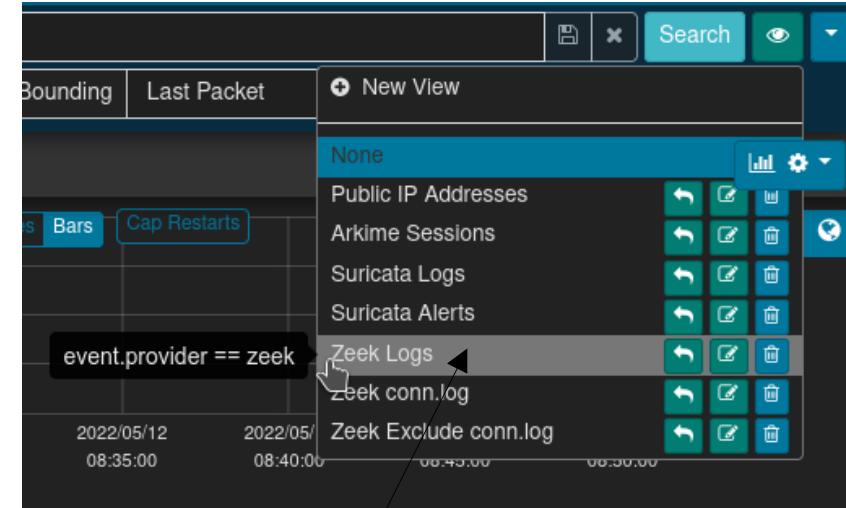
	Arkime	Dashboards (Lucene)	Dashboards (DQL)
GeoIP information available	<code>country == EXISTS!</code>	<code>_exists_:destination.geo OR _exists_:source.geo</code>	<code>destination.geo:* or source.geo:*</code>
Log type	<code>event.dataset == notice</code>	<code>event.dataset:notice</code>	<code>event.dataset:notice</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>



- 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>

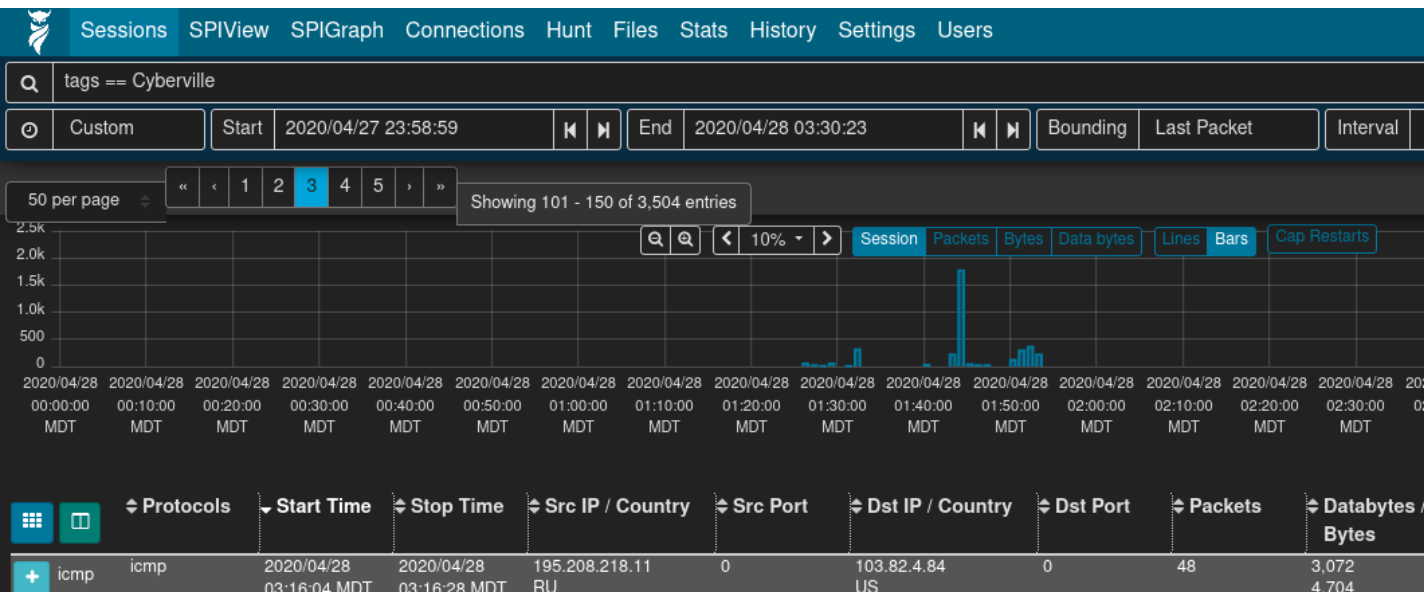
# Arkime Filters and Search

- 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 detailed session information for a Zeek http.log entry. The search filter is 'protocols == http && tags == external\_destination'. The time range is from 2020/11/11 06:23:48 to 2021/05/30 06:00:53. The entry is identified as 'Zeek http.log'.

Field	Value
Log Type	http
Malcolm Data Source	zeek
Malcolm Node	filebeat
Originating Host	217.226.31.170
Originating GeolIP Country	Germany
Originating GeolIP City	Bremen
Responding Host	124.106.97.191
Responding GeolIP Country	Philippines
Responding GeolIP 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
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  <a href="#">PCAnyPass.exe</a>

Packets 200   natural   ascii   utf8   hex   Show Packets   Line Numbers   Uncompress   Show Image & Files   Show Info   File Bytes:   base64   CyberChef

# Export PCAP

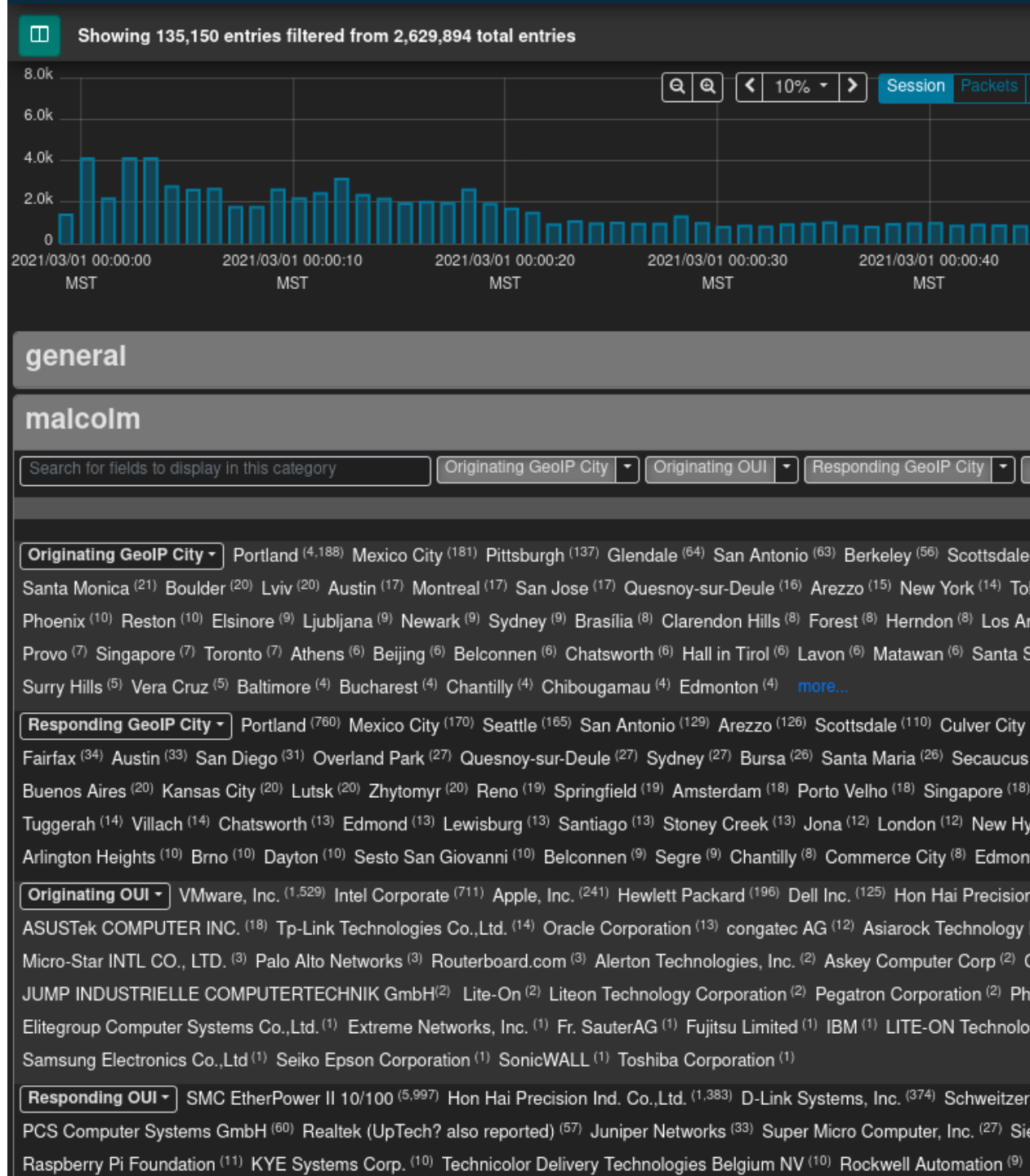
- Creates a new PCAP file from filtered sessions
- Include open, visible or all matching sessions
- Apply “Arkime Sessions” view to sessions first
- Narrow as much as possible prior to exporting (huge PCAP files are a pain)

The screenshot displays the Arkime web interface for session management. At the top, the navigation menu includes Sessions, SPIView, SPIGraph, Connections, Hunt, Files, Stats, History, Settings, and Users. The search bar contains the query 'country != US && protocols == http'. Below the search bar, the 'Custom' filter is active, with a start time of '2021/02/28 23:59:11' and an end time of '2021/03/01 00:28:26'. The 'Bounding' box is set to 'Last Packet' and the 'Interval' is 'Auto' (00:29:15). The 'Include' options are 'same time period' and 'linked segments (slow)'. The 'Filename' field is set to 'US\_HTTP.pcap'. The 'Export PCAP' button is visible on the right. The main area shows a timeline view of sessions with a bar chart and a world map. The bar chart displays session activity over time, with a peak around 00:02:00. The world map shows the location of the sessions, with a focus on the United States. The bottom of the interface shows a table of session details with 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 of data is: 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.



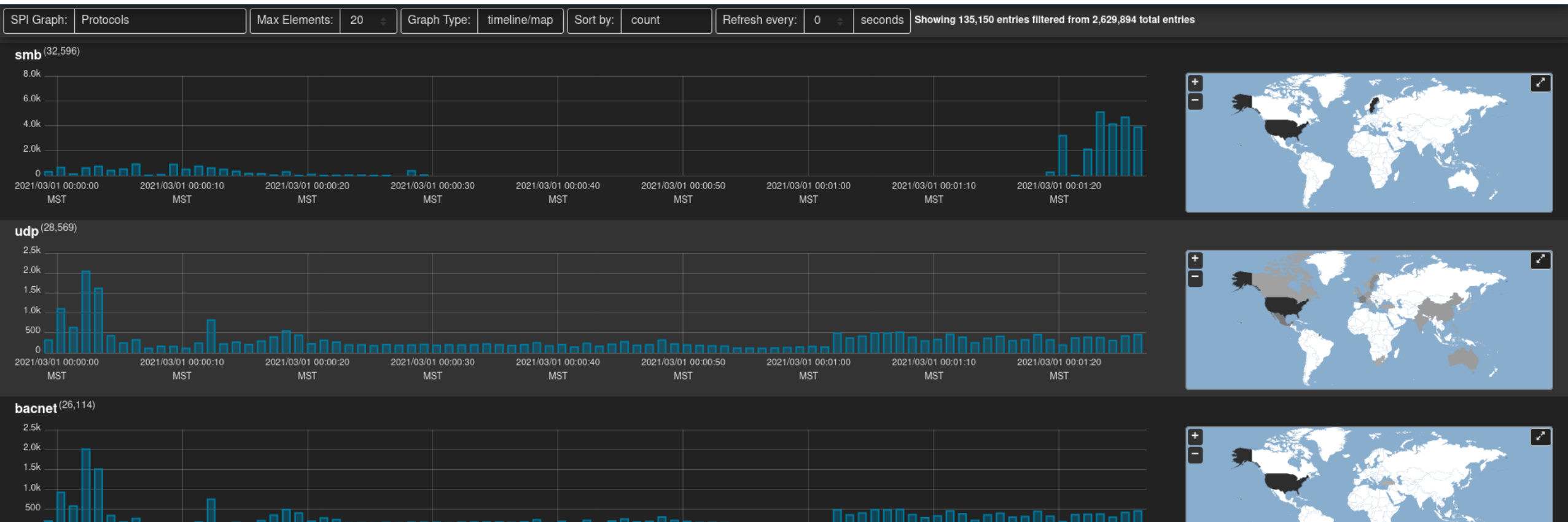
# SPIView

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



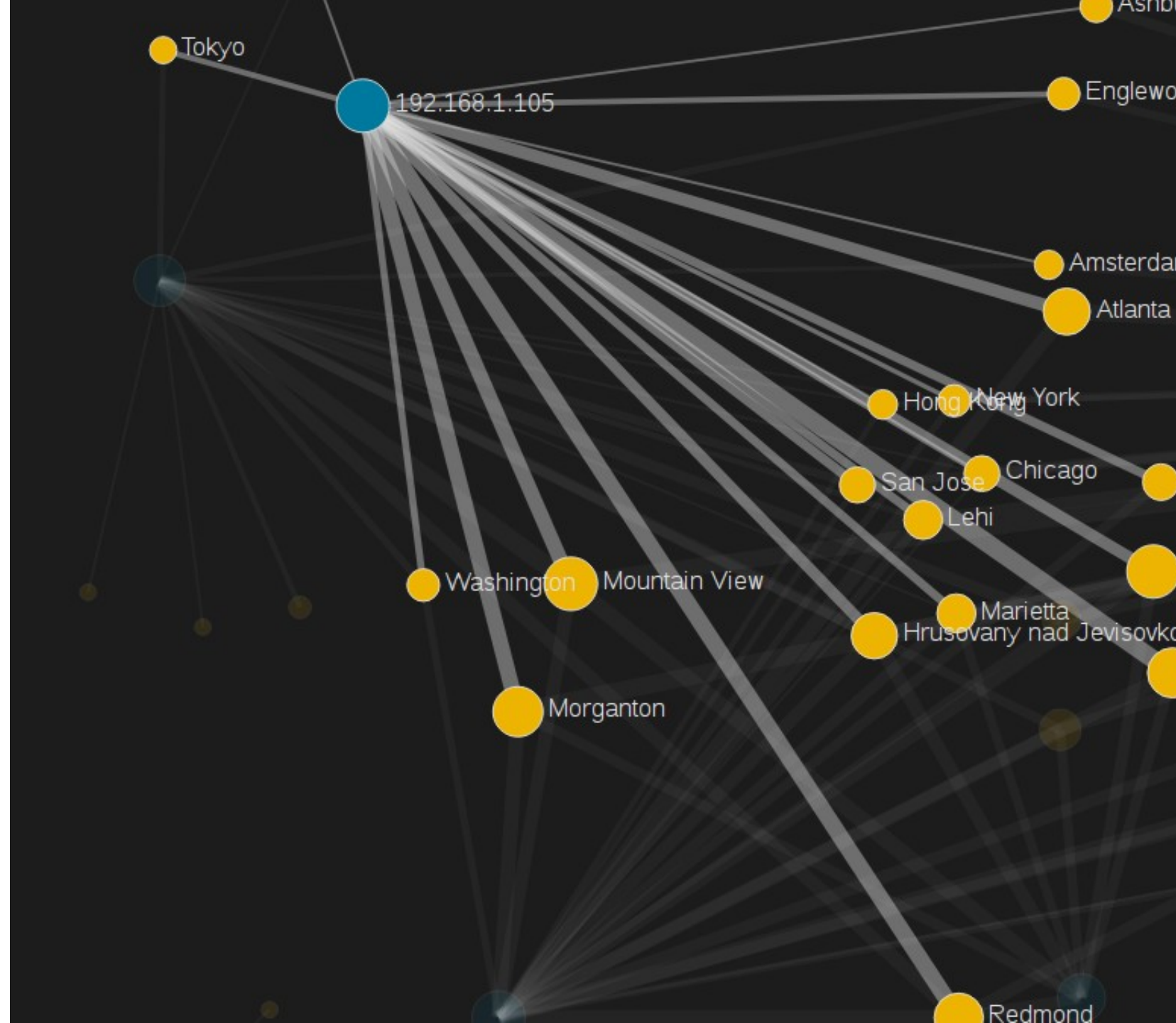
# SPIGraph

- 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

The screenshot displays the Arkime Hunt interface. At the top, there is a navigation bar with tabs for Sessions, SPIView, SPIGraph, Connections, Hunt (selected), Files, Stats, History, Settings, and Users. The version number v3.1.1 and utility icons are on the right. Below the navigation bar is a search bar containing the query 'protocols == http'. A filter dropdown is set to 'All (careful)'. The search parameters are: Start: 1969/12/31 17:00:00, End: 2021/12/06 12:10:02, Bounding: Last Packet. A notification states: 'Creating a new packet search job will search the packets of 2,906 sessions.' Below this is the 'Hunt Job History' section with a search bar and a table of search jobs. The table has columns: Status, Matches, Name, User, Search text, Notify, Created, and ID. One job is listed with a status of 'finished', 141 matches, and search text 'password (ascii)'. Below the table, a list of details for the selected job is shown, including the number of sessions found (141 out of 2,908), creation time, and search parameters.

Sessions SPIView SPIGraph Connections **Hunt** Files Stats History Settings Users v3.1.1 ? ⓘ ⬆

Q 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

Q 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
✓ 100%	141	HTTP with password		password (ascii)		2021/12/06 12:12:27 MST	s5YpkX0BTA40FhD4X7dA [C] [↺] [📧] [✕] [🗑️]

✓ 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

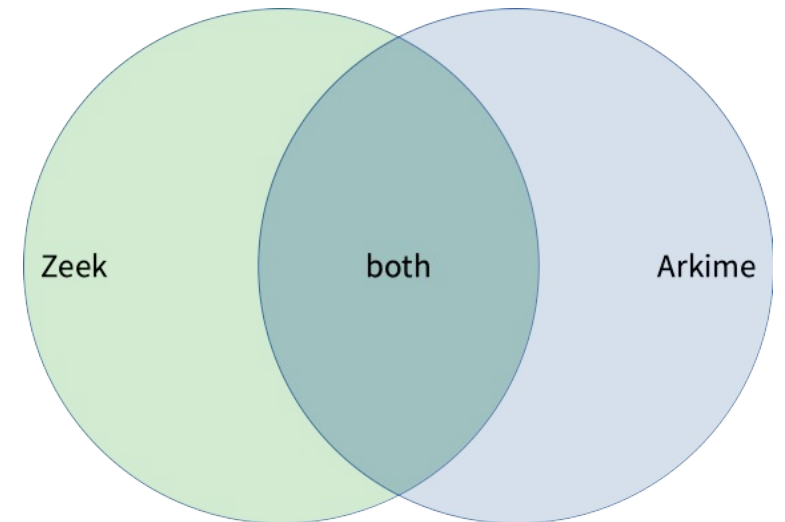
🔍 The sessions query expression was: **protocols == http**

🔍 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

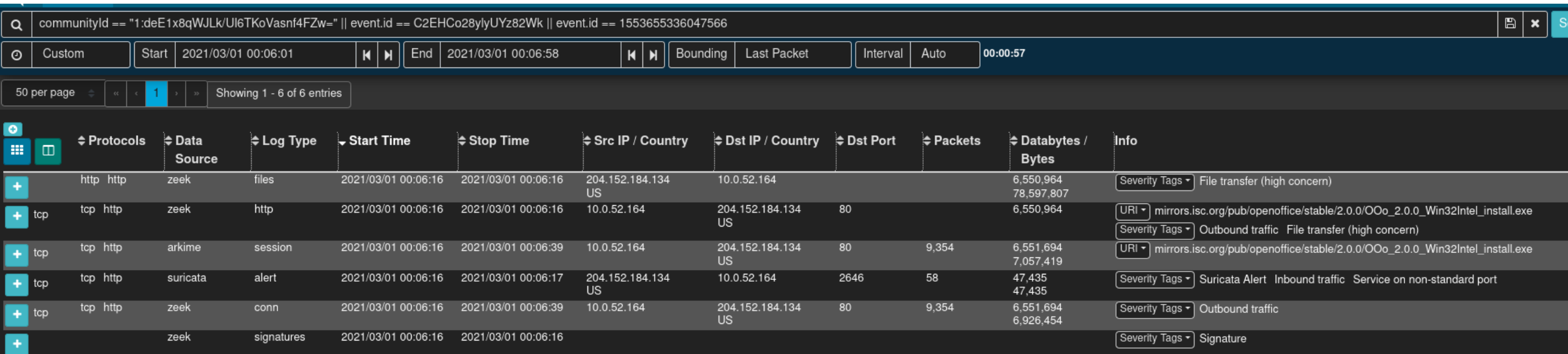
- Search syntax is different between Arkime and Dashboards (and in some cases, so are field names)
  - See search syntax comparison table, Malcolm and Arkime docs
- Despite considerable overlap, there are differences in protocol parser support among Zeek, Suricata and Arkime
  - Learning the strengths 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 a network analysis tool interface with a search bar at the top containing the query: `communityId == "1:deE1x8qWJLk/Ul6TKoVasnf4FZw=" || event.id == C2EHCo28ylyUYz82Wk || event.id == 1553655336047566`. Below the search bar are controls for filters (Custom, Start, End, Bounding, Last Packet, Interval, Auto) and a timer (00:00:57). A pagination bar shows "50 per page" and "Showing 1 - 6 of 6 entries". The main table displays log entries with columns for Protocols, Data Source, Log Type, Start Time, Stop Time, Src IP / Country, Dst IP / Country, Dst Port, Packets, Databytes / Bytes, and Info. The entries include Zeek files, http, tcp, and session logs, as well as Suricata alerts and signatures.

	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

The screenshot displays a security dashboard interface. On the left is a navigation menu with sections: 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, rlogiqn, rsh, Tunnels), and ICS/IoT Protocols (BACnet, BSAP, DNP3, EtherCAT, EtherNet/IP, GENISYS, Modbus, OPCUA Binary, PROFINET, S7comm, Best Guess).

The top right features a bar chart showing scanned file hits over time, with a peak at 23:00. The number '112' is displayed prominently above the chart.

The main content area is divided into two panels. The left panel, titled 'Signatures - Engine', contains a word cloud of antivirus vendors including TrendMicro, Microsoft, Zillya, Alibaba, MicroWorld-eScan, ESET-NOD32, Cynet, Avast, NANO-Antivirus, Ikarus, ClamAV, Sophos, McAfee, Capa, GData, Rising, DrWeb, AVG, zeek, Ad-Aware, Tencent, Kaspersky, APEX, Comodo, Emsisoft, MAX, BitDefender, Avira, FireEye, Symantec, Antiy-AVL, Panda, VIPRE, and Fortinet.

The right panel, titled 'Signatures - Name', shows a list of signature details for 'smb-nt-transact-rename', including execution paths, discovery methods, and a score of 100.

At the bottom, a 'Signatures - Category' bar chart shows the distribution of signatures across categories: Sensitive\_Signature, Defense\_Evasion, Execution, Multiple\_Signatures, Discovery, Collection, and Impact.



# 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.