

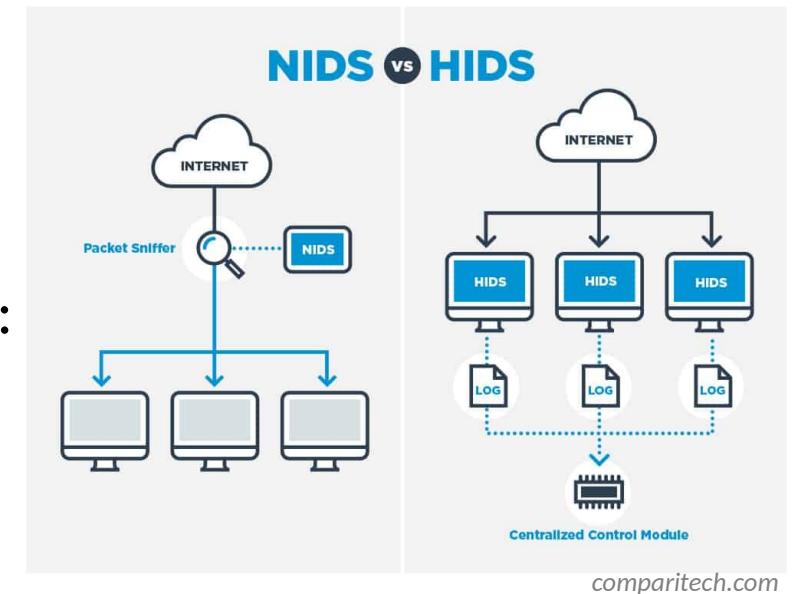
# Network Traffic Analysis with **Malcolm**

A faint watermark of the Malcolm logo is visible behind the word "Malcolm". The logo consists of a stylized yellow 'M' shape with intricate internal patterns, centered over a grid of thin blue lines.

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
    - Statistical anomaly-based detection
    - Stateful protocol analysis detection



# 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

The diagram illustrates the structure of four Zeek log files:

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

FIELD	TYPE	DESCRIPTION
id	int	Timestamp of the first packet.
src	string	Unique ID of the connection.
src_ip	addr	Originating endpoint's IP address (IPv4).
src_port	port	Originating endpoint's TCP/UDP port (or ICMP code).
dst_ip	addr	Receiving endpoint's IP address (IPv4).
dst_port	port	Receiving endpoint's TCP/UDP port (or ICMP code).
proto	proto	Transport layer protocol of connection.
service	string	Defined application protocol, if any.
duration	interval	Connection duration.
http_type	object	Http payload type from sequence numbers (HTTP).
log_type	object	Http payload type from sequence numbers (HTTP).
www_state	string	Connection state (see <code>state</code> , <code>www_state</code> ).
first_orig	bool	Is Origin in functional net?
last_orig	bool	Is Last in functional net?
maxwell_index	object	Number of bytes missing due to content gaps.
memory	string	Free memory (see <code>mem_usage</code> , <code>mem_usage_percent</code> ).
avg_pkts	object	Average of CNG packets.
avg_pkts_before	object	Number of CNG packets (see <code>P1</code> and <code>avg_pkts_after</code> ).
max_pkts	object	Number of flow controls.
max_pkts_before	object	Number of flow controls (see <code>P1</code> and <code>avg_pkts_after</code> ).
formed_packets	int	Number of connection-IO of evasions being removed.
ping_id_addr	string	Last upper address of the originator.
pong_ip_addr	string	Last lower address of the responder.
mac	obj	The user MAC for this connection.
inet_ifname	obj	The inner VLAN for this connection.
- http.log | HTTP request/reply details**:

FIELD	TYPE	DESCRIPTION
id	int	Timestamp of the HTTP request.
src_ip	string	Underlying connection id. See <code>conn.log</code> .
trans_depth	object	Protocol depth into the connection.
method	string	HTTP Request method (GET, POST, HEAD, etc.).
host	string	Name of the host header.
uri	string	URI used in the request.
referer	string	Value of the "Referer" header.
user_agent	string	Name of the User-Agent header.
response_body_hex	string	Uncompressed content-use of big data.
response_body_hex_size	string	Uncompressed content-size of big data.
status_code	string	Status code returned by the server.
status_msg	string	Status message returned by the server.
info_code	string	Last seen fail info message code by server.
info_gmsg	string	Last seen fail info message gmsg by server.
tags	set	Indicators of various attributes discovered.
username	string	Username if user auth is performed.
password	string	Password if user auth is performed.
present	obj	Headers indication of a present request.
req_headers	vector	The unique On-the-Fly headers.
req_thunks	vector	The thunks On-the-Fly.
req_name_types	vector	The types On-the-Fly.
req_pkts	vector	The unique On-the-Fly.
req_thunks	vector	The thunks On-the-Fly.
req_name_types	vector	The types On-the-Fly.
client_header	vector	The names of HTTP readers sent by client.
server_header	vector	The names of HTTP readers sent by client.
cookies_name	vector	Name value names extracted from cookies.
url_name	vector	Variable names extracted from the URL.
- files.log | File analysis results**:

FIELD	TYPE	DESCRIPTION
id	int	Timestamp when the analysis began.
file	string	Unique identifier for average file.
is_file	int	Method that scanned the data.
is_filedir	int	Method that scanned the data.
conn_with	int	Connection ID/observer which the transferred.
source	string	An identification of the source of the data.
target	string	Target or the related to source (e.g., HTTP request response).
analysis	int	Method of analysis when handling the analysis.
minm_size	string	The size of the file scanned by this signatures.
maxm_size	string	Maximun size of the file scanned by this signatures.
duration	interval	The duration that the file was analyzed.
local_orig	bool	Did the data originate locally?
is_forg	bool	Was the file written by the Originator?
source_ipfile	object	Number of files scanned by the analysis engine.
total_bytes	object	Total number of bytes that should compress the file.
missing_bytes	object	Number of bytes in the stream missed.
corrupt_bytes	object	Out of range size bytes in the stream due to corruption.
streamed	bool	If the file analysis module ran at least once.
parent_file	string	Container file ID this was extracted from.
modified	string	MD5/SHA1 hash of the file.
extnames	string	List of names of renamed files, if renamed.
entropy	double	Information density of the file contents.
- pe.log | Portable Executable (PE)**:

FIELD	TYPE	DESCRIPTION
id	int	Current timestamp.
file	string	The final file path under which the file was saved.
machine	string	The processor that the file was created on.
compile_id	string	The process that the file was created on.
os	string	The measured operating system.
dependencies	string	The dependencies that are required for the file.
is_exec	bool	Is the file an executable, or just an object?
is_dll	bool	Is the file a DLL file or shared library?
is_malware	bool	Does the file suspect to be a known malware?
is_virus	bool	Does the file suspect to be a known virus?
is_worm	bool	Does the file suspect to be a known worm?
is_rootkit	bool	Does the file suspect to be a known rootkit?
is_backdoor	bool	Does the file suspect to be a known backdoor?
is_spammer	bool	Does the file suspect to be a known spammer?
has_imports	bool	Does the file have an import table?
has_exports	bool	Does the file have an export table?
has_crypt	bool	Does the file have an anti-decompression code?
has_inj	bool	Does the file have an injecting table?
section_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

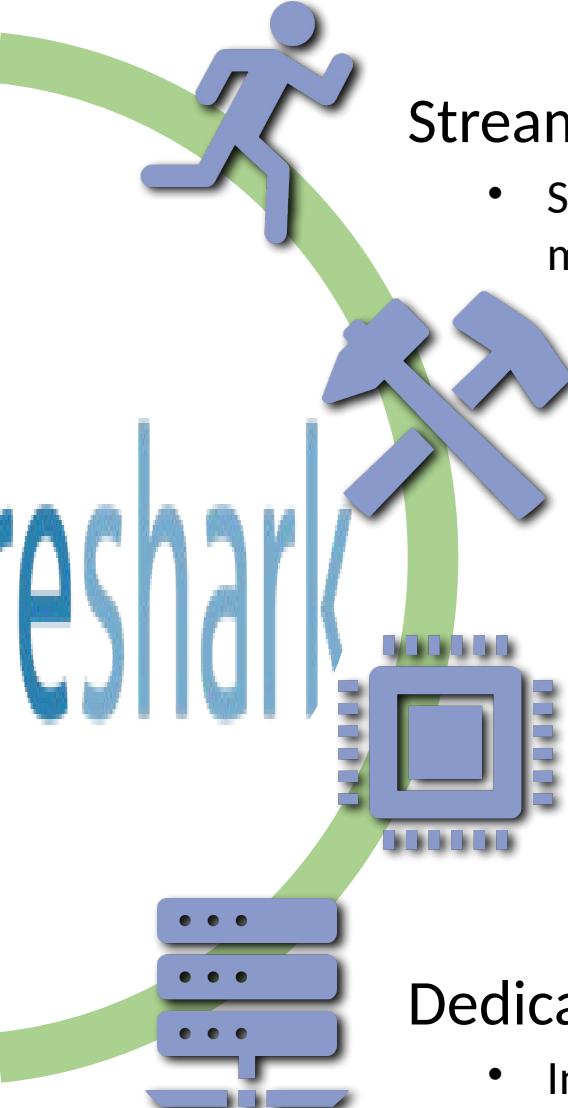
## Weaknesses

- 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, 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>



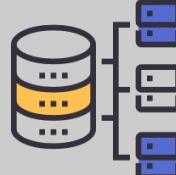
Capture



File Scanning



Forwarding &  
Enrichment



Storage



Anomaly  
Detection



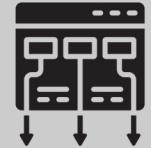
Alerting



Visualization



Payload  
Analysis



Framework



zeek



yara



Arkime



ClamAV®



beats



OpenSearch



OpenSearch  
Anomaly  
Detection  
Plugin



OpenSearch  
Alerting  
Plugin



OpenSearch  
Dashboards



CyberChef



docker



netsniff-ng



CAPA



logstash



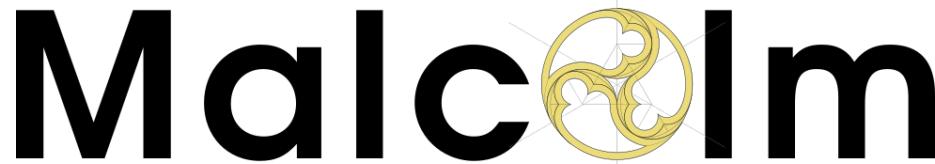
Arkime  
session PCAP  
export to  
WIRESHARK



TCPDUMP



VIRUSTOTAL



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)  
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>

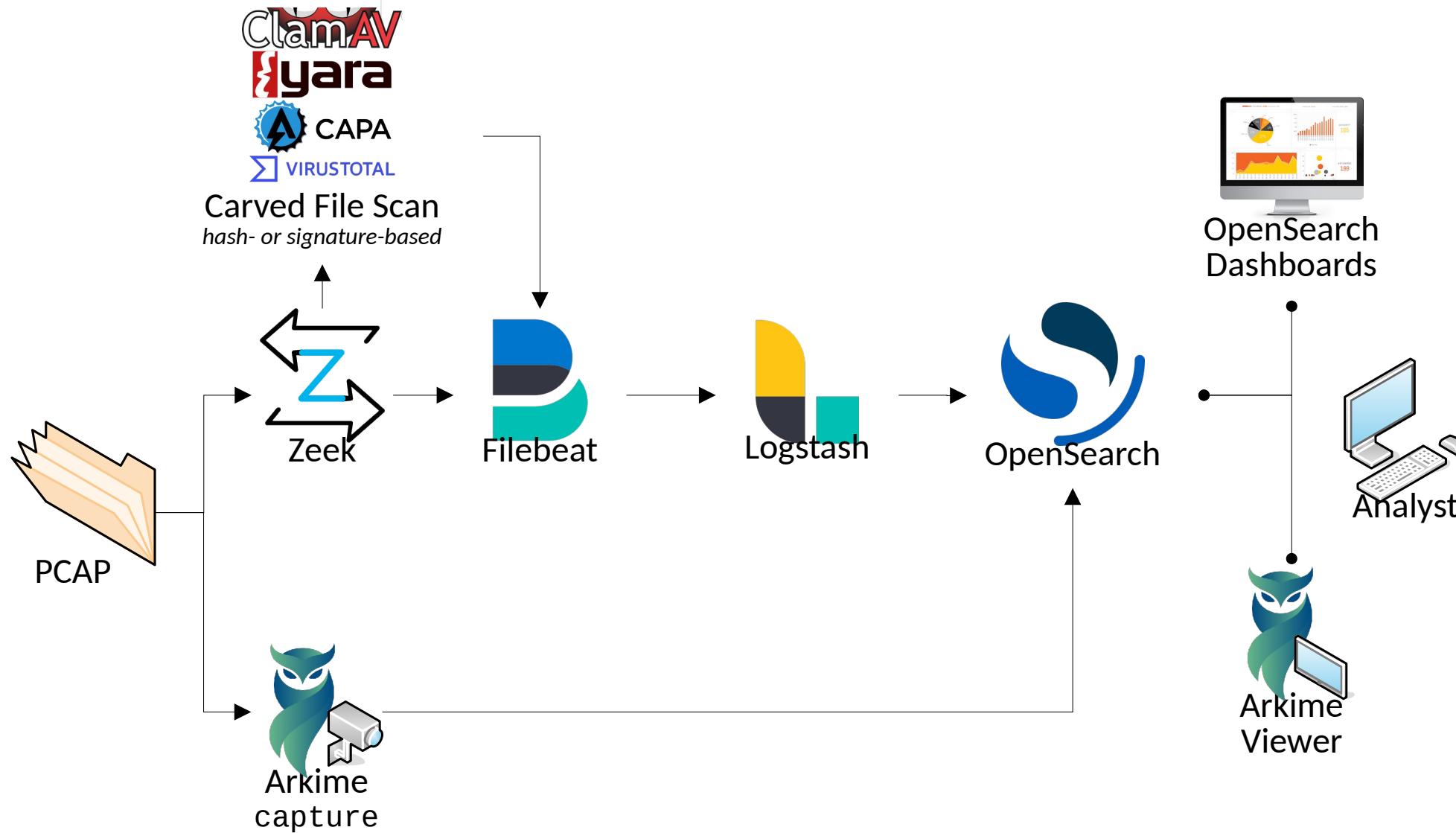
Open Shortest Path First (OSPF)  
OpenVPN  
PostgreSQL  
**Process Field Net (PROFINET)**  
Remote Authentication Dial-In User Service (RADIUS)  
Remote Desktop Protocol (RDP)  
Remote Framebuffer (RFB)  
**S7comm / Connection Oriented Transport Protocol (COTP)**  
Secure Shell (SSH)  
Secure Sockets 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>

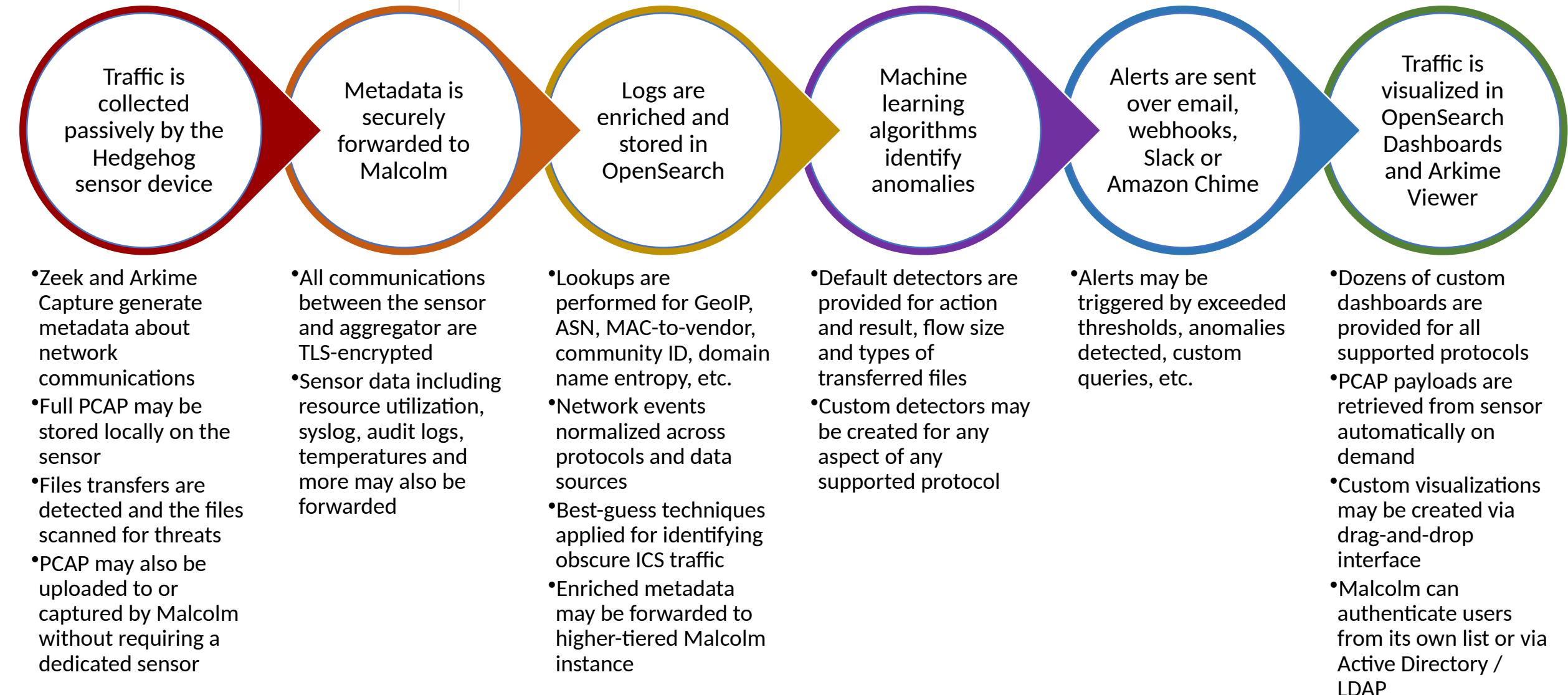


# Malcolm



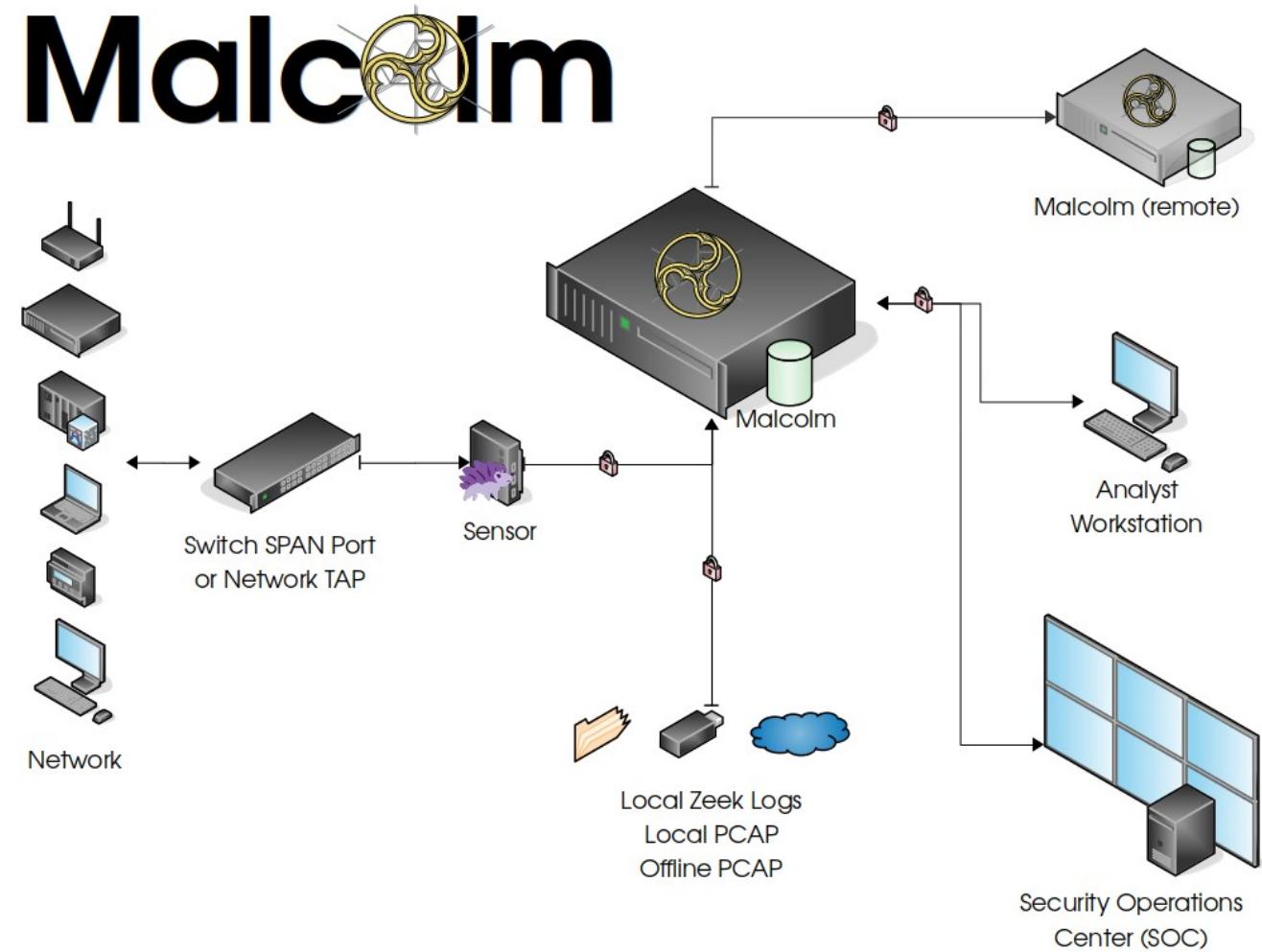
## Data Pipeline

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



# 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”



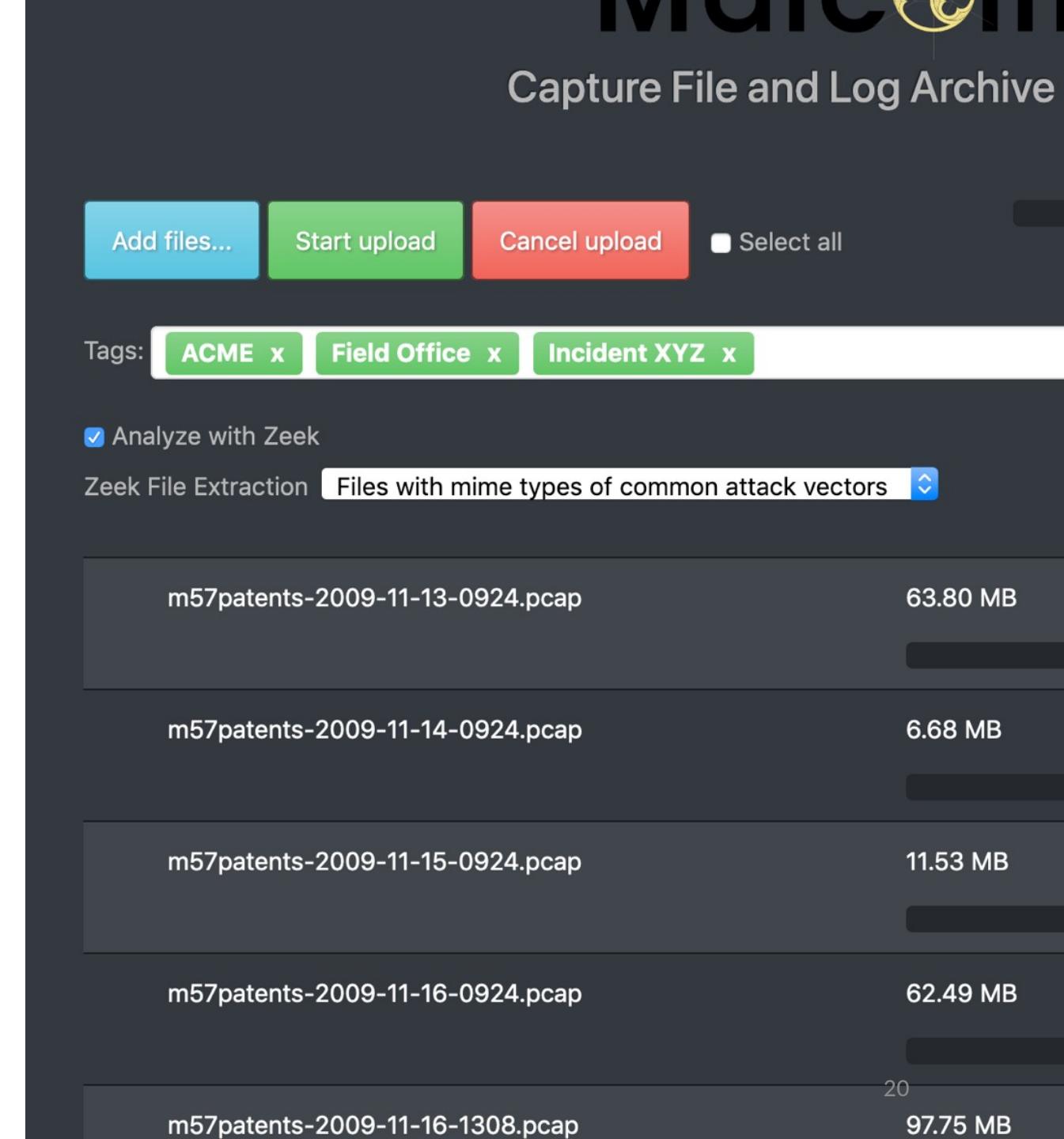
# Identifying Network Hosts and Subnets

- 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>

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

# Importing Traffic Captures for Analysis

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



# Data Tagging and Enrichment



- Logstash enriches Zeek log data
  - 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 both Arkime sessions and Zeek logs 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
- 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>

The screenshot shows the OpenSearch Dashboards interface with the 'Malcolm' dashboard selected. The top navigation bar includes 'Dashboard' and 'Security Overview'. The main content area is divided into several sections:

- Zeek Logs**: A sidebar with links to various log categories: General, Overview, Security Overview, ICS/IoT Security Overview, Severity, Connections, Actions and Results, Files, Executables, Software, Notices, Weird, Signatures, Intel Feeds, and Arkime.
- Common Protocols**: A list of network 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, and Tunnels.
- ICS/IoT Protocols**: A section listing Outdated/Insecure Application Protocols: Application Protocol (Protocol Version, Count). The data is as follows:

Application Protocol	Protocol Version	Count
ftp	-	1,063
smb	1	535
tftp	-	64
ntp	3	42
tls	TLSv10	38

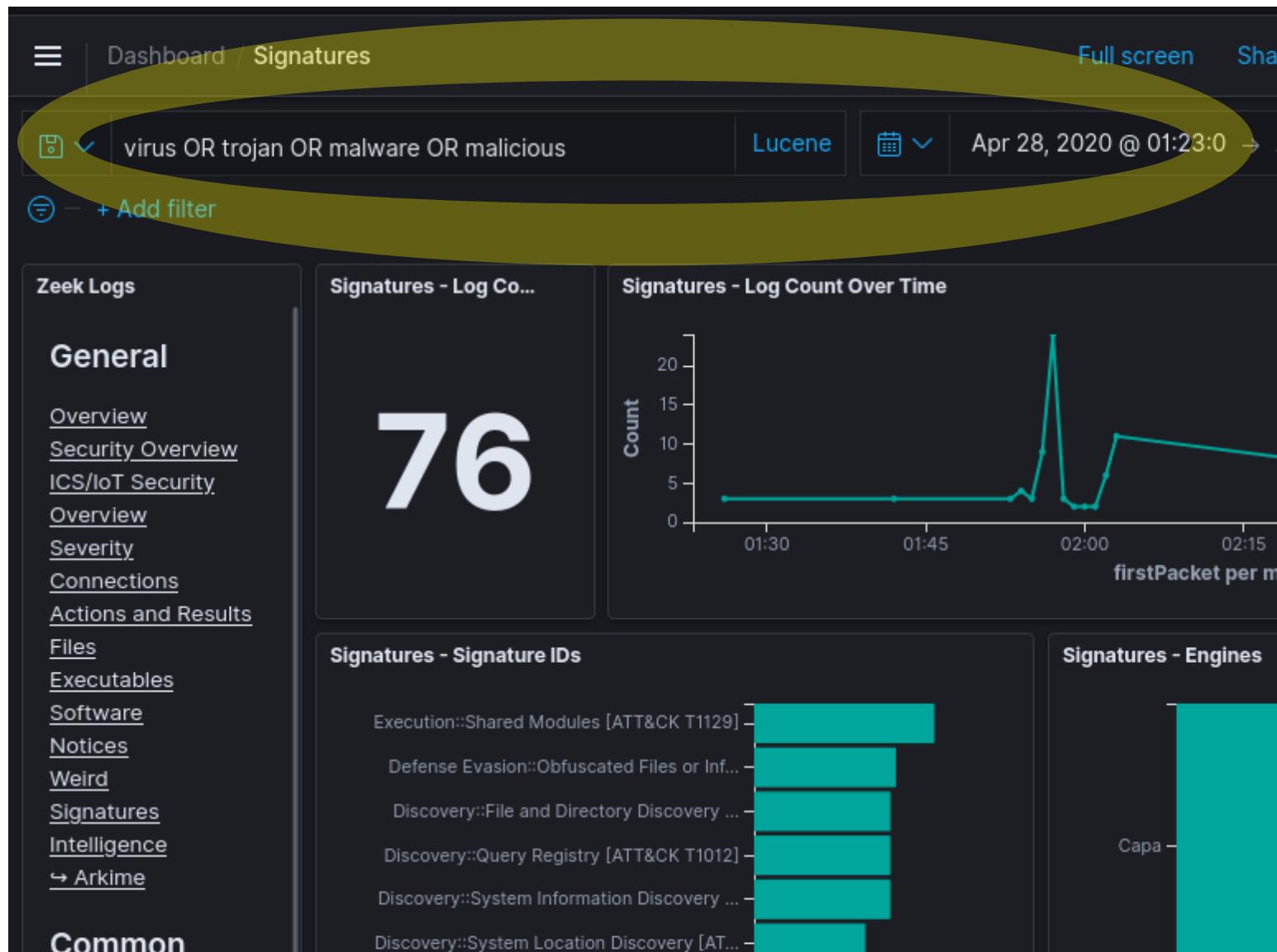
- Notices by Category**: A table showing the count of notices categorized by type. The data is as follows:

Notice Category	Count
SSL::Invalid_Server_Cert	50
ATTACK::Execution	27
ATTACK::Lateral_Movement	6
EternalSafety::EternalSynergy	5
ATTACK::Lateral_Movement_Multiple_Attempts	4
Signatures::Sensitive_Signature	4
ATTACK::Lateral_Movement_Extracted_File	2
EternalSafety::EternalChampion	2
EternalSafety::ViolationNtRename	2
EternalSafety::ViolationTx2Cmd	2
ATTACK::Discovery	1
EternalSafety::DoublePulsar	1
FTP::Bruteforcing	1
Ripple20::Treck_TCP_observed	1

- Connections by Destination Country**: A map visualization showing the distribution of connections by destination country.

# 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

The screenshot shows the Malcolm security dashboard. At the top, there's a navigation bar with the title "Malcolm" and a globe icon. Below it, a secondary navigation bar shows "Dashboard / Security Overview". On the left, a sidebar titled "Zeek Logs" contains a "General" section with links to various log categories: Overview, Security Overview, ICS/IoT Security Overview, Severity, Connections, Actions and Results, Files, Executables, Software, Notices, Weird, Signatures, Intel Feeds, and Arkime. To the right of the sidebar, a large panel displays "Common Protocols" with icons for DCE/RPC, DHCP, DNS, FTP / TFTP, and others. A vertical sidebar on the far right lists "Notices by Category" with items like SSL::Invalid\_Server\_..., ATTACK::Execution, ATTACK::Lateral\_Mo..., EternalSafety::Etern..., ATTACK::Lateral\_Mo..., Signatures::Sensitive..., ATTACK::Lateral\_Mo..., EternalSafety::Etern..., EternalSafety::Violat..., EternalSafety::Violat..., ATTACK::Discovery, EternalSafety::Doub..., and FTP::Bruteforcing.

Malcolm

☰ Dashboard / 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](#)  
[Arkime](#)

Common Protocols

DCE/RPC ● DHCP ● DNS ● FTP / TFTP ●

Notices by Category

Notice Category

SSL::Invalid\_Server\_...  
ATTACK::Execution  
ATTACK::Lateral\_Mo...  
EternalSafety::Etern...  
ATTACK::Lateral\_Mo...  
Signatures::Sensitive...  
ATTACK::Lateral\_Mo...  
EternalSafety::Etern...  
EternalSafety::Violat...  
EternalSafety::Violat...  
ATTACK::Discovery  
EternalSafety::Doub...  
FTP::Bruteforcing

# 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 / Notices

## Zeek Logs

### 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
- 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
- Modbus
- PROFINET
- S7comm
- Best Guess

## Notices - Log Count

108

## Notices - Log Count Over Time

Time	Count
23:00:00	108

## Notices - Notice Type

Notice Category	Notice Subcategory	Count
SSL	Invalid_Server_Cert	50
ATTACK	Execution	27
ATTACK	Lateral_Movement	6
EternalSafety	EternalSynergy	5
Signatures	Sensitive_Signature	4
ATTACK	Lateral_Movement_Multiple_Attempts	4
EternalSafety	ViolationTx2Cmd	2
EternalSafety	ViolationNtRename	2
EternalSafety	EternalChampion	2
ATTACK	Lateral_Movement_Extracted_File	2

Export: [Raw](#) [Formatted](#)

# Security & ICS/IoT Security Overviews

**Malcolm**

Dashboard / 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
- ↳ 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**

**Outdated/Insecure Application Protocols**

Application Protocol	Protocol Version	Count
ftp	-	1,063
smb	1	535
tftp	-	64
ntp	3	42
tls	TLSv10	38

**Connections by Destination Country (region map)**

**Signatures - Signature IDs**

Defense Evasion:Obfuscated Files or Inf... 50  
Execution:Shared Modules [ATT&CK T1129] 27  
Discovery:Query Registry [ATT&CK T1012] 6  
Discovery:File and Directory Discovery ... 5  
Collection:Clipboard Data [ATT&CK T115... 4  
Discovery:System Information Discovery ...  
Execution:Command and Scripting Interpr.  
Signatures:Sensitive\_Signature 4  
ATTACK:Lateral\_Movement\_Extracted\_File 2  
EternalSafety:EternalChampion 2  
EternalSafety:ViolationNTRename 2  
EternalSafety:ViolationTx2Cmd 2  
ATTACK:Discovery 1  
EternalSafety:DoublePulsar 1  
FTP:Bruteforcing 1  
Ripple20:Treck\_TCP\_observed 1

**Clear-text Transmission of Passwords**

Application Protocol	Username
ftp	anonymous
ftp	ind@psg420.com
http	Login
http	salesxfer
http	Unknown
ldap	xxxxxxxxxx@xx.xxxx.xxxx.net
ldap	cn=Administrator,cn=Users,dc=cloudshark-a,dc=example,dc=com
ldap	CN=xxxxxx,OU=Users,OU=Accounts,DC=xx,DC=xxx,DC=xxxxx,DC=r
ldap	CN=Tom,CN=Users,DC=cloudshark-a,DC=example,DC=com

**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
- ↳ 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**

**Network Layer**

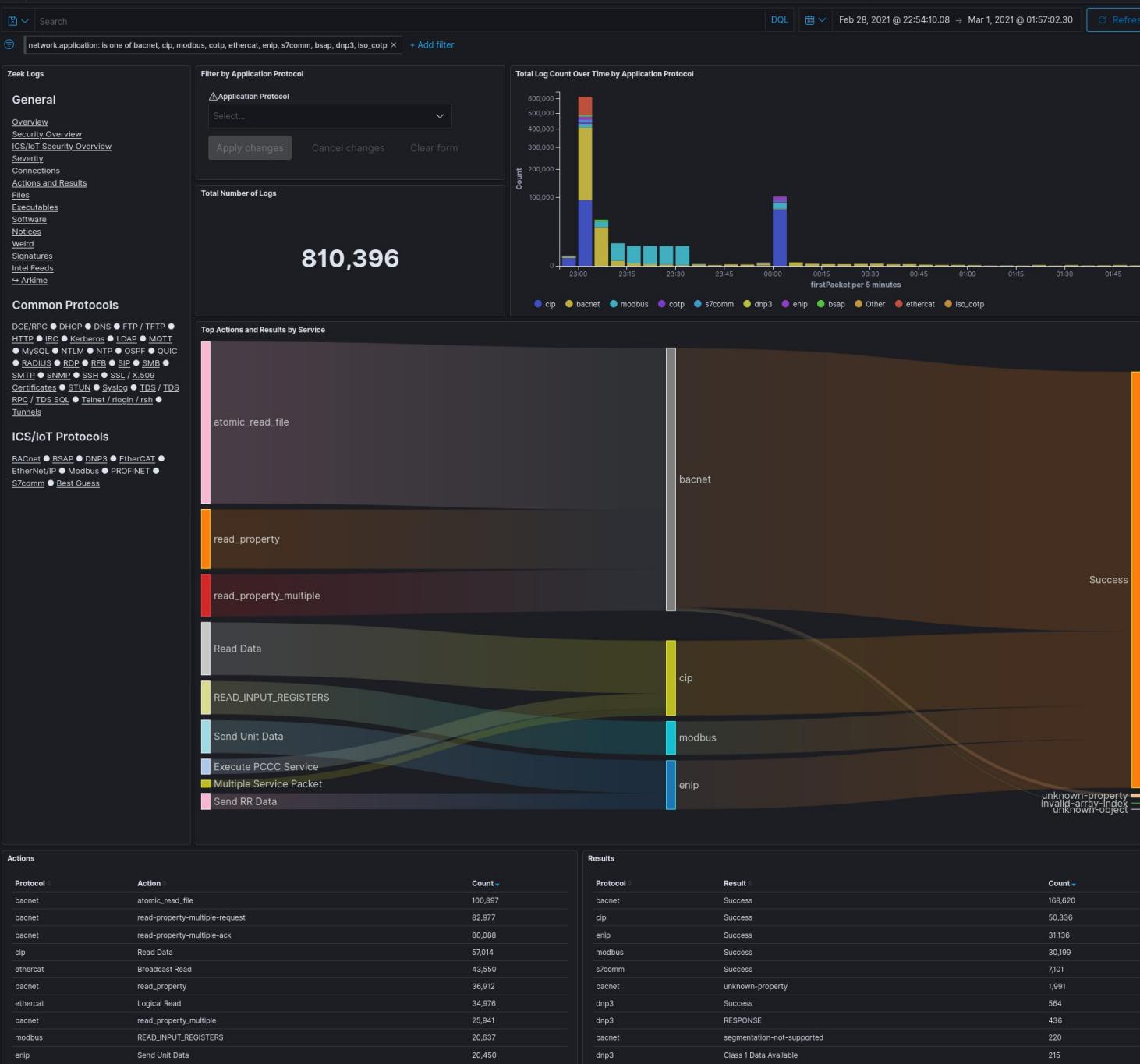
**ICS/IoT Log Counts**

- ethercat - Count: 100,802
- cip - Count: 23,104
- bacnet - Count: 16,570
- cotp - Count: 12,657
- s7comm - Count: 10,924
- modbus - Count: 6,514
- enip - Count: 3,829
- bsap - Count: 462

**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	134.249.62.209	Ukraine	134.249.61.182	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
  - Modbus
  - PROFINET
  - S7comm

[Intel Feeds](#)

[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](#) ● [Modbus](#) ● [PROFINET](#) ●  
[S7comm](#) ● [Best Guess](#)

# 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

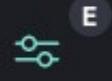


## New Visualization

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

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

# 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	srcIp:"0.0.0.0/0" OR dstIp:"0.0.0.0/0"	srcIp:"0.0.0.0/0" OR dstIp:"0.0.0.0/0"
IPv6 values	(ip.src == EXISTS!    ip.dst == EXISTS!) && (ip != 0.0.0.0/0)	(_exists_:srcIp AND NOT srcIp:"0.0.0.0/0") OR (_exists_:dstIp AND NOT dstIp:"0.0.0.0/0")	(srcIp:* and not srcIp:"0.0.0.0/0") or (dstIp:* and not dstIp:"0.0.0.0/0")

# Search Syntax Comparison (cont.)

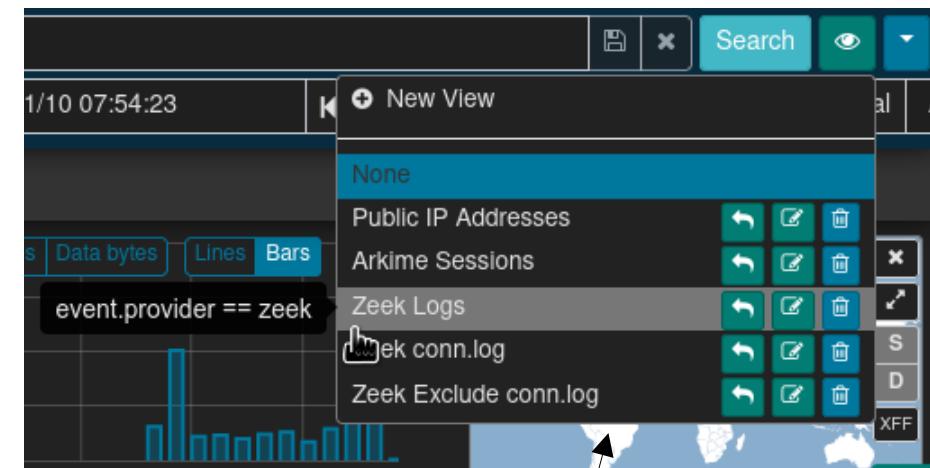
	Arkime	Dashboards (Lucene)	Dashboards (DQL)
GeolP information available	country == EXISTS!	_exists_:zeek.destination_geo OR _exists_:zeek.source_geo	zeek.destination_geo:* or zeek.source_geo:*
Zeek log type	zeek.logType == notice	zeek.logType:notice	zeek.logType:notice
IP CIDR Subnets	ip.src == 172.16.0.0/12	srcIp:"172.16.0.0/12"	srcIp:"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_:zeek.destination_geo OR _exists_:zeek.source_geo	zeek.destination_geo:* or zeek.source_geo:*



- Front end for **both** enriched Zeek logs and Arkime sessions
  - Malcolm's custom Arkime Zeek data source adds full support for Zeek logs to Arkime, including ICS protocols
- Filter by Zeek logs or Arkime sessions; or, view both 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



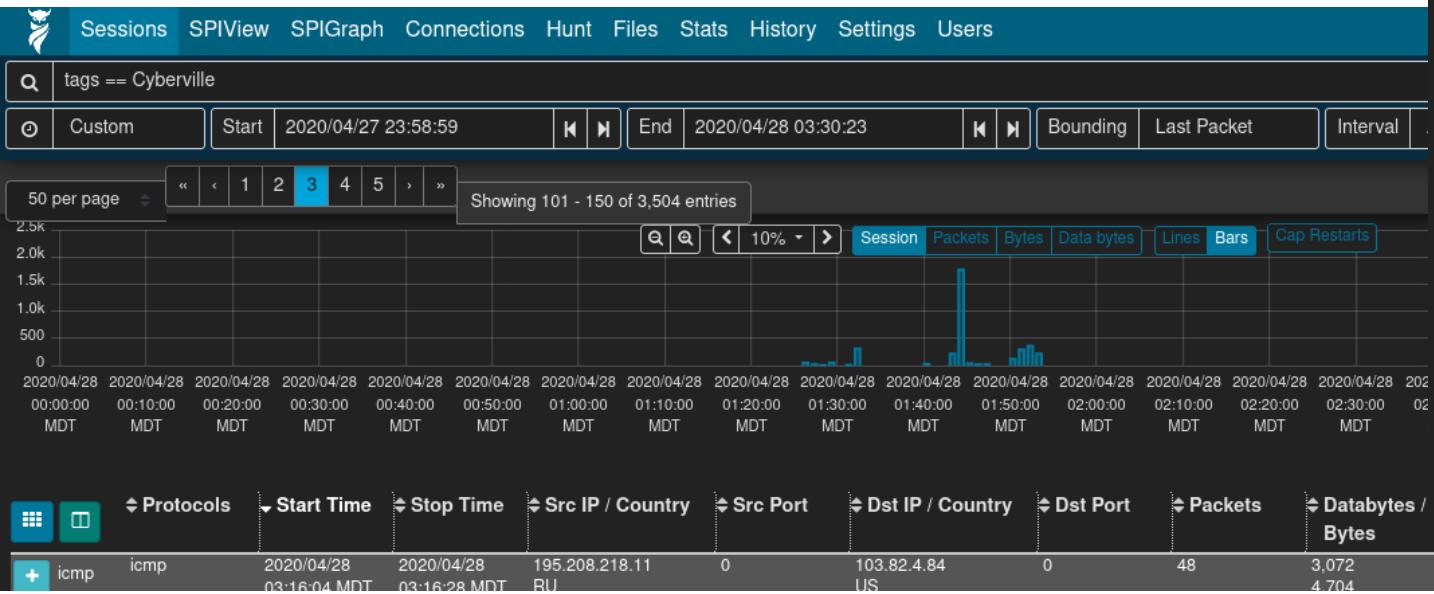
The Arkime interface is shown with the following components:

- Top Bar:** Sessions, SPIView, SPIGraph, Connections, Hunt, Files, Stats, History, Settings, Users.
- Search Bar:** Contains the query "tags == Cyberville".
- Time Filter:** Set to "Custom" from "Start" (2020/04/27 23:58:59) to "End" (2020/04/28 03:30:23).
- Timeline:** Shows a histogram of packet counts over time, with a single sharp peak around 2020/04/28 01:40:00 MDT.
- Map:** A world map showing the location of the "Cyberville" tag.
- Session List:** A table showing two sessions:

Protocol	Start Time	Stop Time	Src IP / Country	Src Port	Dst IP / Country	Dst Port	Packets	Databytes / Bytes	Tags
icmp	2020/04/28 03:16:04 MDT	2020/04/28 03:16:28 MDT	195.208.218.11 RU	0	103.82.4.84 US	0	48	3,072 4,704	Cyberville
icmp	2020/04/28 03:16:04 MDT	2020/04/28 03:16:28 MDT	195.208.218.11 RU	8	103.82.4.84 US	0	48	2,688 4,032	Cyberville external_source external_destination

# Sessions

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



The screenshot shows the Sessions interface with the following details:

- Navigation Bar:** Sessions, SPIView, SPIGraph, Connections, Hunt, Files, Stats, History, Settings, Users.
- Search Bar:** protocols == http && tags == external\_destination
- Filter Bar:** Custom, Start: 2020/11/11 06:23:48, End: 2021/05/30 06:00:53.
- Table Headers:** Log Type, Malcolm Data Source, Malcolm Node, Originating Host, Originating GeoIP Country, Originating GeoIP City, Responding Host, Responding GeoIP Country, Responding GeoIP City, Originating Port, Responding Port, Related IP, Protocol, Service, Service Version, Action, Result, Severity, Risk Score, Severity Tags, File Magic.
- Table Data:** Showing 1 - 50 of 12,150 entries.
- Log Entries:**
  - Log Type: http
  - Malcolm Data Source: zeek
  - Malcolm Node: filebeat
  - Originating Host: 217.226.31.170
  - Originating GeoIP Country: Germany
  - Originating GeoIP City: Bremen
  - Responding Host: 124.106.97.191
  - Responding GeoIP Country: Philippines
  - Responding GeoIP City: Santa Elena
  - Originating Port: 4230
  - Responding Port: 80
  - Related IP: 217.226.31.170 124.106.97.191
  - Protocol: tcp
  - Service: http
  - Service Version: 1.1
  - Action: GET
  - Result: Bad Gateway
  - Severity: 20
  - Risk Score: 20
  - Severity Tags: External traffic
  - File Magic: text/html

Zeek http.log

The screenshot shows the Zeek http.log interface with the following details:

- Navigation Bar:** Sessions, SPIView, SPIGraph, Connections, Hunt, Files, Stats, History, Settings, Users.
- Search Bar:** Pipeline Depth, Request Method, URI, Version.
- Table Headers:** Pipeline Depth, Request Method, URI, Version.
- Table Data:** 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

```
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 interface with the following details:

- Top Navigation:** Sessions, SPIView, SPIGraph, Connections, Hunt, Files, Stats, History, Settings, Users.
- Search Bar:** country != US && protocols == http
- Filter Bar:** Custom, Start: 2021/02/28 23:59:11, End: 2021/03/01 00:28:26, Bounding, Last Packet, Interval: Auto, Duration: 00:29:15.
- Session View Buttons:** Open Items, Visible Items, Matching Items, Include: same time period, linked segments (slow), Filename: US\_HTTP.pcap.
- Bottom Buttons:** Export PCAP, Page: 1 of 120.
- Timeline Chart:** Shows packet activity over time from 2021/03/01 00:00:00 to 2021/03/01 00:28:00. The chart includes a search bar, zoom controls (10%), and various data series like Session, Packets, Bytes, Data bytes, Lines, Bars, and Cap Restarts.
- World Map:** A world map showing network traffic distribution.
- Bottom Filter Bar:** Protocols: tcp, http; Start Time: 2021/03/01; Stop Time: 2021/03/01; Src IP / Country: 10.0.52.164; Src Port: 2550; Dst IP / Country: 61.8.0.17; Dst Port: 80; Packets: 7,195; Databytes / Bytes: 5,160,414; Tags: HTTP, out-of-order-dst; Info: 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)



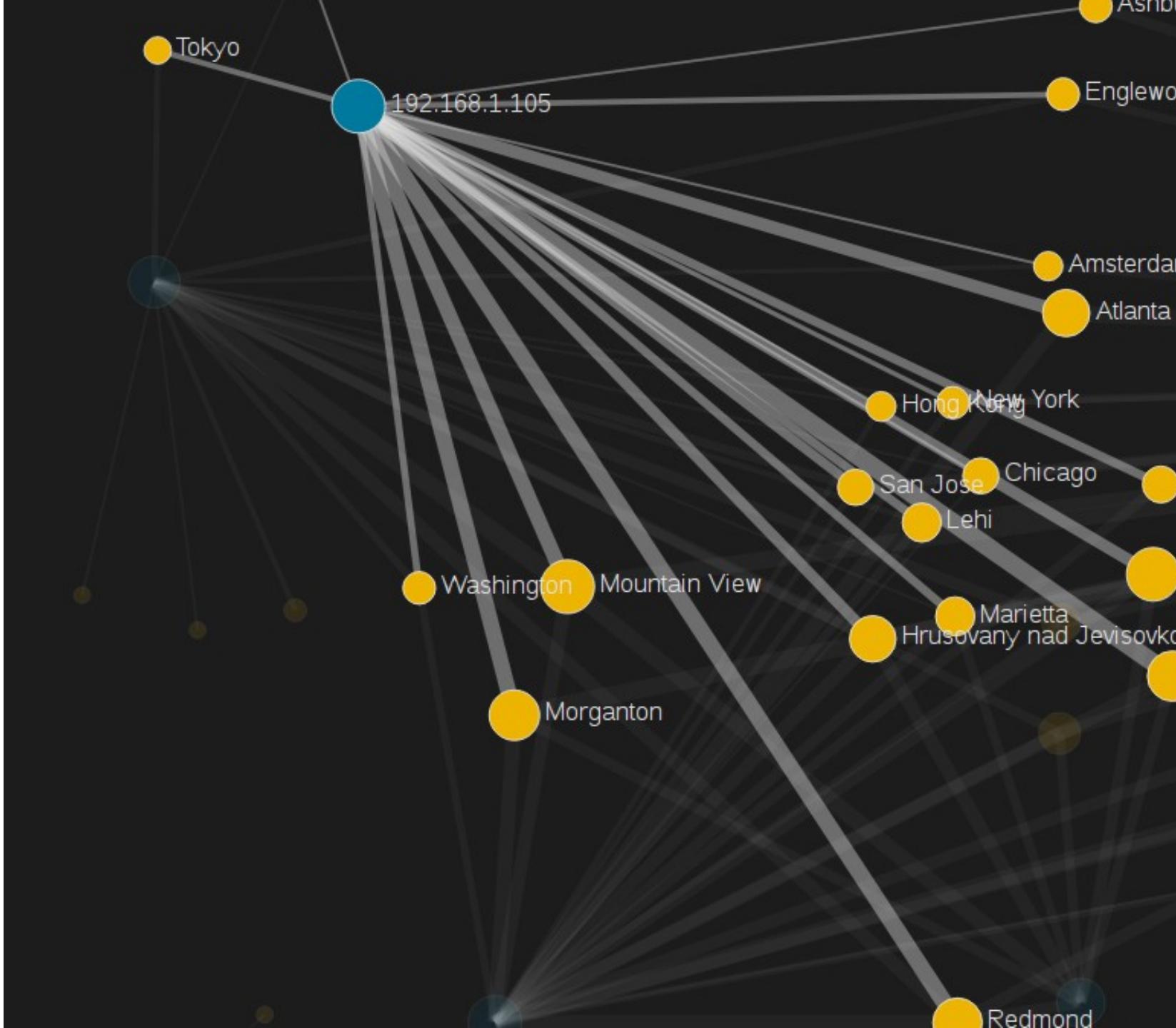
# SPIGraph

- 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

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

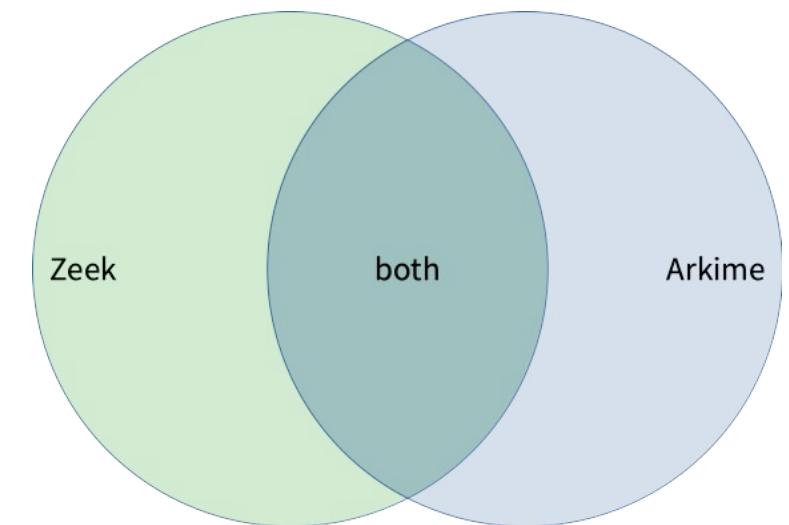
Search your packet search job history 50 per page 1 Showing 1 - 1 of 1 entries

Status	Matches	Name	User	Search text	Notify	Created	ID
<input checked="" type="checkbox"/> 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

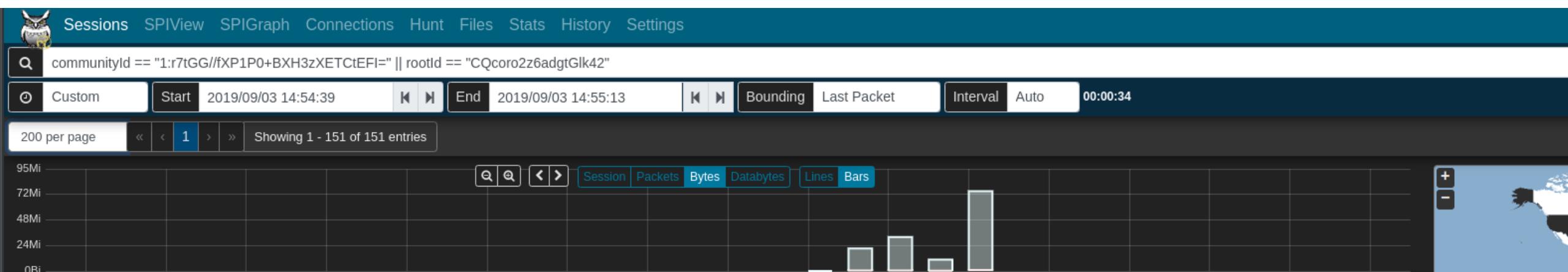
- Search syntax is different between Arkime and Dashboards (and in some cases, so are field names)
  - See search syntax comparison table, Malcolm and Arkime docs
- Despite considerable overlap, there are differences in protocol parser support between Zeek and Arkime
  - Learning the strengths of each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



# Correlate Zeek Logs and Packet Payloads

- Correlate Zeek logs and Arkime sessions using common fields
- communityId fingerprints flows in both and can bridge the two
- rootId / zeek.uid filters Zeek logs for the same session
- Filter community ID OR'ed with Zeek UID to see all Arkime sessions and Zeek logs for the same traffic

```
communityId == "1:r7tGG//fXP1P0+BXH3zXETCtEFI=" || rootId == "CQcoro2z6adgtGlk42"
```



# 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

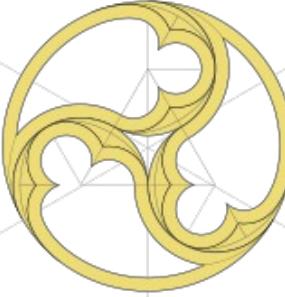
- Signatures dashboard in Dashboards shows scanned file hits
- Use `zeek.fuid` field in *Signatures - Logs* table to pivot to connection UID (`zeek.uid`) and other logs with pertinent session details



# 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 (1000+)
- Filter on `zeek.logType` (e.g., `conn` to see `conn.log`)
- Filter on `protocol` or both Arkime and Zeek 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.