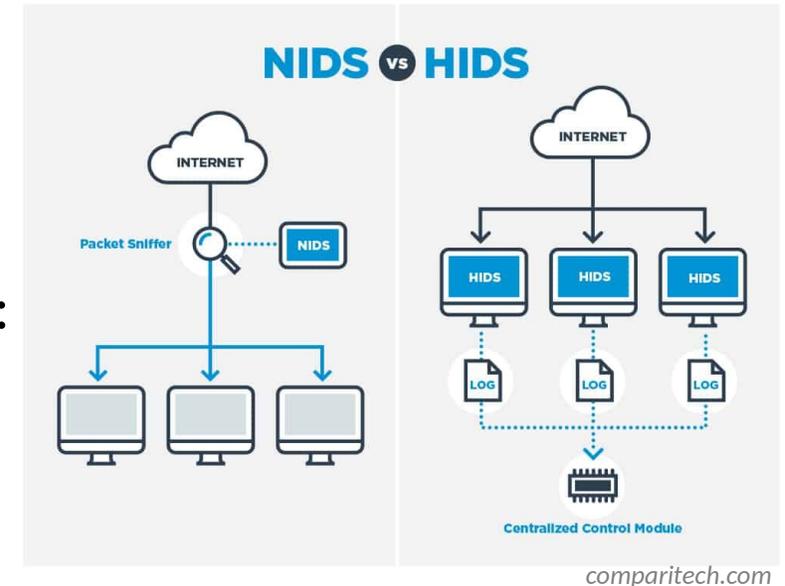


Network Traffic Analysis with Malcolm

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
 - 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

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

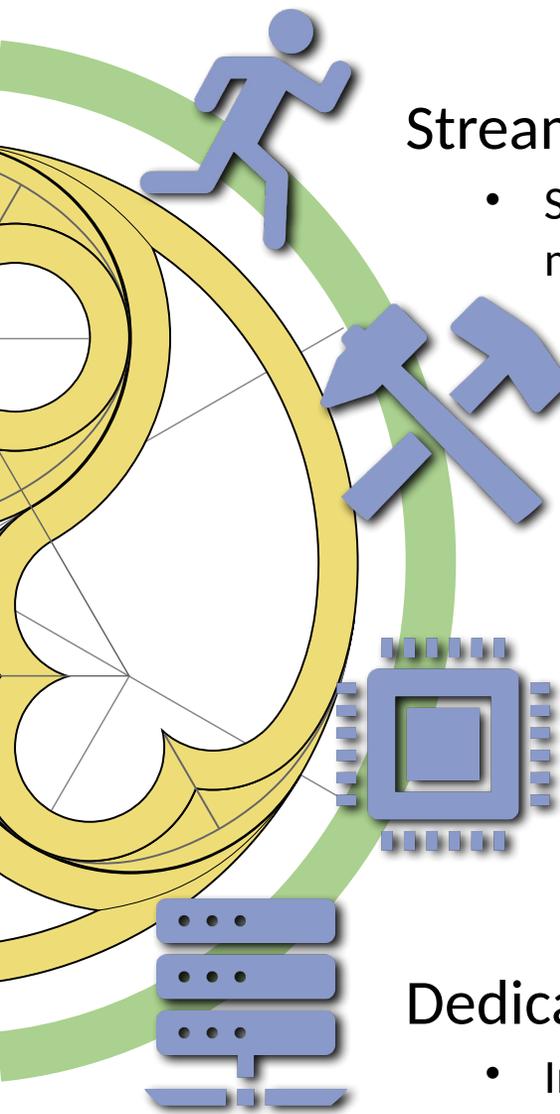
- Large scale index packet capture and search tool
- Packet analysis engine with support for many common IT protocols
- Web interface for browsing, searching, analysis and PCAP carving for exporting
- PCAP payloads (not just session header/metadata) are viewable and searchable

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, 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>



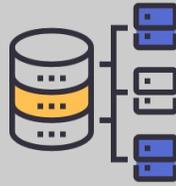
Capture



File Scanning



Forwarding &
Enrichment



Storage



Anomaly
Detection



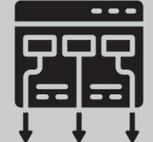
Alerting



Visualization



Payload
Analysis



Framework



zeek



yara



beats



OpenSearch



OpenSearch
Anomaly
Detection
Plugin



OpenSearch
Alerting
Plugin



OpenSearch
Dashboards



CyberChef



docker



Arkime



ClamAV



netsniff-ng



CAPA



logstash



TCPDUMP



VIRUSTOTAL



Arkime

Arkime
session PCAP
export to
WIRESHARK



NGINX

Malcolm

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)

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 (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

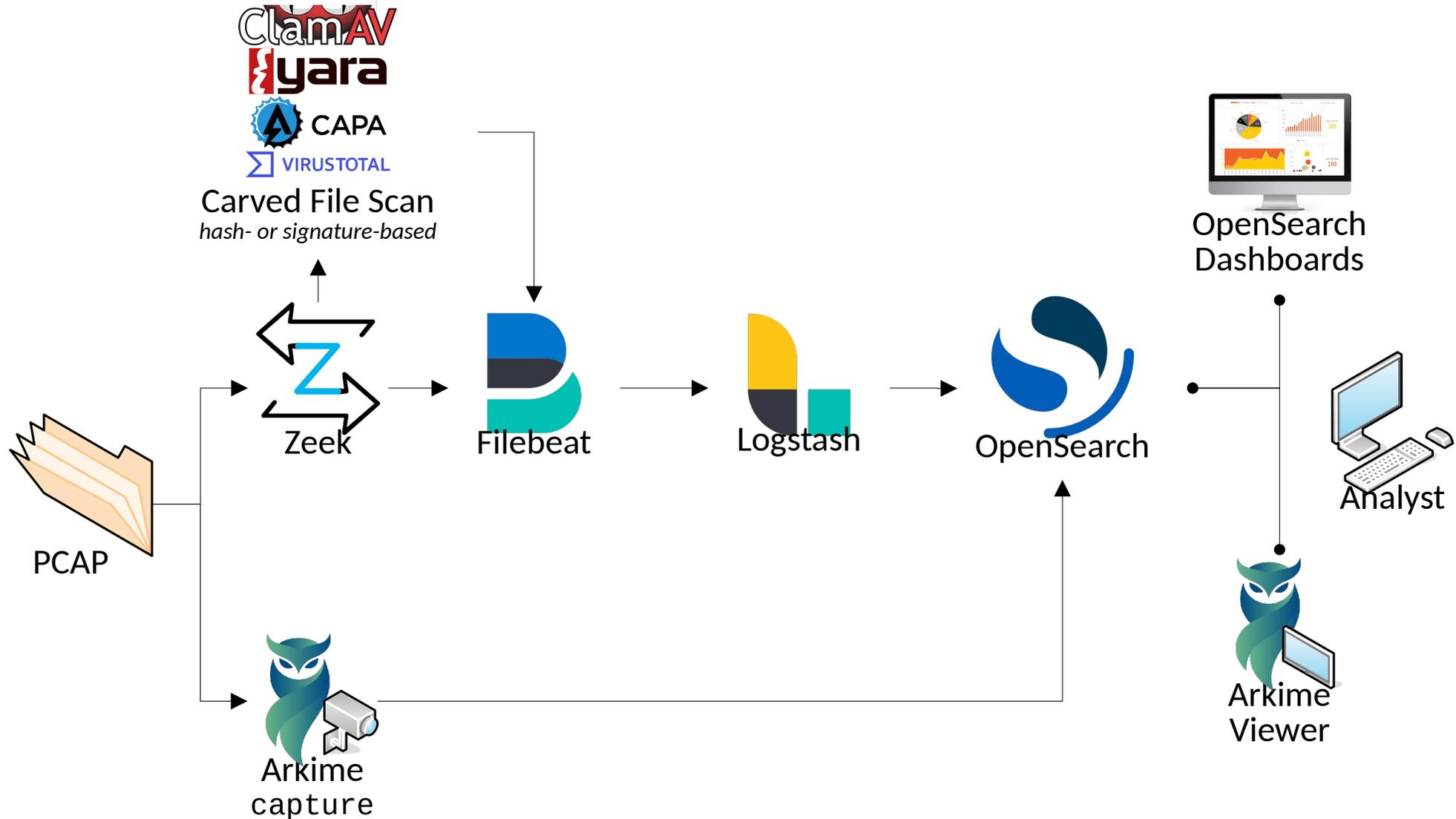
various tunnel protocols (e.g., GTP, GRE, Teredo, AYIYA, IP-in-IP, etc.)

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

Malcolm

Data Pipeline

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



Malcolm



Data Pipeline

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

Traffic is collected passively by the Hedgehog sensor device

- Zeek and Arkime Capture 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

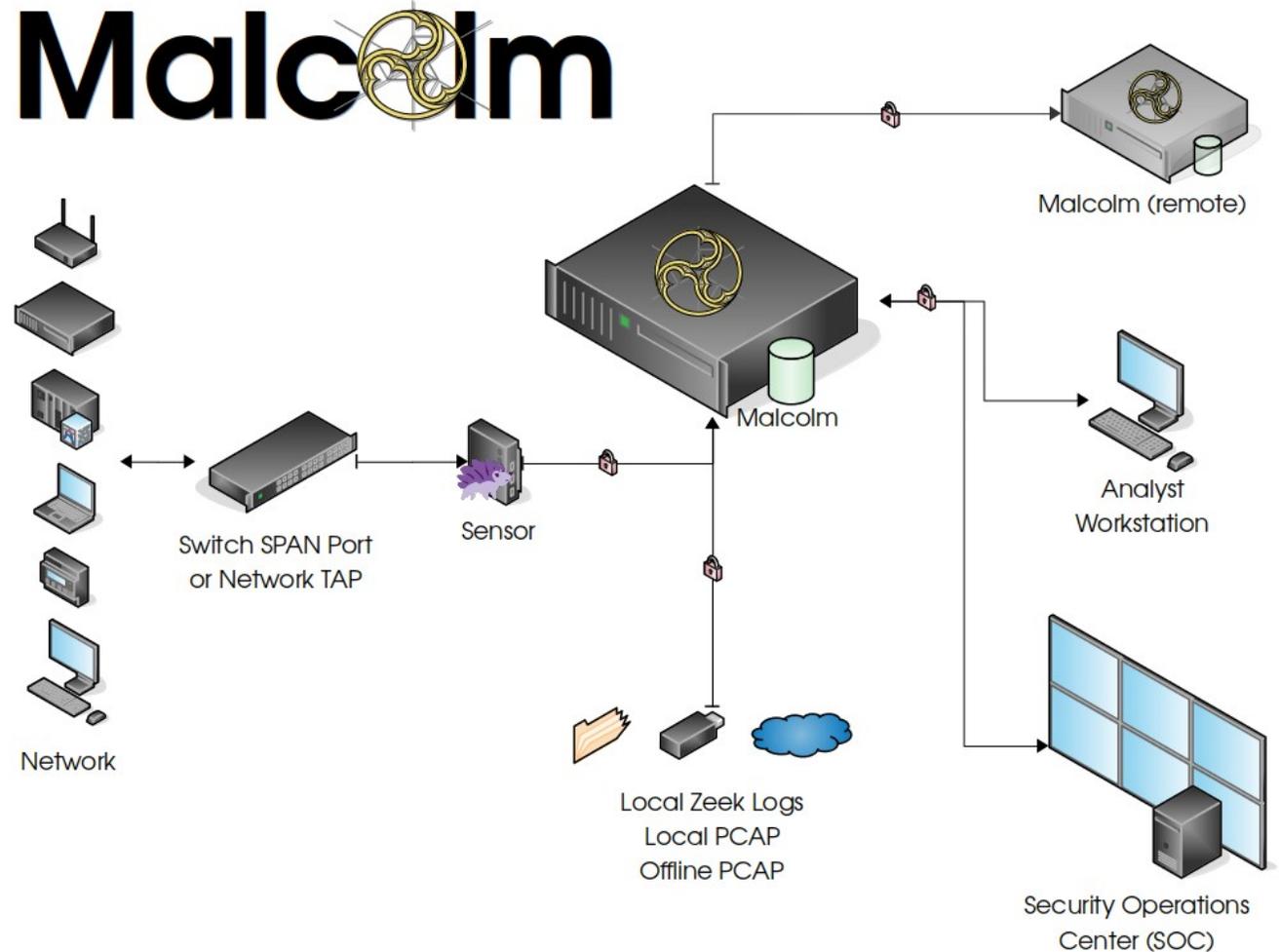
- Alerts may be triggered by exceeded thresholds, anomalies detected, custom queries, etc.

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
- 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 the following columns: Address, Name, Tag, and two action columns (edit and delete). The table contains the following data:

Address	Name	Tag	Edit	Delete
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			

At the bottom of the interface, there are input fields for "Address", "Name", and "Tag (optional)", along with a save icon.

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>

The screenshot shows the Manticore web interface for uploading traffic captures. At the top right, the text "Capture File and Log Archive" is visible. Below this, there are three buttons: "Add files..." (blue), "Start upload" (green), and "Cancel upload" (red). To the right of these buttons is a "Select all" checkbox. Below the buttons, there is a "Tags:" section with three green tags: "ACME x", "Field Office x", and "Incident XYZ x". Below the tags, there is a checkbox labeled "Analyze with Zeek" which is checked. Below this, there is a "Zeek File Extraction" dropdown menu with the selected option "Files with mime types of common attack vectors". Below the dropdown, there is a table of uploaded files with columns for filename and size. The table contains five rows of data.

m57patents-2009-11-13-0924.pcap	63.80 MB
m57patents-2009-11-14-0924.pcap	6.68 MB
m57patents-2009-11-15-0924.pcap	11.53 MB
m57patents-2009-11-16-0924.pcap	62.49 MB
m57patents-2009-11-16-1308.pcap	97.75 MB

20

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`



- 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>

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

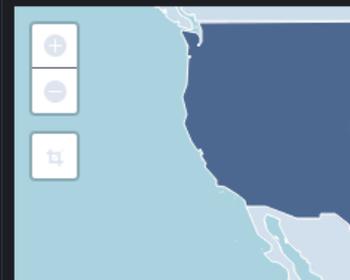
Application Protocol	Protocol Version	Count
ftp	-	1,063
smb	1	535
tftp	-	64
ntp	3	42
tls	TLSv10	38

Notices by Category

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

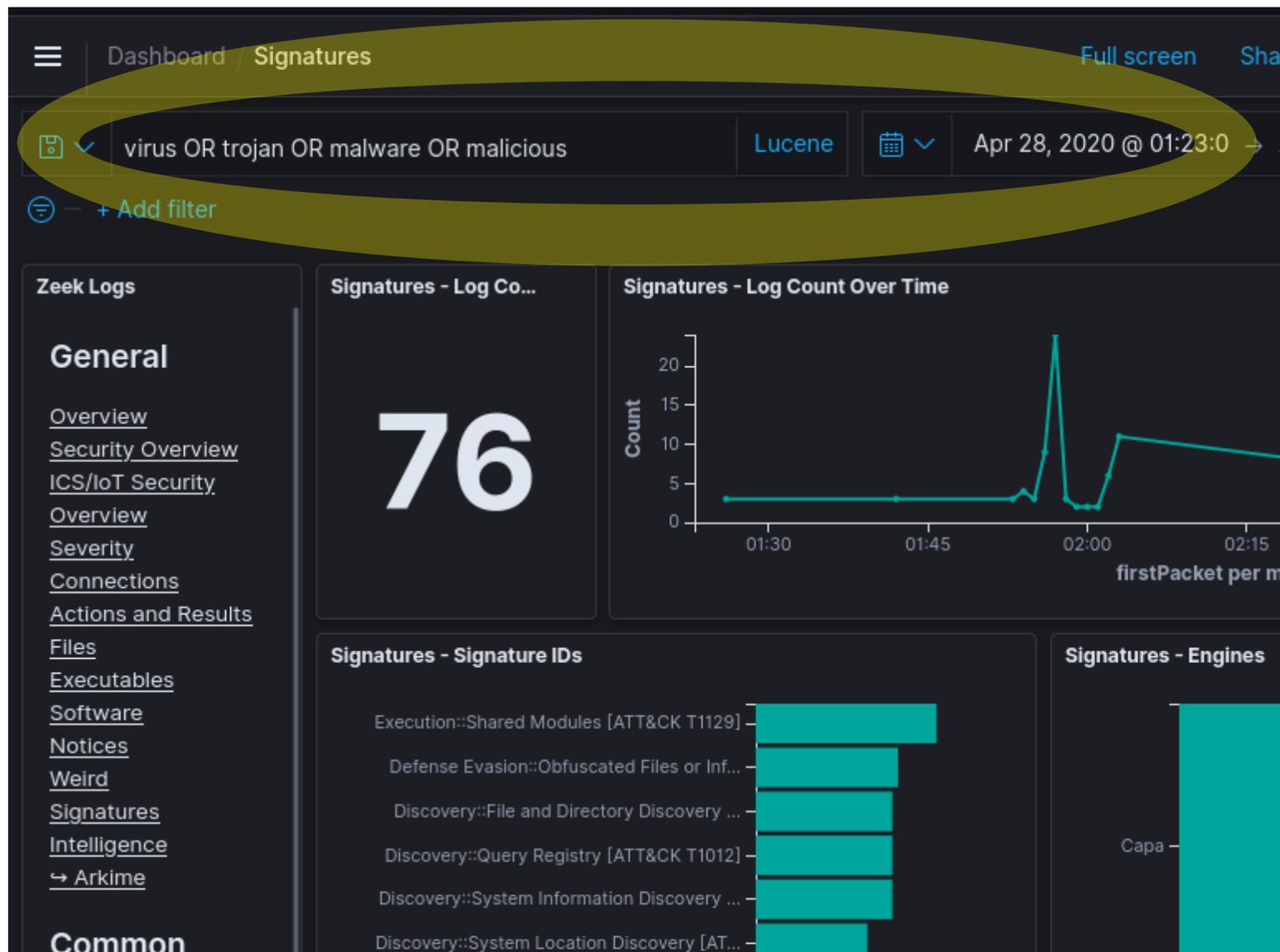
Export: [Raw](#) [Formatted](#)

Connections by Destination Country (r)



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 Overview dashboard. At the top, the Malcolm logo is displayed. Below it, a navigation bar includes a hamburger menu icon and the text "Dashboard / Security Overview". The main content area is divided into two columns. The left column, titled "Zeek Logs", contains a "General" section with a list of links: Overview, Security Overview, ICS/IoT Security Overview, Severity, Connections, Actions and Results, Files, Executables, Software, Notices, Weird, Signatures, Intel Feeds, and a link to Arkime. Below this is a "Common Protocols" section with a row of status indicators for DCE/RPC, DHCP, DNS, and FTP/TFTP. The right column, titled "Notices by Category", features a "Notice Category" dropdown menu and a list of categories including 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, and EternalSafety::Doub. At the bottom of the right column, the category "FTP::Bruteforcing" is visible.

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

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](#)

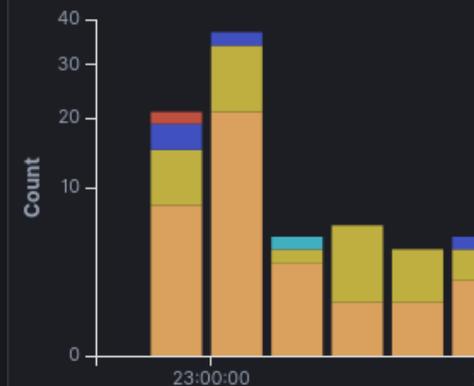
ICS/IoT Protocols

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

Notices - Log Count

108

Notices - Log Count Over Time



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
ftpt	-	64
ntp	3	42
tls	TLSv10	38

Notices by Category

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-ViolationNrRename	2
EternalSafety-ViolationTx2Cmd	2
ATTACK-Discovery	1
EternalSafety-DoublePulsar	1
FTP-Bruteforcing	1
Ripple20-Treck_TCP_observed	1

Export: Raw Formatted

Signatures - Signature IDs

Signature ID	Count
Defense Evasion-Obfuscated Files or Inf...	~100
Execution-Shared Modules [ATT&CK T1129]	~80
Discovery-Query Registry [ATT&CK T1012]	~70
Discovery-File and Directory Discovery ...	~60
Collection-Clipboard Data [ATT&CK T1115...	~50
Discovery-System Information Discovery ...	~45
Execution-Command and Scripting Interpr...	~40
Execution-Command and Scripting Interpr...	~35
Collection-Data from Information Reposi...	~30
Credential Access-Credentials from Pass...	~25
Defense Evasion-Deobfuscate/Decode File...	~20
Defense Evasion-Hide Artifacts-Hidden	~15
Discovery-Account Discovery [ATT&CK T10...	~10
Discovery-Process Discovery [ATT&CK T10...	~10
Discovery-Software Discovery [ATT&CK T1...	~10
Discovery-System Location Discovery [AT...	~10
Discovery-System Location Discovery:Sy...	~10
Privilege Escalation-Access Token Manip...	~10
smb-nt-transact-rename	~10
smb-nt-transact-rename-secondary	~10
Other	~10

Clear-text Transmission of Passwords

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

Export: Raw Formatted

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

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

Network Layer

ICS/IoT Log Counts

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

Connections by Destination Country (region map)

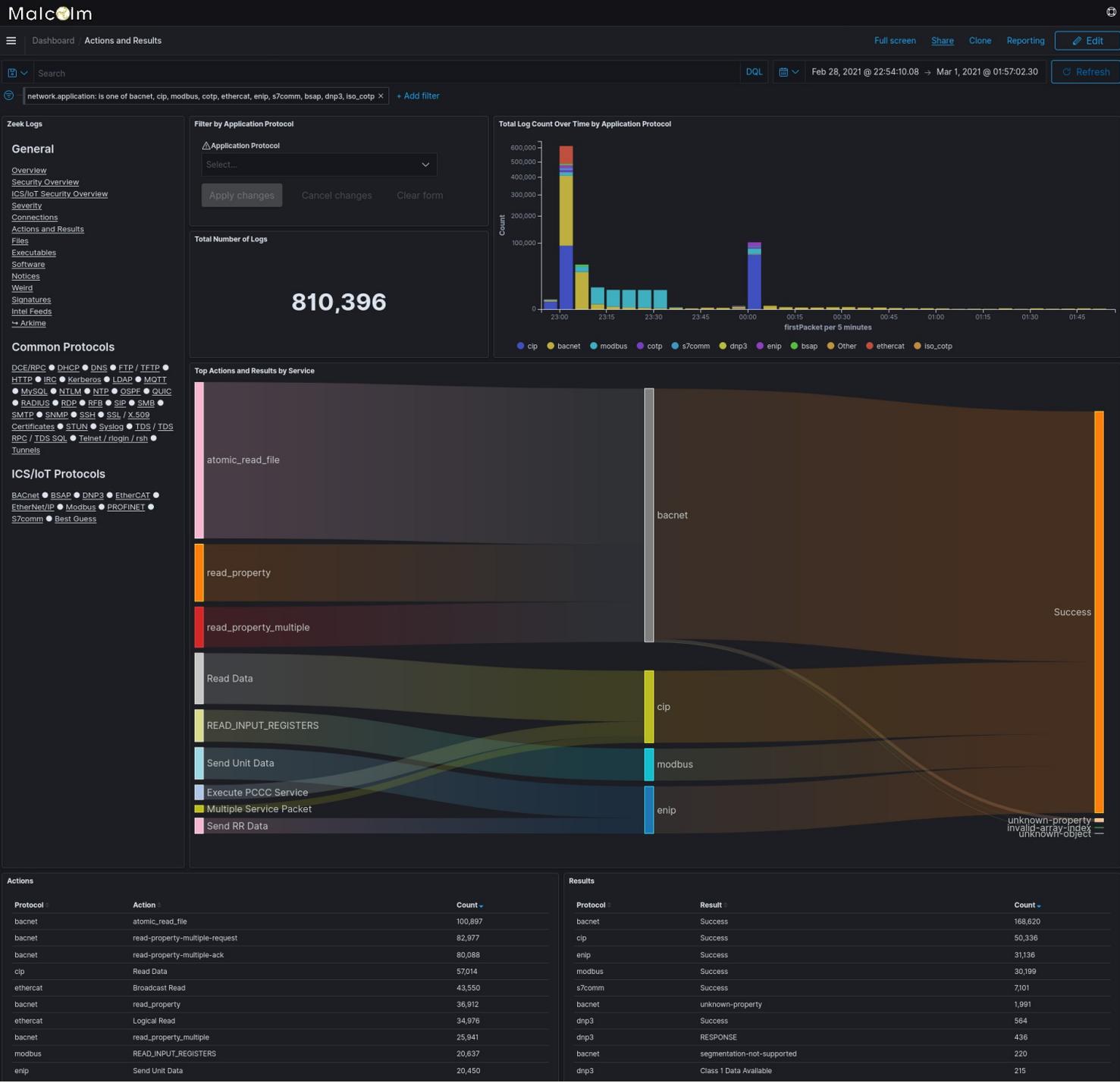
Inbound Connections

Origin	Count
Un	~100
Ne	~100
Ge	~100
Ca	~100
Ital	~100

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	134.249.62.100	Ukraine	134.249.61.163	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
 - OPCUA Binary
 - PROFINET
 - S7comm

[Intelligence](#)

[↔ 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](#) ● [OPCUA Binary](#) ●
[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

The screenshot displays the Splunk Discover interface. At the top, the search bar contains 'event.dataset:software'. The search results are 2,092 hits. A histogram shows the distribution of events over time, with a significant peak at 00:00. Below the histogram, a table lists the first packet per 10 minutes.

Search: 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	104.117.42.242	HTTP::SERVER	letty/	letty(8.1.4.v20120524)	-

New Visualization

Q Filter

The image shows a grid of 20 visualization options in a dark theme. The options are arranged in five rows and four columns. The 'TSVB' option is highlighted with a white border and a hand cursor icon. The options are:

- Area
- Controls ^E
- Coordinate Map
- Data Table
- Gantt Chart
- Gauge
- Goal
- Heat Map
- Horizontal Bar
- Line
- Markdown
- Metric
- Pie
- Region Map
- Sankey Diagram
- TSVB** (highlighted)
- Tag Cloud
- Timeline
- Vega
- Vertical Bar

Custom Visualizations

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

Search Syntax Comparison

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

Search Syntax Comparison (cont.)

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

Search Syntax Comparison (cont.)

	Arkime	Dashboards (Lucene)	Dashboards (DQL)
Global string search (anywhere in the document)	all Arkime search expressions are field-based	microsoft	microsoft
Wildcards	host.dns == "*micro?oft*" (? for single character, * for any characters)	dns.host:*micro?oft* (? for single character, * for any characters)	dns.host:*micro*ft* (* for any characters)
Regex	host.http == /. *www\.f.*k\.com.* /	zeek_http.host: /. *www\.f.*k\.com.* /	Dashboards Query Language does not currently support regex
IPv4 values	ip == 0.0.0.0/0	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)
GeoIP information available	<code>country == EXISTS!</code>	<code>_exists_:zeek.destination_geo OR _exists_:zeek.source_geo</code>	<code>zeek.destination_geo:* or zeek.source_geo:*</code>
Zeek log type	<code>zeek.logType == notice</code>	<code>zeek.logType:notice</code>	<code>zeek.logType:notice</code>
IP CIDR Subnets	<code>ip.src == 172.16.0.0/12</code>	<code>srcIp:"172.16.0.0/12"</code>	<code>srcIp:"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_:zeek.destination_geo OR _exists_:zeek.source_geo</code>	<code>zeek.destination_geo:* or zeek.source_geo:*</code>

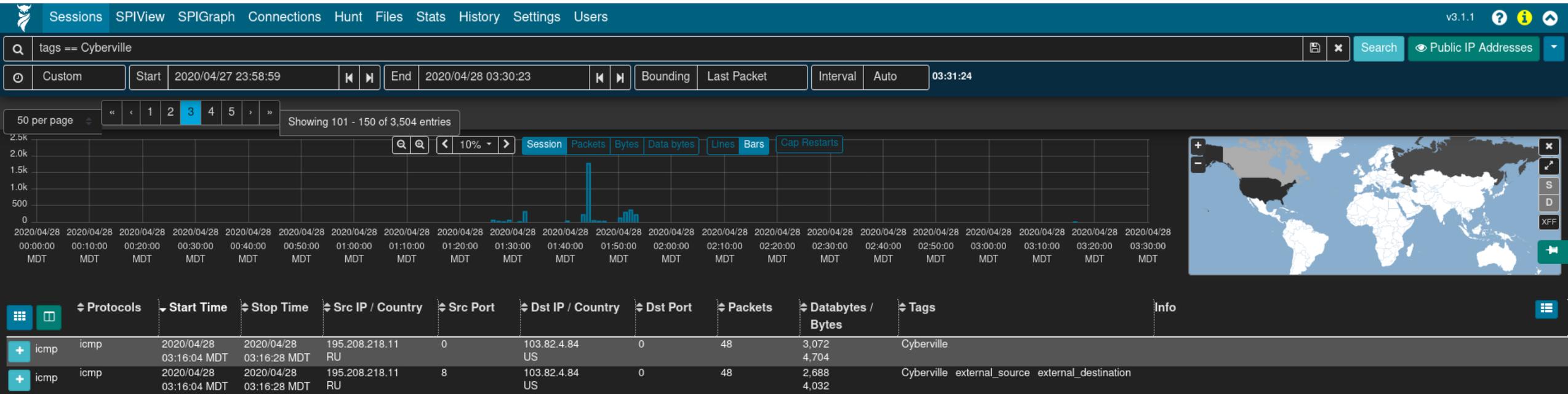
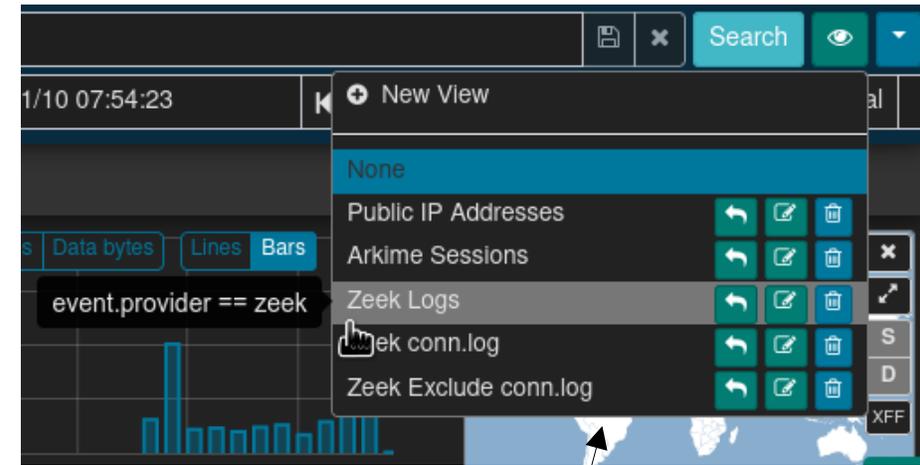


Arkime

- 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>

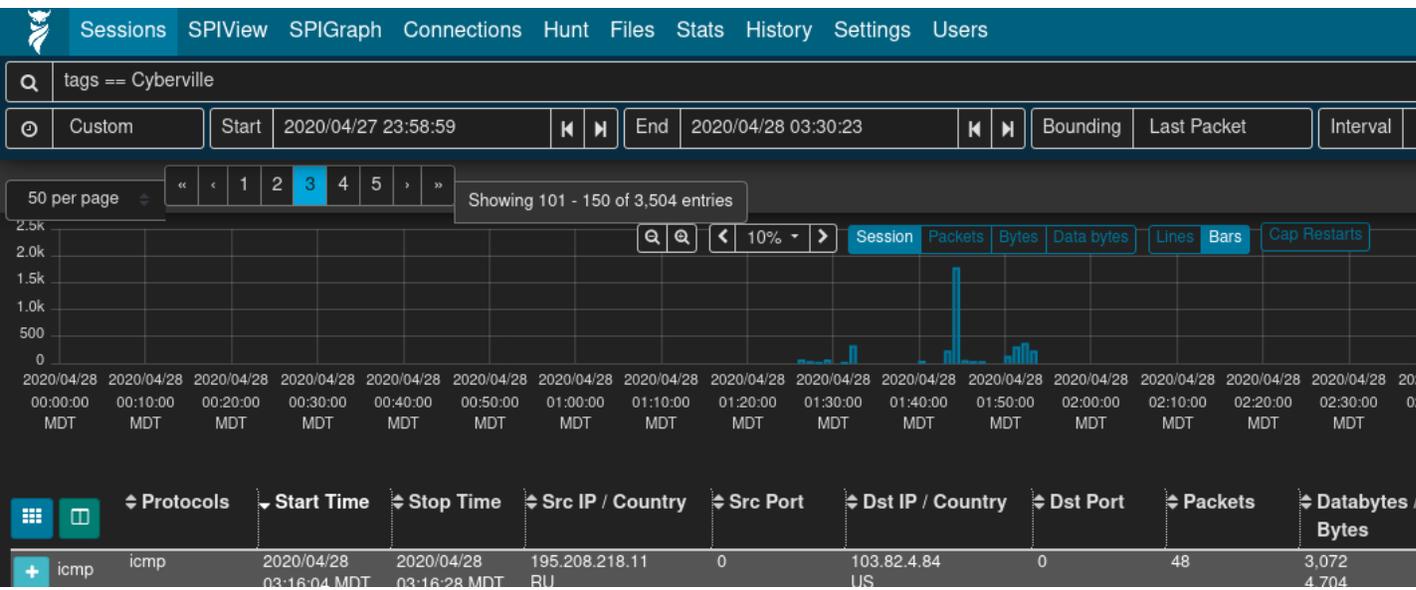
Arkime Filters and Search

- 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



Sessions

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



The screenshot shows the field-level details 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 showing 1 - 50 of 12,150 entries. The details include:

- 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 PCAnyPass.exe

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

Export PCAP

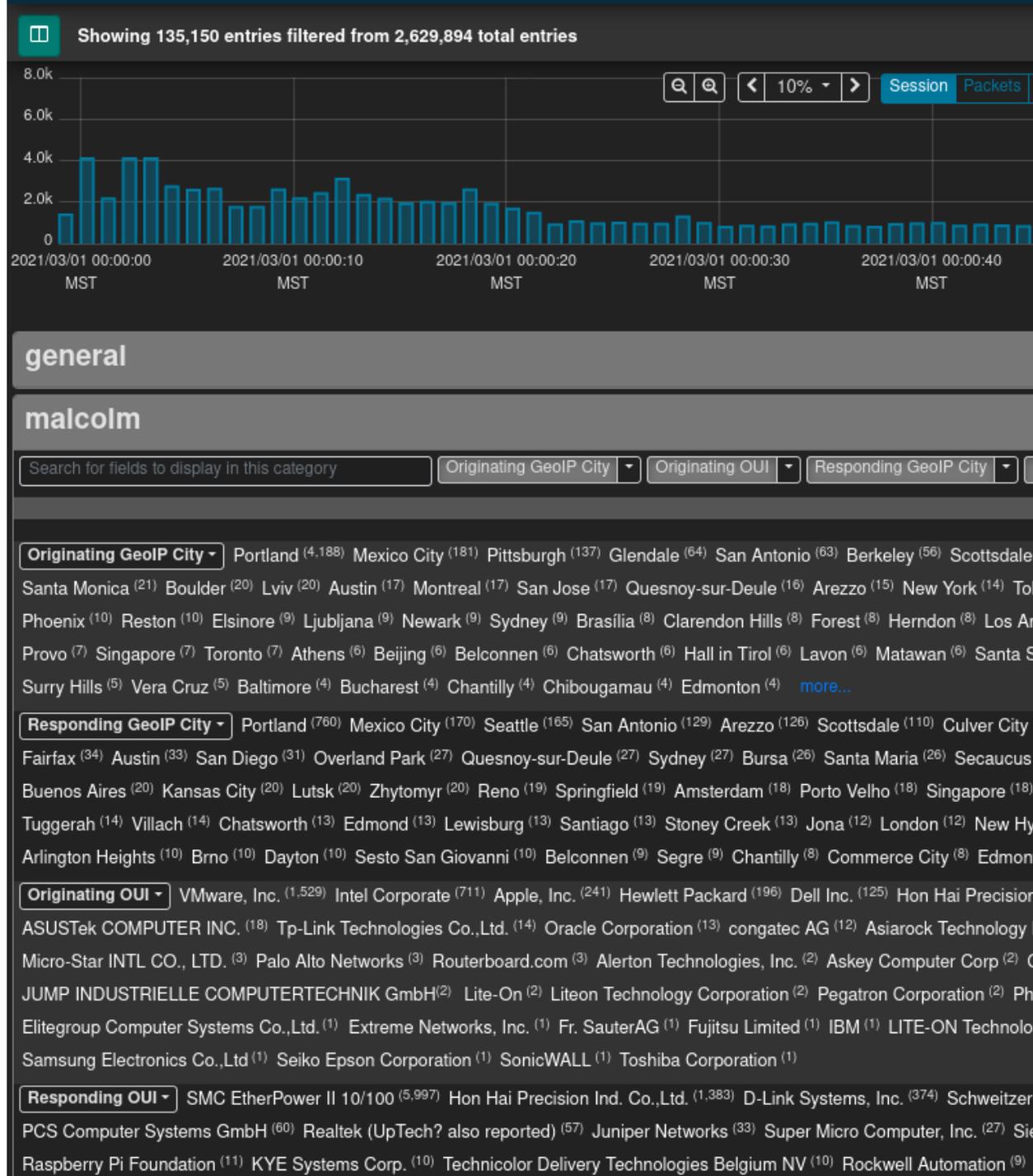
- Creates a new PCAP file from filtered sessions
- Include open, visible or all matching sessions
- Apply “Arkime Sessions” view to sessions first
- Narrow as much as possible prior to exporting (huge PCAP files are a pain)

The screenshot displays the Arkime interface for exporting PCAP files. The search filter is set to "country != US && protocols == http". The time range is from 2021/02/28 23:59:11 to 2021/03/01 00:28:26. The filename is "US_HTTP.pcap". The interface shows a bar chart of session activity and a world map.

Protocols	Start Time	Stop Time	Src IP / Country	Src Port	Dst IP / Country	Dst Port	Packets	Databytes / Bytes	Tags	Info
tcp http	2021/03/01 00:00:00	2021/03/01 00:00:00	10.0.52.164	2550	61.8.0.17	80	7,195	5,160,414	HTTP out-of-order-dst	URI mirror.pacific.net.au/openoffice/stable/2.0.0/OOo_2.0.0_Win32Intel_install.exe

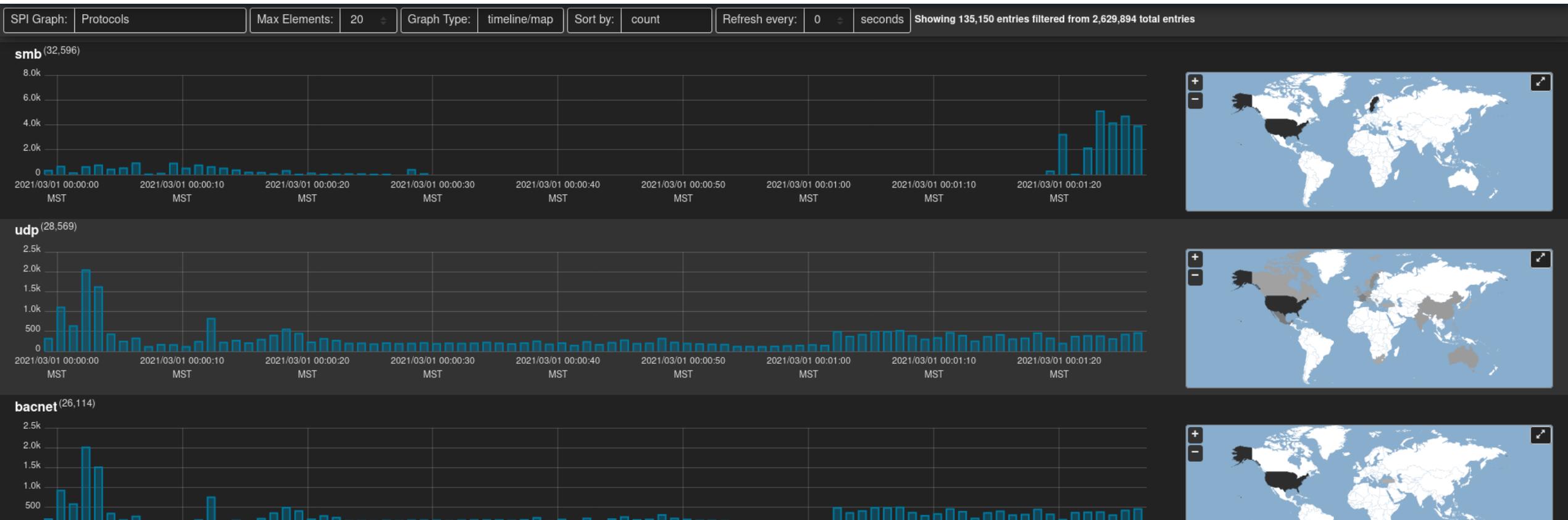
SPIView

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



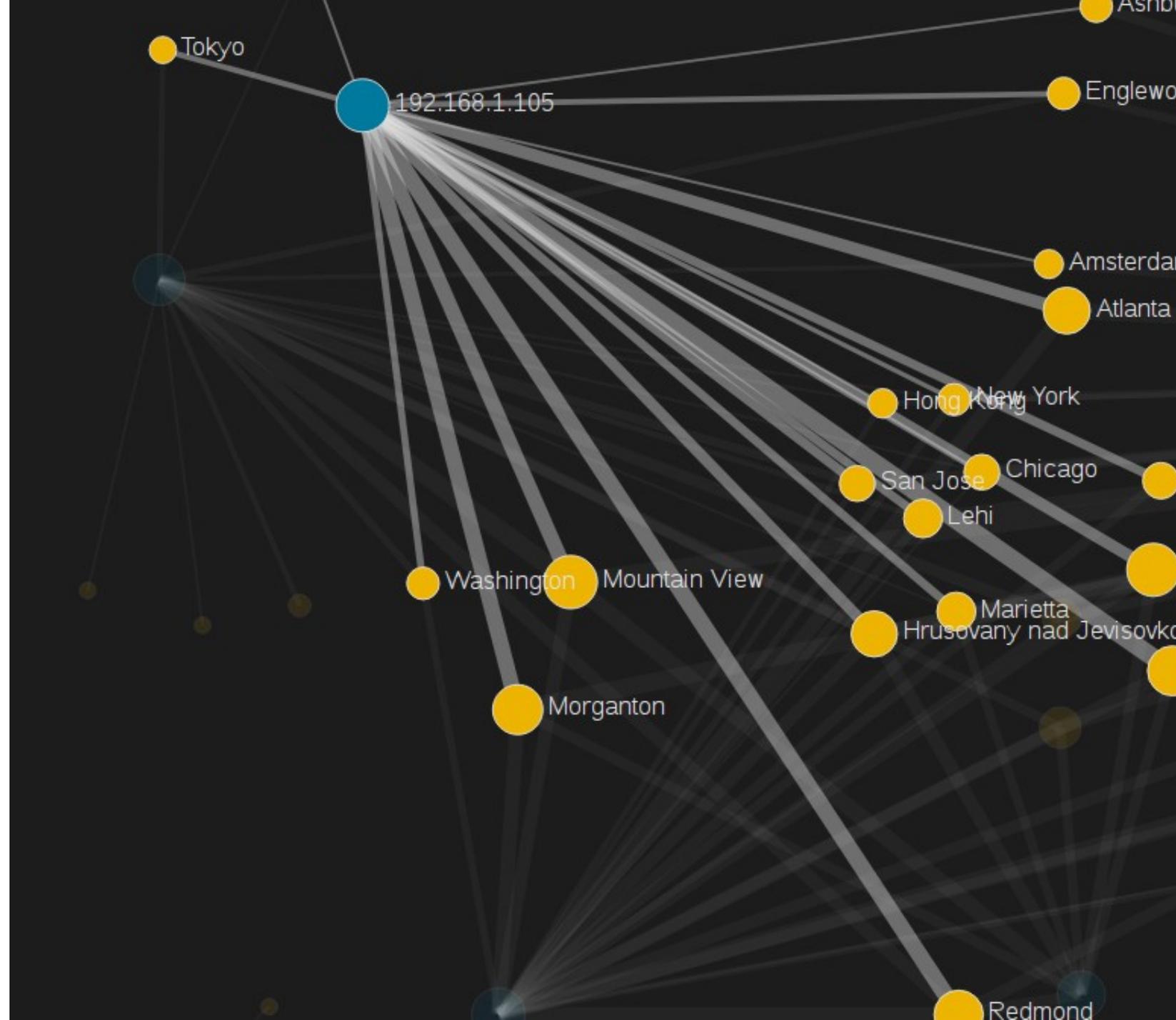
SPIGraph

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



Connections

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



Packet Search (“Hunt”)

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

The screenshot displays the Arkime Hunt interface. At the top, 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 range is from '1969/12/31 17:00:00' to '2021/12/06 12:10:02'. The bounding box is set to '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 for 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

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 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

✓ 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

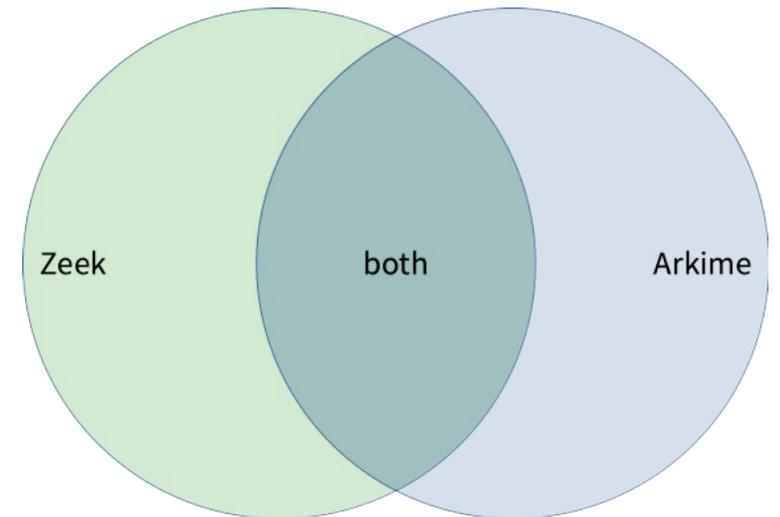
🔍 The sessions query expression was: **protocols == http**

🔍 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"
```

The screenshot shows the Arkime web interface. At the top, there are navigation tabs: Sessions, SPIView, SPIGraph, Connections, Hunt, Files, Stats, History, and Settings. Below the tabs is a search bar containing the query: `communityId == "1:r7tGG//fXP1P0+BXH3zXETCtEFI=" || rootId == "CQcoro2z6adgtGlk42"`. Below the search bar are controls for the search: a dropdown menu set to "Custom", a "Start" time of 2019/09/03 14:54:39, an "End" time of 2019/09/03 14:55:13, a "Bounding" button, a "Last Packet" button, an "Interval" dropdown set to "Auto", and a timer showing 00:00:34. Below these controls is a pagination bar showing "200 per page" and "Showing 1 - 151 of 151 entries". At the bottom, there is a bar chart with a y-axis ranging from 0Bi to 95Mi. The chart has tabs for "Session", "Packets", "Bytes", "Databytes", "Lines", and "Bars". The "Bytes" tab is selected, and the chart shows a single bar representing the total data size. On the right side of the chart, there are zoom controls (+ and -) and a small map of the world.

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.