

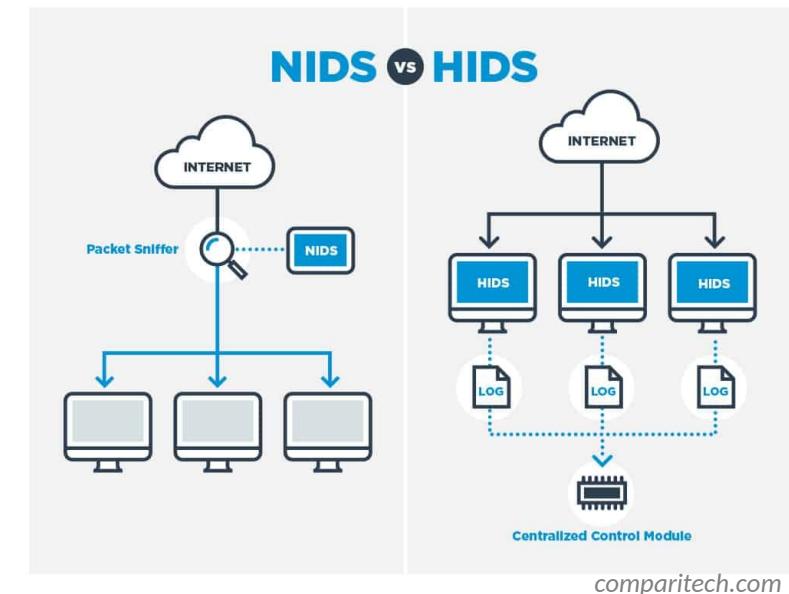
Network Traffic Analysis with **Malcolm**

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

Malcolm Development Team • Cybersecurity R&D • Idaho National Lab

Intrusion Detection Systems

- HIDS: Host Intrusion Detection Systems
 - Agents run on individual hosts or devices on a network
- 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

The diagram illustrates four Zeek log files: `conn.log`, `http.log`, `files.log`, and `pe.log`. Each file is represented by a table with columns for Field, Type, and Description. Red circles highlight specific fields in each table.

- conn.log | IP, TCP, UDP, ICMP connection details**
 - Field: `id` (Type: string, Description: Unique ID of the connection)
 - Field: `src_ip` (Type: string, Description: Originating endpoint's IP address:string)
 - Field: `src_port` (Type: port, Description: Originating endpoint's TCP/UDP port)
 - Field: `dst_ip` (Type: string, Description: Receiving endpoint's IP address:string)
 - Field: `dst_port` (Type: port, Description: Receiving endpoint's TCP/UDP port)
 - Field: `proto` (Type: proto, Description: Transport layer protocol of connection)
 - Field: `service` (Type: string, Description: Detailed application protocol, if any)
 - Field: `duration` (Type: time, Description: Length of connection)
 - Field: `http_bytes` (Type: count, Description: Total bytes from sequence number:HTTP)
 - Field: `tcp_bytes` (Type: count, Description: Total bytes from sequence number:TCP)
 - Field: `www_state` (Type: string, Description: Connection state (see `state`, `www_state`))
 - Field: `first_orig` (Type: bool, Description: Is this in its original ref?)
 - Field: `last_orig` (Type: bool, Description: Is this in its original ref?)
 - Field: `received_bytes` (Type: count, Description: Total bytes received due to content-gz)
 - Field: `memory` (Type: string, Description: See `pe_file`, `mem`)
 - Field: `avg_pkts` (Type: count, Description: Number of CNG packets)
 - Field: `avg_pkts` (Type: count, Description: Number of CNG packets (See P and length header field))
 - Field: `max_pkts` (Type: count, Description: Number of flows (flows))
 - Field: `max_pkts` (Type: count, Description: Number of flows (flows) (See P and length header field))
 - Field: `forward_packets` (Type: int, Description: # forwarded connection (UO of initiator during sampling))
 - Field: `http_id` (Type: string, Description: Unique identifier of the initiator)
 - Field: `http_ip` (Type: string, Description: Long ip address of the responder)
 - Field: `anon` (Type: int, Description: The user ID# for this connection)
 - Field: `anon_ip` (Type: int, Description: The user IP# for this connection)
- http.log | HTTP request/reply details**
 - Field: `id` (Type: string, Description: Unique ID of the HTTP request)
 - Field: `httpd_id` (Type: string, Description: Underlying connection id# - See `conn.log`)
 - Field: `trans_depth` (Type: count, Description: Pipelined depth into the connection)
 - Field: `method` (Type: string, Description: HTTP Request Method (GET, POST, HEAD, etc))
 - Field: `host` (Type: string, Description: Value of the Host header)
 - Field: `uri` (Type: string, Description: URI used in the request)
 - Field: `referer` (Type: string, Description: Value of the Referer header)
 - Field: `user_agent` (Type: string, Description: Value of the User-Agent header)
 - Field: `response_body_hex` (Type: string, Description: Uncompressed content-use of hexig-data)
 - Field: `response_body_hex` (Type: string, Description: Uncompressed content-use of hexig-data)
 - Field: `status_code` (Type: string, Description: Status code returned by the server)
 - Field: `status_msg` (Type: string, Description: Status message returned by the server)
 - Field: `info_code` (Type: string, Description: Last seen fail info ready code by server)
 - Field: `info_gong` (Type: string, Description: Last seen fail info ready message by server)
 - Field: `tags` (Type: set, Description: Indicators of various attributes discovered)
 - Field: `username` (Type: string, Description: Username if user auth is performed)
 - Field: `password` (Type: string, Description: Password if user auth is performed)
 - Field: `present` (Type: set, Description: Headers indication of a present request)
 - Field: `req_headers` (Type: vector, Description: The unique On-the-Fly req headers)
 - Field: `req_thunks` (Type: vector, Description: The unique On-the-Fly thunks)
 - Field: `req_name_types` (Type: vector, Description: The types from the req)
 - Field: `http_pkts` (Type: vector, Description: The unique On-the-Fly http_pkts)
 - Field: `http_thunks` (Type: vector, Description: The unique On-the-Fly thunks)
 - Field: `http_name_types` (Type: vector, Description: The types from the http_pkts)
 - Field: `client_header` (Type: vector, Description: The names of HTTP readers sent by client)
 - Field: `server_header` (Type: vector, Description: The names of HTTP readers sent by host)
 - Field: `cookies_hex` (Type: vector, Description: The cookie names extracted from cookies)
 - Field: `url_hex` (Type: vector, Description: Variable names extracted from the URL)
- files.log | File analysis results**
 - Field: `id` (Type: string, Description: Unique identifier for the file)
 - Field: `file_hex` (Type: set, Description: Hexes that occurred in the file)
 - Field: `file_hexes` (Type: set, Description: Hexes that occurred in the file)
 - Field: `conn_with` (Type: set, Description: Connection UO# over which the transferred file was made)
 - Field: `source` (Type: string, Description: An identification of the source of the file)
 - Field: `target` (Type: string, Description: Target of the transfer to source (e.g., HTTP request target))
 - Field: `analysis` (Type: string, Description: The analysis method that found the signatures)
 - Field: `minm_hex` (Type: string, Description: The hex that occurred first)
 - Field: `filename` (Type: string, Description: Name of the file that was analyzed)
 - Field: `duration` (Type: count, Description: The duration that the file was analyzed)
 - Field: `local_orig` (Type: bool, Description: Did the data originate locally?)
 - Field: `is_orig` (Type: bool, Description: Was the file sent by the originator?)
 - Field: `source_hexes` (Type: count, Description: Number of hexes recorded in the analysis engine)
 - Field: `total_hexes` (Type: count, Description: Total number of hexes that occurred in the file)
 - Field: `missing_hexes` (Type: count, Description: Number of hexes in the stream missed)
 - Field: `corrupted_hexes` (Type: count, Description: Out of range hexes (types of the stream due to overflow))
 - Field: `streamed` (Type: bool, Description: If the analysis module ran at least once)
 - Field: `parent_file` (Type: string, Description: File O this was extracted from)
 - Field: `modified` (Type: string, Description: MD5/SHA1 hash of the file)
 - Field: `examined` (Type: string, Description: Last filename of examined files, if needed)
 - Field: `entropy` (Type: double, Description: Information density of the file contents)
- pe.log | Portable Executable (PE)**
 - Field: `id` (Type: string, Description: Current temporary)
 - Field: `file_hex` (Type: string, Description: The longer hex that the file was converted to)
 - Field: `machine` (Type: string, Description: The hex that the file was created on)
 - Field: `compile_id` (Type: string, Description: The hex that the file was created on)
 - Field: `os` (Type: string, Description: The measured operating system)
 - Field: `dependencies` (Type: string, Description: The hexes that are required for the file)
 - Field: `is_hex` (Type: bool, Description: Is the file an executable or just an object file?)
 - Field: `is_dwarf` (Type: bool, Description: Does the file contain debug information?)
 - Field: `is_executable` (Type: bool, Description: Does the file contain executable code)
 - Field: `is_pef` (Type: bool, Description: Does the file contain PE executable)
 - Field: `is_coff` (Type: bool, Description: Does the file contain COFF executable)
 - Field: `is_macho` (Type: bool, Description: Does the file contain Mach-O executable)
 - Field: `has_import_table` (Type: bool, Description: Does the file have an import table)
 - Field: `has_export_table` (Type: bool, Description: Does the file have an export table)
 - Field: `has_cmt_table` (Type: bool, Description: Does the file have an NT resource table)
 - Field: `has_debug_table` (Type: bool, Description: Does the file have a debug table)
 - Field: `section_size` (Type: count, Description: The size of the sections, in bytes)

corelight.com



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

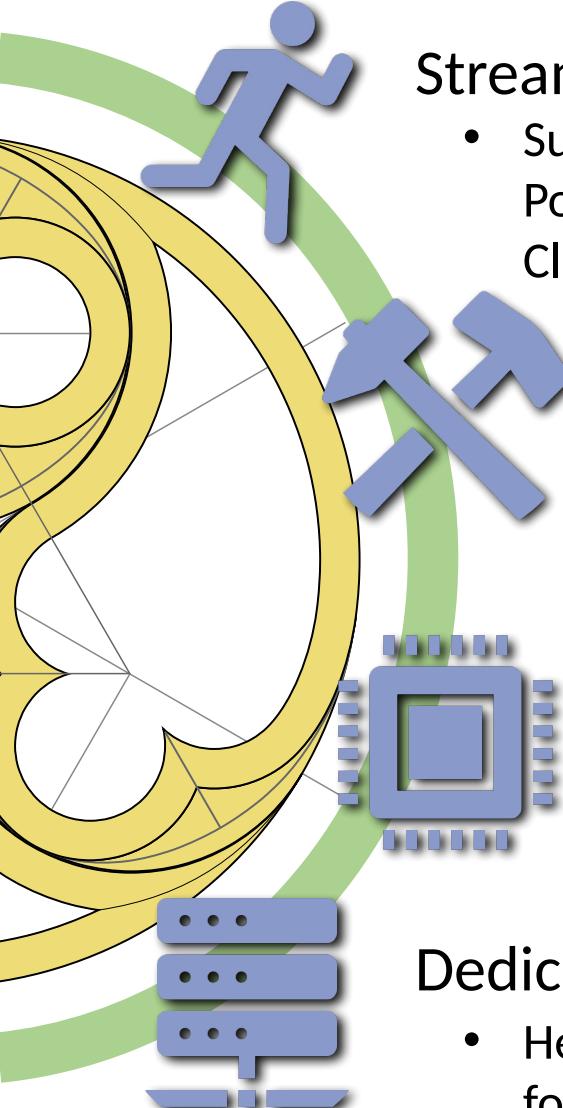
Weaknesses

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



A powerful open-source network traffic analysis tool suite.

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



Streamlined deployment

- Suitable for field use (hunt or incident response) or SOC deployment. Runs in Docker or Podman on Linux, macOS and Windows. ISO installer for bare metal installations. Cloud-deployable with Kubernetes. Provides easy-to-use web-based user interfaces.

Industry-standard tools

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

Expanding control systems visibility

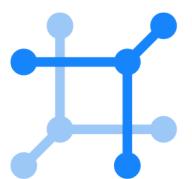
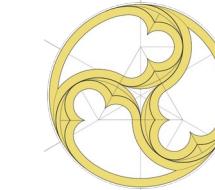
- Analyzes more protocols used in operational technology (OT) networks than other open-source or paid solutions. Ongoing development is focused on increasing the quantity and quality of industrial control systems (ICS) traffic.

Dedicated sensor appliance

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

Malcolm Origins and Milestones

- 2018.Q2 – Development begins under USBR/CISA work agreement
- 2018.Q3 to 2019.Q2 – Malcolm field tested at USBR facilities
- 2019.Q2 – Initial public release
- 2019.Q4 – Hedgehog Linux released
- 2021.Q1 – 1k st★rs on GitHub
- 2022.Q3 – First Malcolm-based simulated engagements at INL's ICS Control Environment Lab Resource (CELR)
- 2022.Q3 – Malcolm discussed during session of the U.S. House of Representatives Homeland Security Committee
- 2022.Q4 – NetBox provides asset inventory and interaction analysis
- 2023.Q1 – Kali announces “Purple” distro bundling Malcolm
- 2023.Q2 – Cloud deployable with K8s
- 2024.Q2 – 2k st★rs on GitHub, community discussions board and new training offerings
- 2024.Q3 – First public Malcolm user conference, Mal.Con24



Malcolm

What Can It Do For Me?

- Get to know your network: Malcolm **characterizes** traffic by devices and the protocols they use to communicate.
- Understand risks and threats: Malcolm **identifies** active exploits, potential attack vectors, and vulnerable devices and protocols.
- Increase visibility: Malcolm **highlights** inbound, outbound, and internal communications to inform decisions and improve security posture.



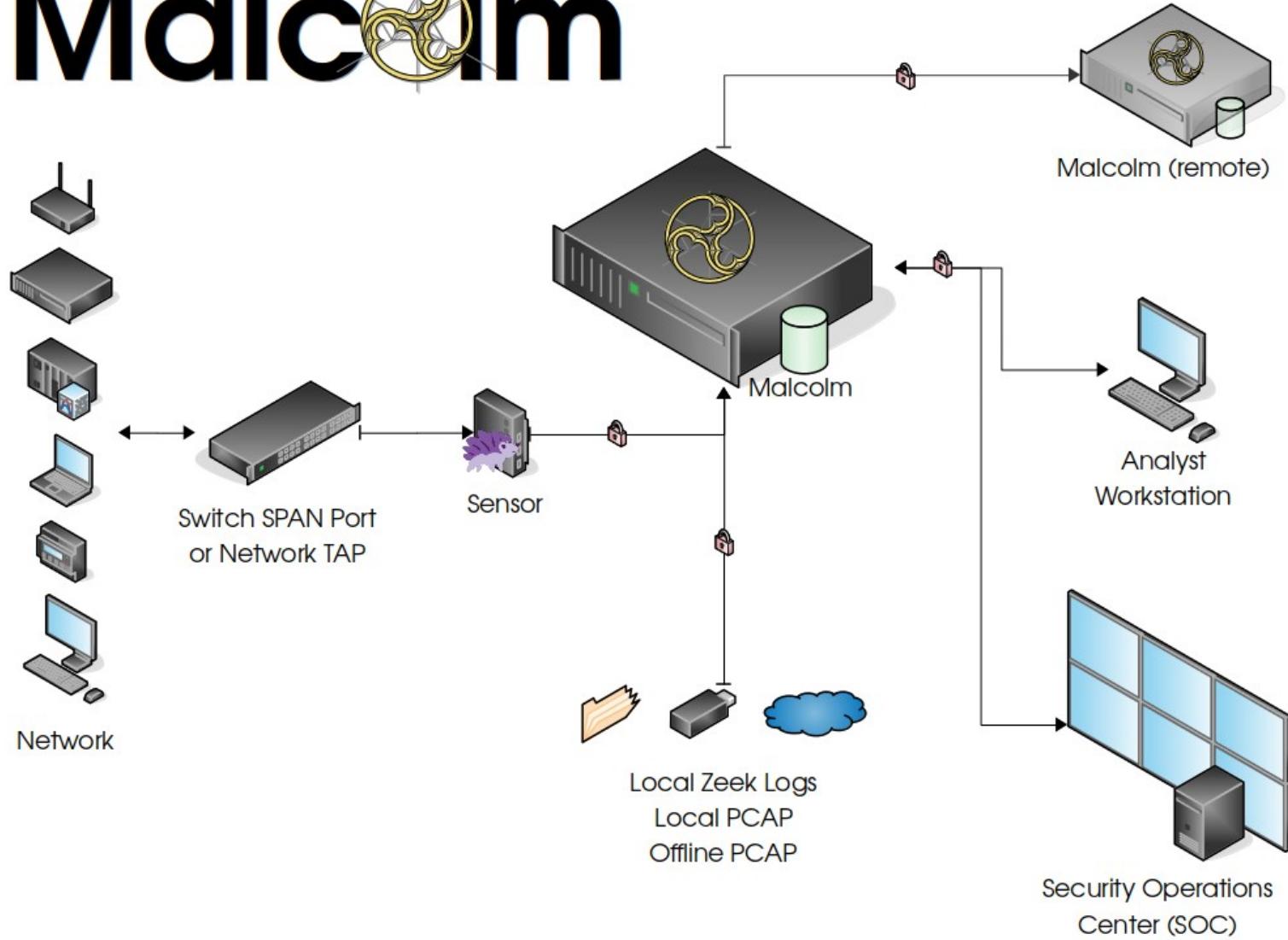
What can network traffic reveal about my cybersecurity posture?

- A lot!
- Malcolm can be used to validate cyber best practices and uncover red flags in network configuration, including:
 - Proper network segmentation
 - East-West (cross-segment) and North-South traffic
 - Unsecure or outdated protocols
 - Unexpected protocols
 - Rogue devices and services
 - Suspicious file transfers
 - Sensitive unencrypted information (e.g., PII, PCII, cleartext credentials)
 - ... and much more



Image credit: kelsercorp.com

Malcolm





Supported Protocols

<https://idaholab.github.io/Malcolm/docs/protocols.html>

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
GE Service Request Transport Protocol (SRTP)
Google Quick UDP Internet Connections (gQUIC)
Highway Addressable Remote Transducer over IP (HART-IP)
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)
Remote Operations Controller (ROC) Plus S7comm / Connection Oriented Transport Protocol (COTP)

Secure Shell (SSH)
Secure Sockets Layer (SSL) / Transport Layer Security (TLS)
Session Initiation Protocol (SIP)
Server Message Block (SMB) / Common Internet File System (CIFS)
Simple Mail Transfer Protocol (SMTP)
Simple Network Management Protocol (SNMP)
SOCKS
STUN (Session Traversal Utilities for NAT)
Synchrophasor (IEEE C37.118)
Syslog
Tabular Data Stream (TDS)
Telnet / remote shell (rsh) / remote login (rlogin)
TFTP (Trivial File Transfer Protocol)
WebSocket
WireGuard
various tunnel protocols (e.g., GTP, GRE, Teredo, AYIYA, IP-in-IP, etc.)

* *Operational Technology (OT) protocols indicated with **bold***

Malcolm



Components

<https://idaholab.github.io/Malcolm/docs/components.html>



Capture &
Analysis



File Scanning



Forwarding &
Enrichment



Storage



Anomaly
Detection



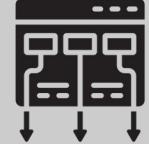
Asset
Management



Visualization



Payload
Analysis



Framework



TCPDUMP



VIRUSTOTAL



beats



OpenSearch



Anomaly
Detection
Plugin



Alerting



Alerting
Plugin



netbox



OpenSearch
Dashboards



CyberChef



podman



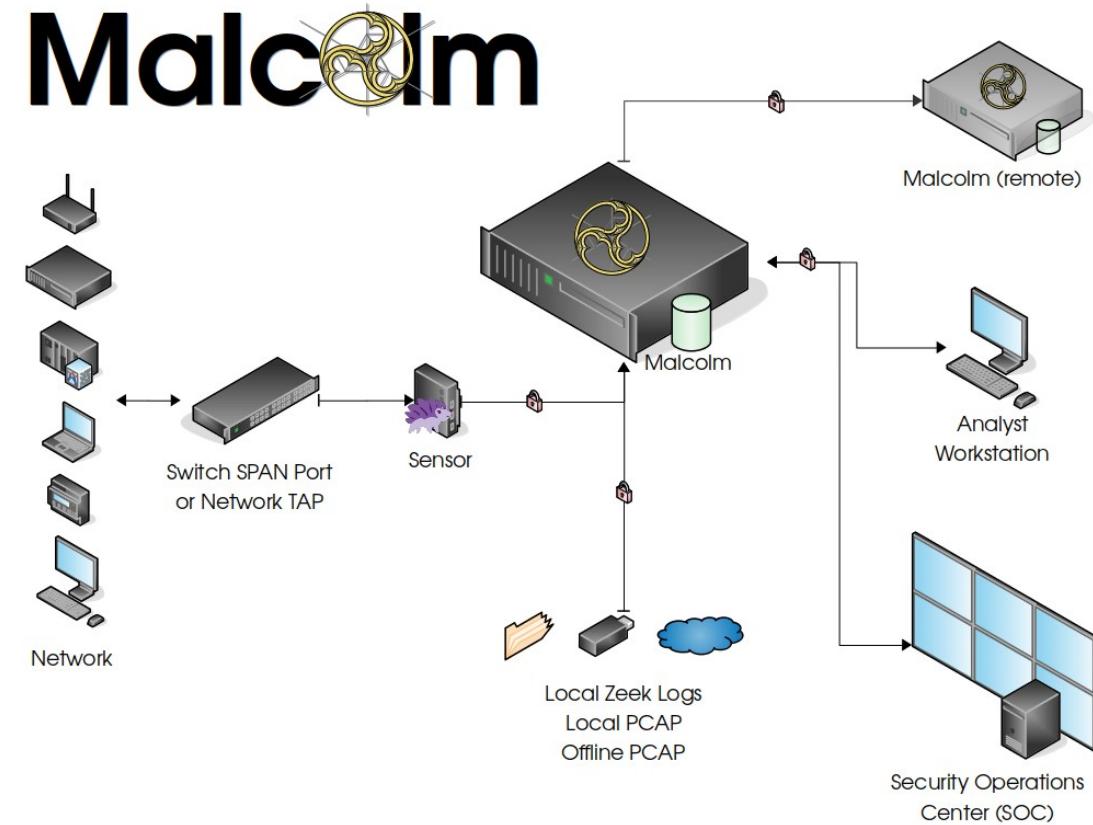
kubernetes



Arkime
session PCAP
export to
WIRESHARK

Configuring and Running Malcolm

- Runs natively in Docker or Podman, use ISO installer for VM or bare metal install, or cloud deploy with Kubernetes or Amazon Machine Image (AMI)
- Recommended system requirements: 64+GB RAM, 16+ CPU cores, “enough” storage for PCAP and logs
- Documentation and source code on GitHub: <https://idaholab.github.io/Malcolm>
- Walkthroughs on YouTube “Malcolm Network Traffic Analysis”





Dashboards

Visualize traffic or track down security concerns with dozens of [pre-built dashboards](#), or create your own



Arkime

Delve into [session details](#) including full packet payloads



NetBox

Model and document your [network infrastructure](#)



CyberChef

Slice and dice data with this web app for encryption, encoding, compression and data analysis



Documentation

Read the Malcolm user guide



Artifact Upload

[Upload](#) previously-captured PCAP files or archived Zeek logs for analysis



Keycloak Authentication

Malcolm is using [Keycloak](#) for authentication



Extracted Files

Browse the preserved [extracted files](#) carved and scanned by Malcolm

Importing Traffic Captures for Analysis

- Upload PCAP files or archived Zeek logs
 - pcapng not supported yet
- Specify tags for search and filter
- Specify NetBox site



Network Traffic Artifact Upload

Field Office Incident XYZ User-defined tags

Commit Uploaded Files

Drag & Drop your files or [Browse](#)

Advantech.pcap
40 KB

BACnet_FIU.pcap
9 MB

BACnet_Host.pcap
1.8 MB

iFix_Client86.pcap
900 KB

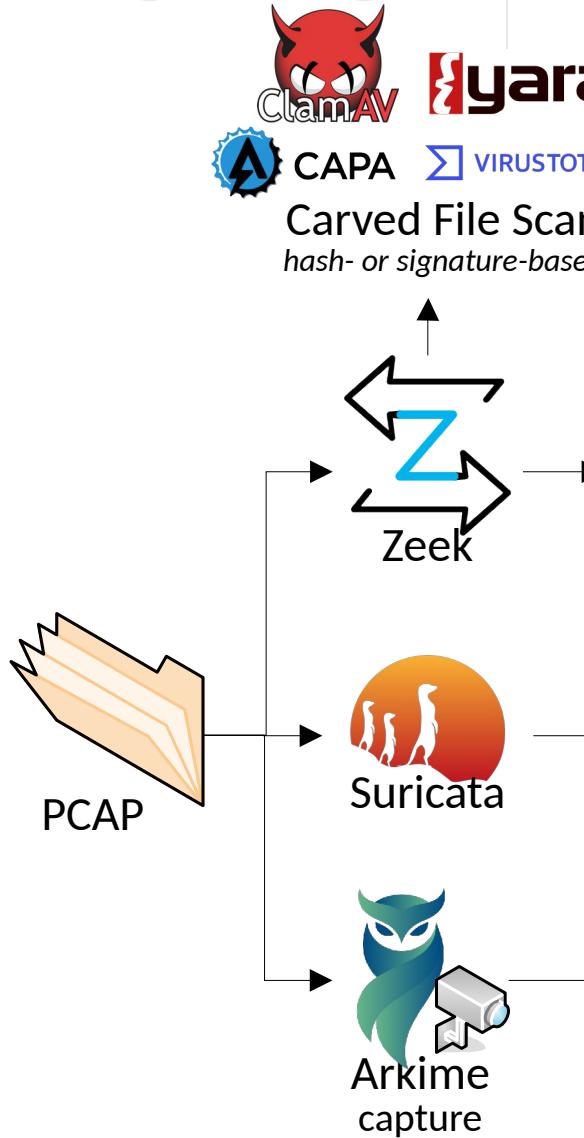
iFix_Server119.pcap
12 MB

MicroLogix56.pcap
9 MB

Modicon.pcap
883 KB

WinXP.pcap
3.4 MB

Malcolm



Data Pipeline

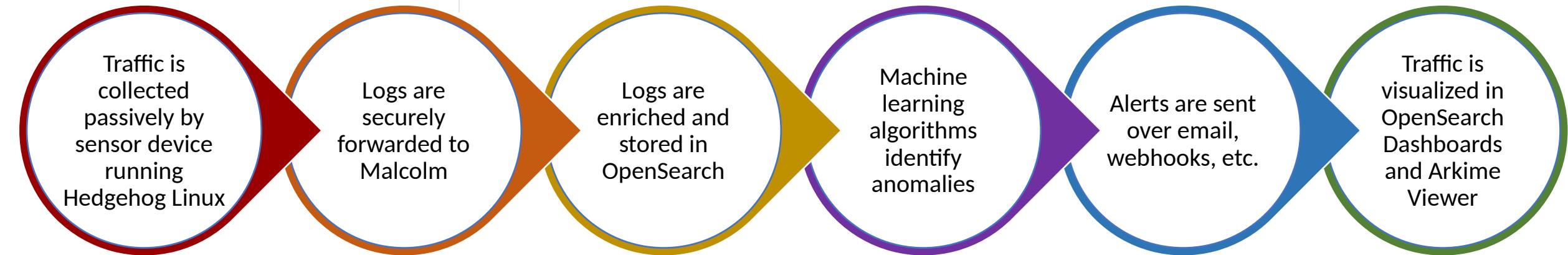
<https://idaholab.github.io/Malcolm/>

Malcolm



Data Pipeline

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



- Traffic is collected passively by sensor device running Hedgehog Linux
- Zeek, Arkime, and Suricata generate metadata about network communications
- Full PCAP is 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

- Communications between the sensor and aggregator are TLS-encrypted
- Sensor data including resource utilization, syslog, audit logs, temperatures, and more may also be forwarded
- Other third-party logs (e.g., Windows event logs, server host logs, etc.) may be shipped using Fluent Bit or Beats

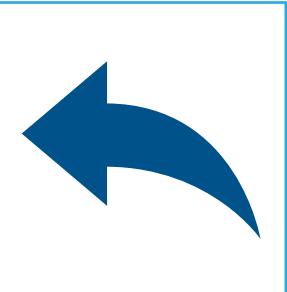
- Lookups are performed for GeoIP, ASN, MAC-to-vendor, community ID, domain name entropy, etc.
- Network events are normalized across protocols and data sources
- Best-guess techniques are applied to identify obscure OT traffic
- Enriched metadata may be forwarded to higher-tiered Malcolm instance

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

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

- Dozens of custom dashboards are provided for all supported protocols
- PCAP payloads are retrieved from sensor on demand
- Create custom visualizations via drag-and-drop interface
- Malcolm authenticates users from its own list, Active Directory / LDAP, or Keycloak

Log Enrichment



Reverse DNS

Domain Name
Entropy
Calculation

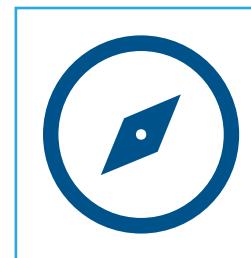
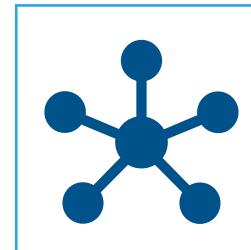


Severity Scoring



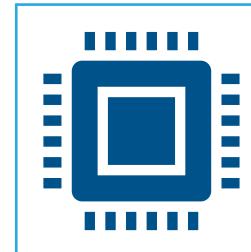
Tagging

GeoIP and ASN
Lookups



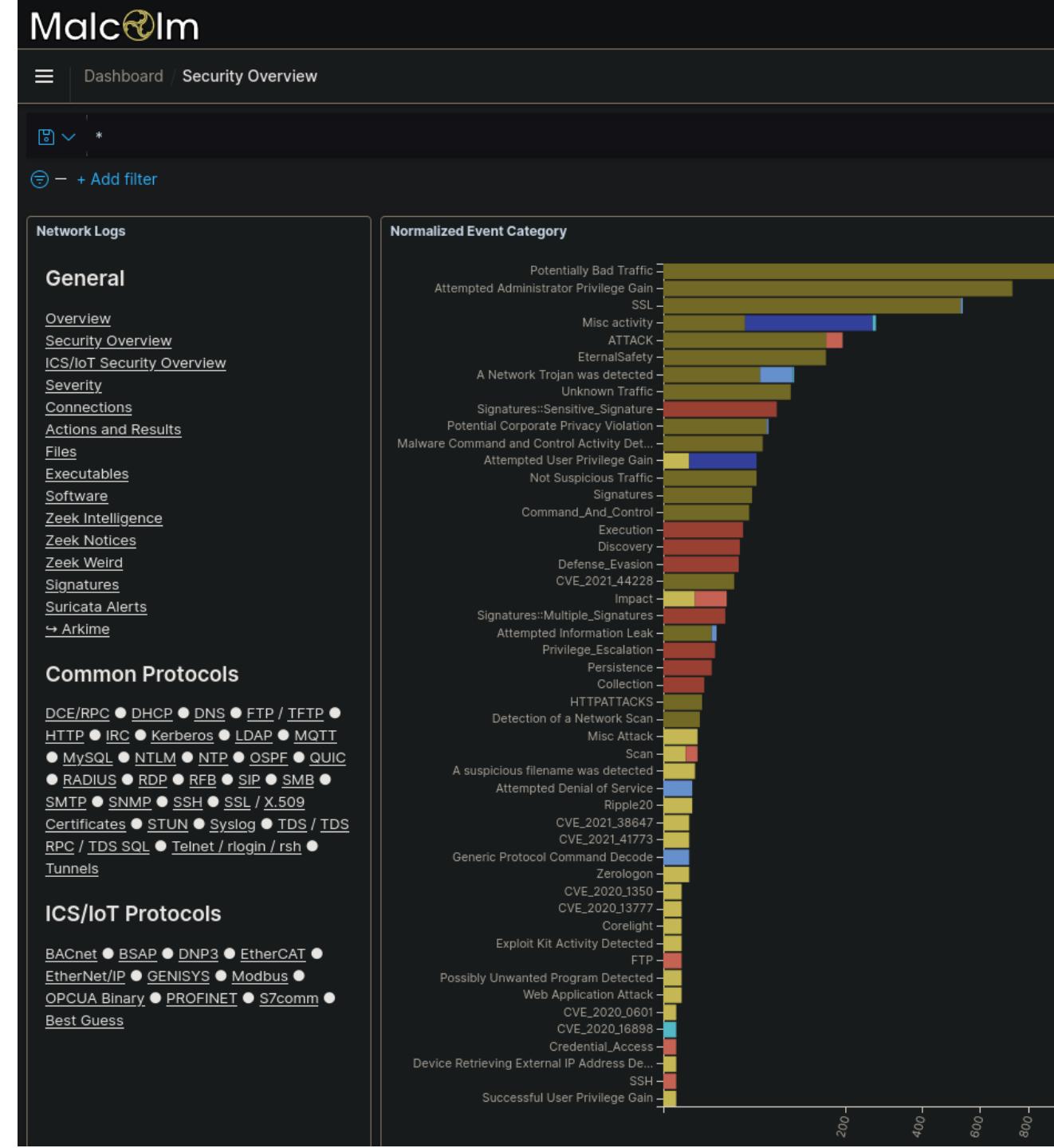
Network
Direction

Hardware Vendor
OUI Lookups



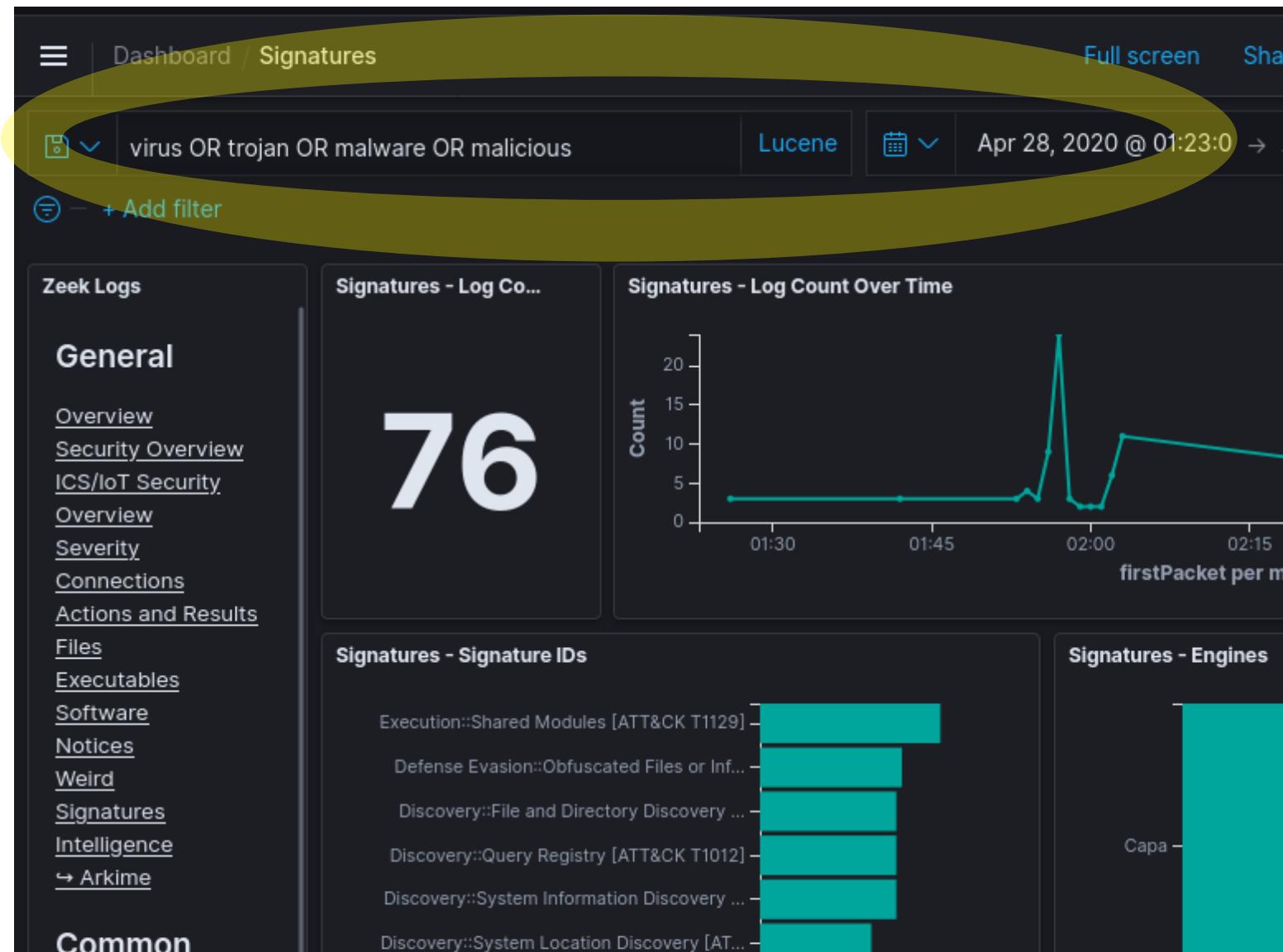
OpenSearch Dashboards

- Front end for Zeek logs and Suricata alerts
- Prebuilt visualizations for all supported protocols
- WYSIWYG editors to create custom visualizations and dashboards
- Drill down from high-level trends to specific items of interest



Dashboards Filters and Search

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



Overview Dashboards

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

Network Logs

General

[Overview](#)

[Security Overview](#)

[ICS/IoT Security Overview](#)

[Severity](#)

[Connections](#)

[Actions and Results](#)

[Files](#)

[Executables](#)

[Software](#)

[Zeek Intelligence](#)

[Zeek Notices](#)

[Zeek Weird](#)

[Signatures](#)

[Suricata Alerts](#)

[↳ Arkime](#)

Common Protocols

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

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

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

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

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

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

Normalized Event Categories

Po

Attempted Adminis

A Network T

Signatures::

Potential Corpora

Malware Command and C

Attempted

No

Com

Signatures::

Attempted

l

Detection

A suspicious file

Attempted

Zeek Notices

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

Malcolm

Dashboard / Zeek Notices

+ Add filter

Network Logs

General

[Overview](#)

[Security Overview](#)

[ICS/IoT Security Overview](#)

[Severity](#)

[Connections](#)

[Actions and Results](#)

[Files](#)

[Executables](#)

[Software](#)

[Zeek Intelligence](#)

[Zeek Notices](#)

[Zeek Weird](#)

[Signatures](#)

[Suricata Alerts](#)

↪ Arkime

Common Protocols

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

ICS/IoT Protocols

PGM ● PSAP ● PNPS ● ENIGMAT ●

Notices - Log Count

749

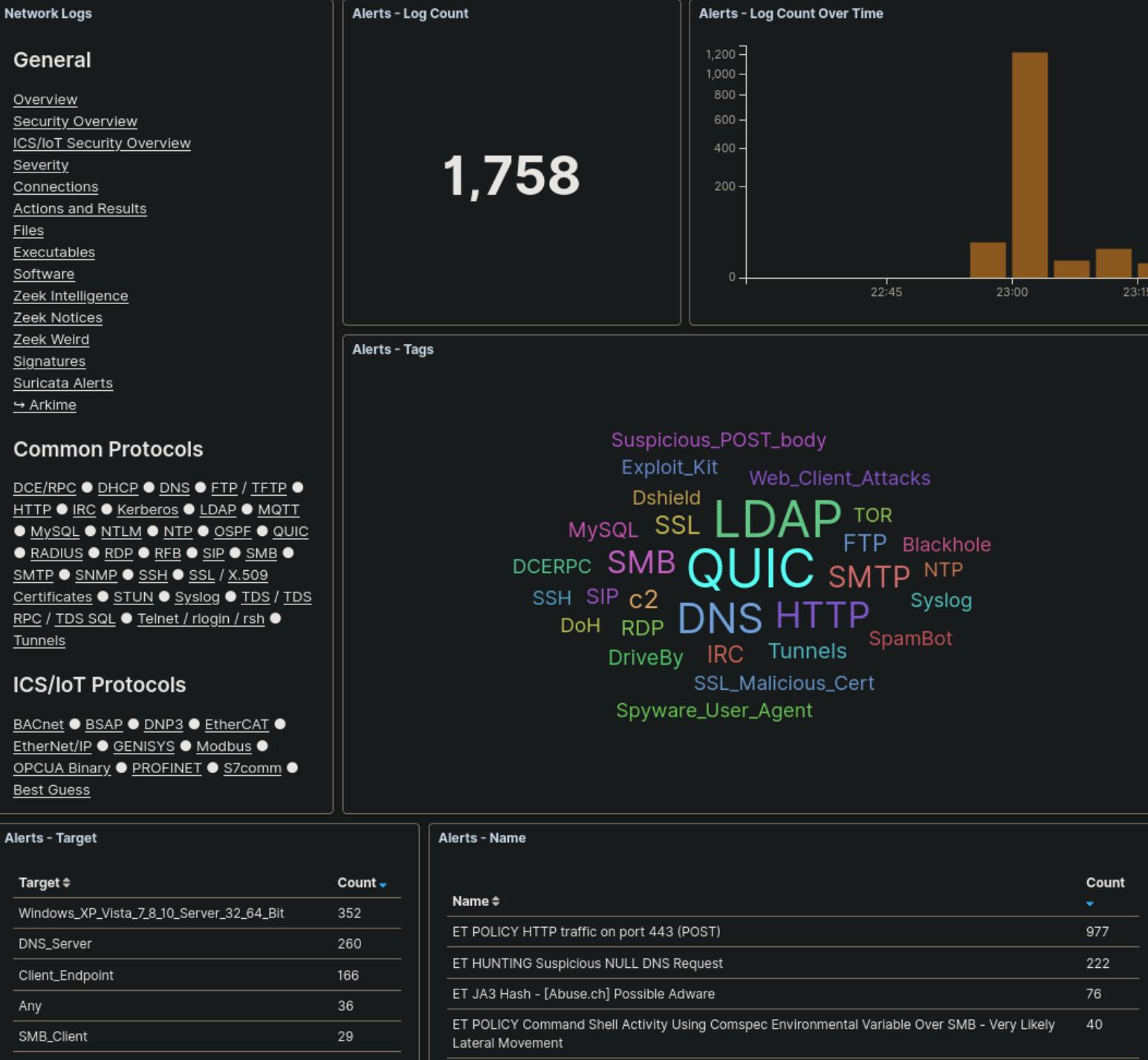
Notices - Log Count Over Time

Notices - Notice Type

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

Suricata Alerts

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



Security & ICS/IoT Security Overviews

Network Logs

General

- Overview
- Security Overview
- ICS/IoT Security Overview
- Severity
- Connections
- Actions and Results
- Files
- Executables
- Software
- Zeek Intelligence
- Zeek Notices
- Zeek Weird
- Signatures
- Suricata Alerts
- Arkime

Common Protocols

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

ICS/IoT Protocols

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

Outdated/Insecure Application Protocols

Application Protocol	Protocol Version	Count
smb	1	124,835
ftp	-	3,099
tls	TLSV10	422
tls	TLSV11	253
tls	-	239
ntp	3	90
ftp	-	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

Network Layer

Malcolm

Dashboard | ICS/IoT Security Overview

Full screen Share Clone Reporting

Normalized Event Category

Notice, Alert, Signature and Weird - Summary

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

Zeek Logs

ICS/IoT Log Counts

ICS/IoT Traffic Over Time

ICS/IoT External Traffic

General

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

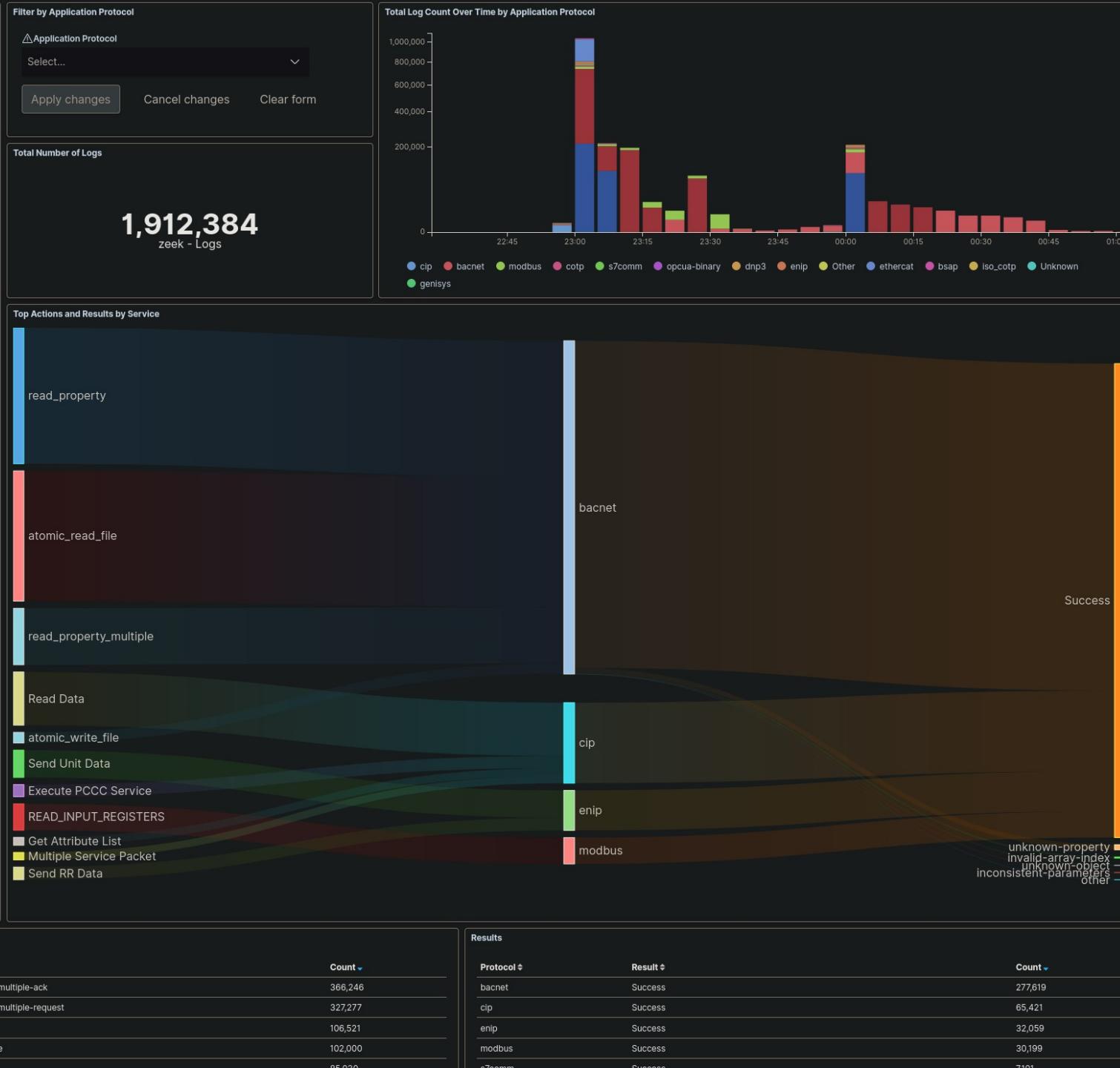
Common Protocols

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

ICS/IoT Protocols

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

Network Layer



Actions and Results

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

Protocol Dashboards

- Highlight application-specific fields of interest
- Grouped by common IT protocols and ICS/IoT protocols
- ICS protocols
 - BACnet
 - BSAP
 - DNP3
 - EtherCAT
 - EtherNet/IP
 - GENISYS
 - GE SRTP
 - HART-IP
 - Modbus
 - OPCUA Binary
 - PROFINET
 - S7comm
 - Synchrophasor (IEEE-C37.118)

[Zeek Intelligence](#)

[Zeek Notices](#)

[Zeek Weird](#)

[Signatures](#)

[Suricata Alerts](#)

[↳ Arkime](#)

Notices - Notice Type

Notice Category

SSL

ATTACK

ATTACK

EternalSafety

Signatures

EternalSafety

ATTACK

EternalSafety

EternalSafety

ATTACK

Common Protocols

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

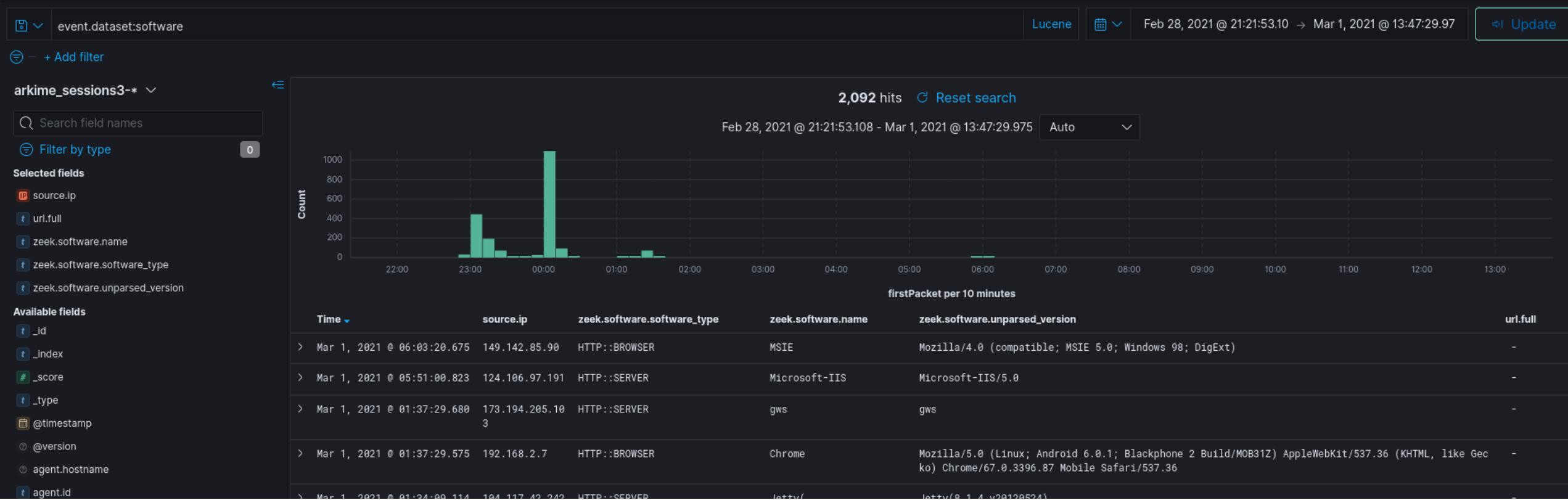
ICS/IoT Protocols

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

Export: Raw  For

Discover

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

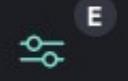


New Visualization

Filter



Area



Controls



Coordinate
Map



Data Table



Gantt Chart



Gauge



Goal



Heat Map



Horizontal Bar



Line



Markdown



Metric



Pie



Region Map



Sankey
Diagram



TSVB



Tag Cloud



Timeline



Vega



Vertical Bar

Custom Visualizations

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

Search Syntax Comparison

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

Search Syntax Comparison (cont.)

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

Search Syntax Comparison (cont.)

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

Search Syntax Comparison (cont.)

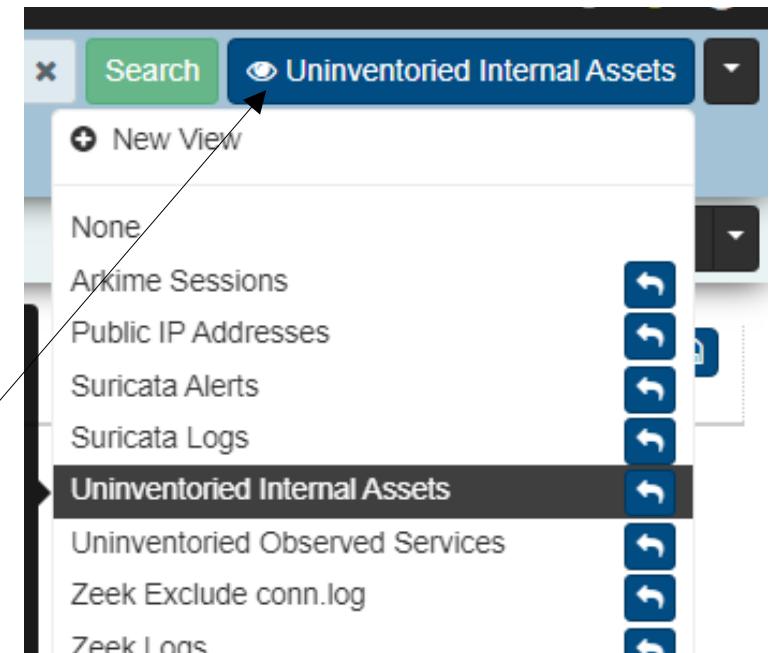
	Arkime	Dashboards (Lucene)	Dashboards (DQL)
GeolP information available	country == EXISTS!	_exists_:destination.geo OR _exists_:source.geo	destination.geo:* or source.geo:*
Log type	event.dataset == notice	event.dataset:notice	event.dataset:notice
IP CIDR Subnets	ip.src == 172.16.0.0/12	source.ip:"172.16.0.0/12"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



- Front end for enriched Zeek logs, Suricata alerts **and** Arkime sessions
 - Malcolm's custom Arkime data source adds full support for Zeek and Suricata 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

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



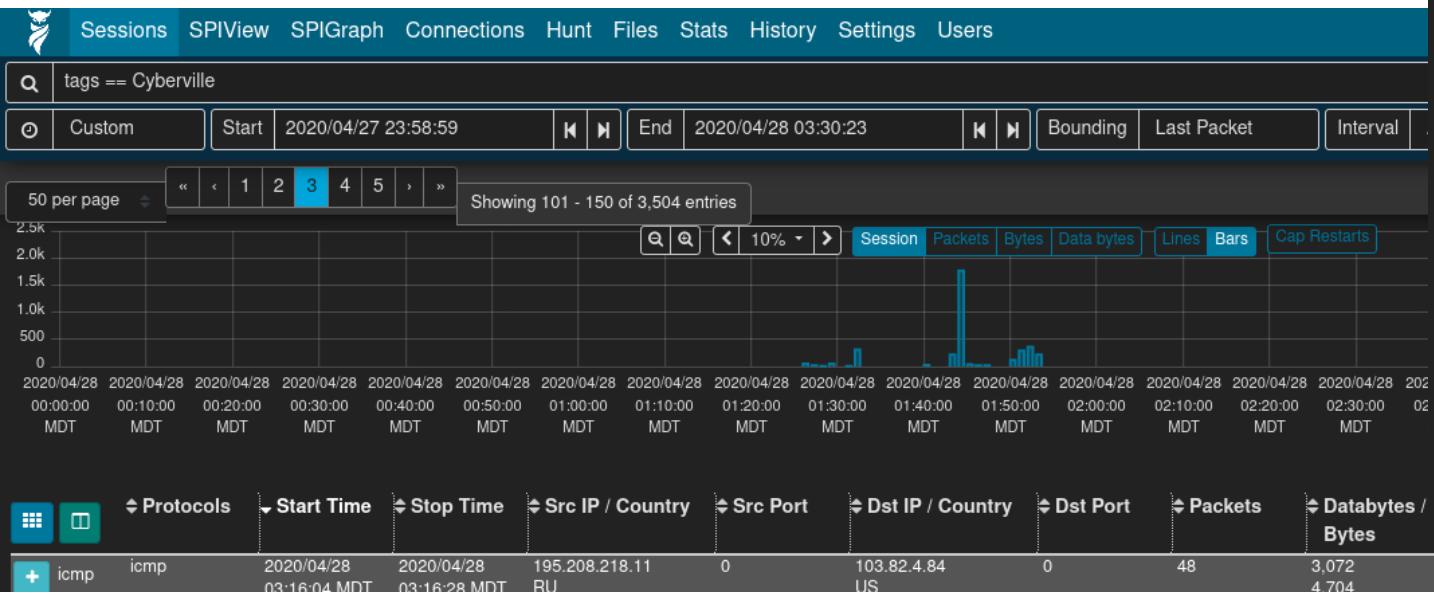
The screenshot shows the Arkime interface with the following details:

- Search Bar:** tags == Cyberville
- Time Filter:** Custom, Start: 2020/04/27 23:58:59, End: 2020/04/28 03:30:23, Interval: Auto, Duration: 03:31:24
- Map View:** Shows a world map with the location of the traffic highlighted.
- Session Timeline:** Shows network traffic over time, with a significant spike around 01:40 MDT on April 28, 2020.
- Table of Captured 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 page with a search bar containing "protocols == http && tags == external_destination". Below the search bar are buttons for Custom, Start (2020/11/11 06:23:48), End (2021/05/30 06:00:53), and Bound. A pagination bar shows "Showing 1 - 50 of 12,150 entries". The main content area lists log entries with dropdown menus for filtering. Key entries include:

- 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 view. At the top, there's a navigation bar with links for Sessions, SPIView, SPIGraph, Connections, Hunt, Files, Stats, History, Settings, and Users. Below the navigation bar is a search bar with the query "Pipeline Depth == 1 AND Request Method == GET AND URI == /_vti_bin/.../winnt/system32/cmd.exe?c+dir+x:\\". The main content area shows a single log entry with dropdown menus for filtering. The entry details:

- Pipeline Depth: 1
- Request Method: GET
- URI: /_vti_bin/.../winnt/system32/cmd.exe?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.
- Export Options:** Export PCAP.
- Bottom Filtering:** 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.
- Bottom Status:** URI: mirror.pacific.net.au/openoffice/stable/2.0.0/OOo_2.0.0_Win32Intel_install.exe
- Right Panel:** A world map showing traffic distribution.

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)



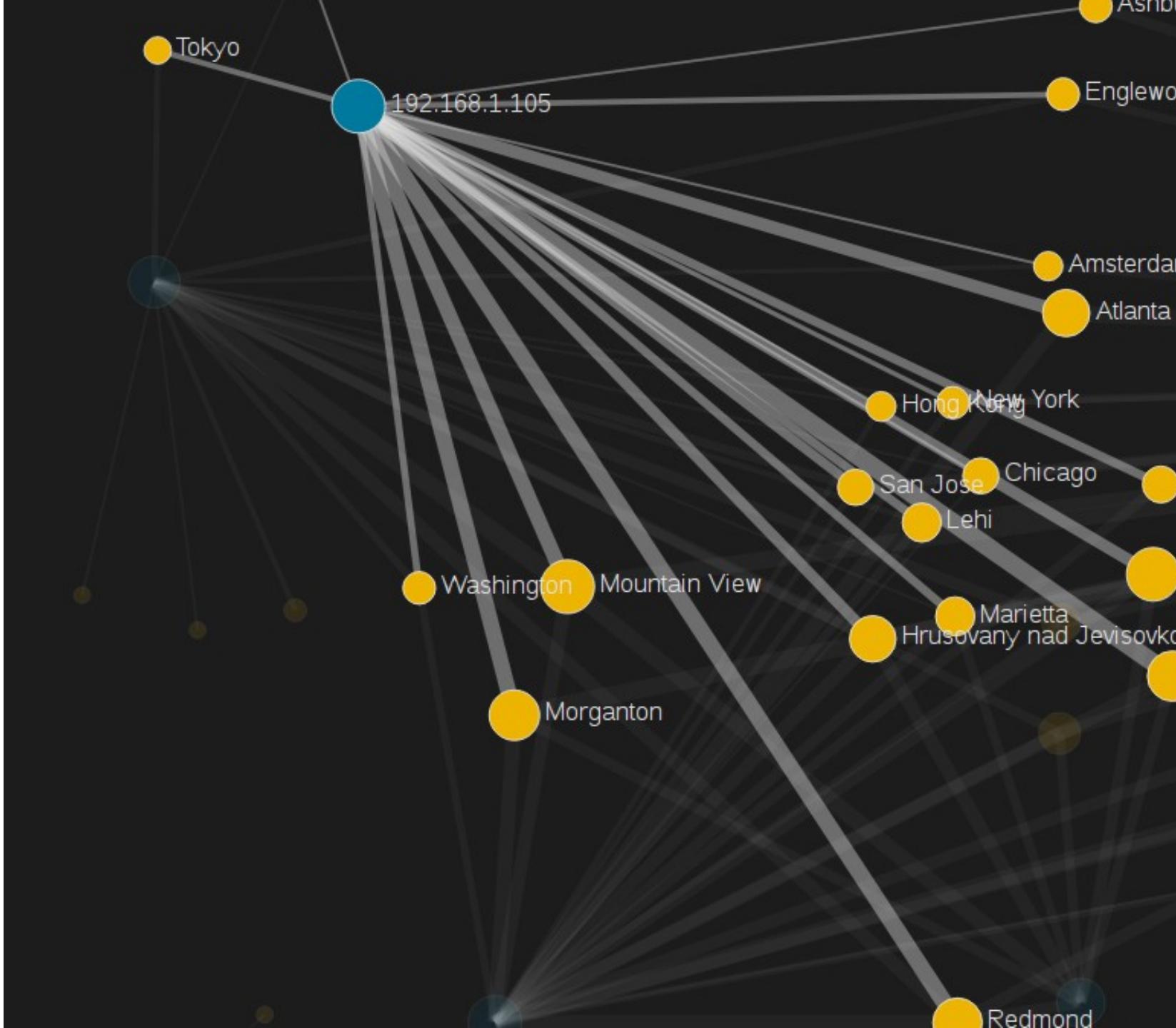
SPIGraph

- 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

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

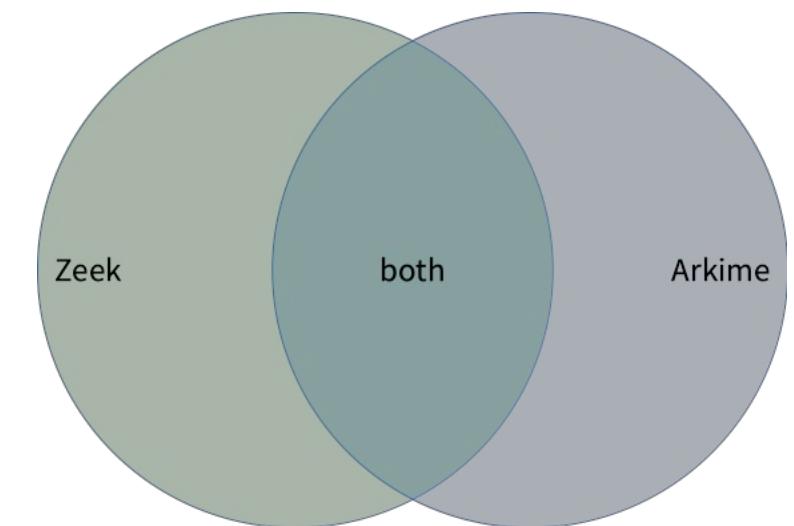
Search your packet search job history 50 per page 1 Showing 1 - 1 of 1 entries

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

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

Data Source Correlation

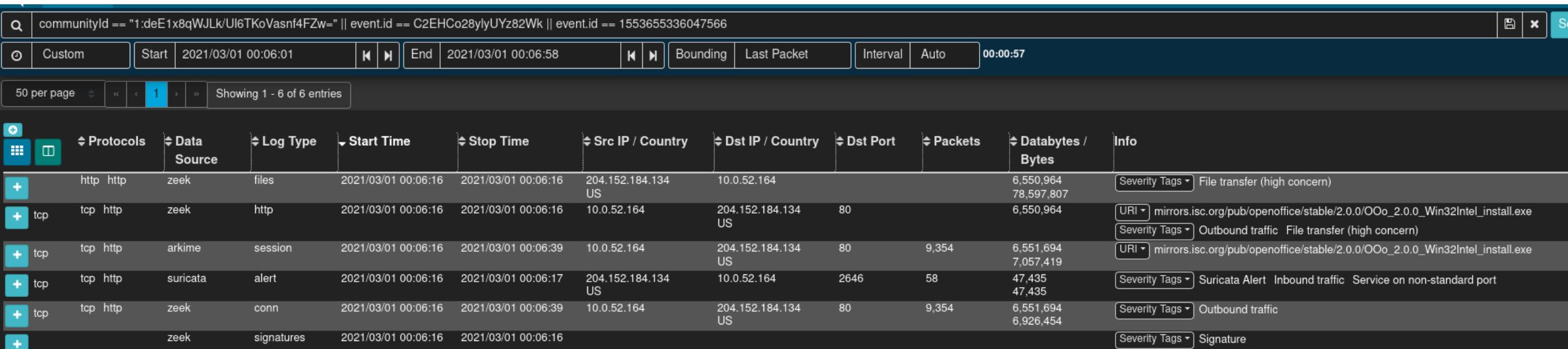
- Search syntax is different between Arkime and Dashboards (and in some cases, so are field names)
 - See search syntax comparison table, Malcolm and Arkime docs
- Despite considerable overlap, there are differences in protocol parser support among Zeek, Suricata and Arkime
 - Learning the strengths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



Correlate Zeek or Suricata Logs and Packet Payloads

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

```
communityId == "1:r7tGG//fXP1P0+BXH3zXETCtEFI=" || event.id == "CQcoro2z6adgtGlk42"
```



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

Protocols	Data Source	Log Type	Start Time	Stop Time	Src IP / Country	Dst IP / Country	Dst Port	Packets	Databytes / Bytes	Info
http http	zeek	files	2021/03/01 00:06:16	2021/03/01 00:06:16	204.152.184.134 US	10.0.52.164			6,550,964 78,597,807	Severity Tags ▾ File transfer (high concern)
tcp http	zeek	http	2021/03/01 00:06:16	2021/03/01 00:06:16	10.0.52.164	204.152.184.134 US	80		6,550,964	URI ▾ mirrors.isc.org/pub/openoffice/stable/2.0.0/OOo_2.0.0_Win32Intel_install.exe Severity Tags ▾ Outbound traffic File transfer (high concern)
tcp http	arkime	session	2021/03/01 00:06:16	2021/03/01 00:06:39	10.0.52.164	204.152.184.134 US	80	9,354	6,551,694 7,057,419	URI ▾ mirrors.isc.org/pub/openoffice/stable/2.0.0/OOo_2.0.0_Win32Intel_install.exe
tcp http	suricata	alert	2021/03/01 00:06:16	2021/03/01 00:06:17	204.152.184.134 US	10.0.52.164	2646	58	47,435 47,435	Severity Tags ▾ Suricata Alert Inbound traffic Service on non-standard port
tcp http	zeek	conn	2021/03/01 00:06:16	2021/03/01 00:06:39	10.0.52.164	204.152.184.134 US	80	9,354	6,551,694 6,926,454	Severity Tags ▾ Outbound traffic
	zeek	signatures	2021/03/01 00:06:16	2021/03/01 00:06:16						Severity Tags ▾ Signature

Verify Network Segmentation with NetBox and Asset Interaction Analysis

- Flat networks are problematic (and unfortunately common)
- Security-minded network design incorporates segregating (isolating) and segmenting (dividing) assets into zones with differing levels of trust and well-defined boundaries
 - Enterprise
 - DMZ
 - Operation and Control
 - Supervisory Control
 - Process
 - etc.

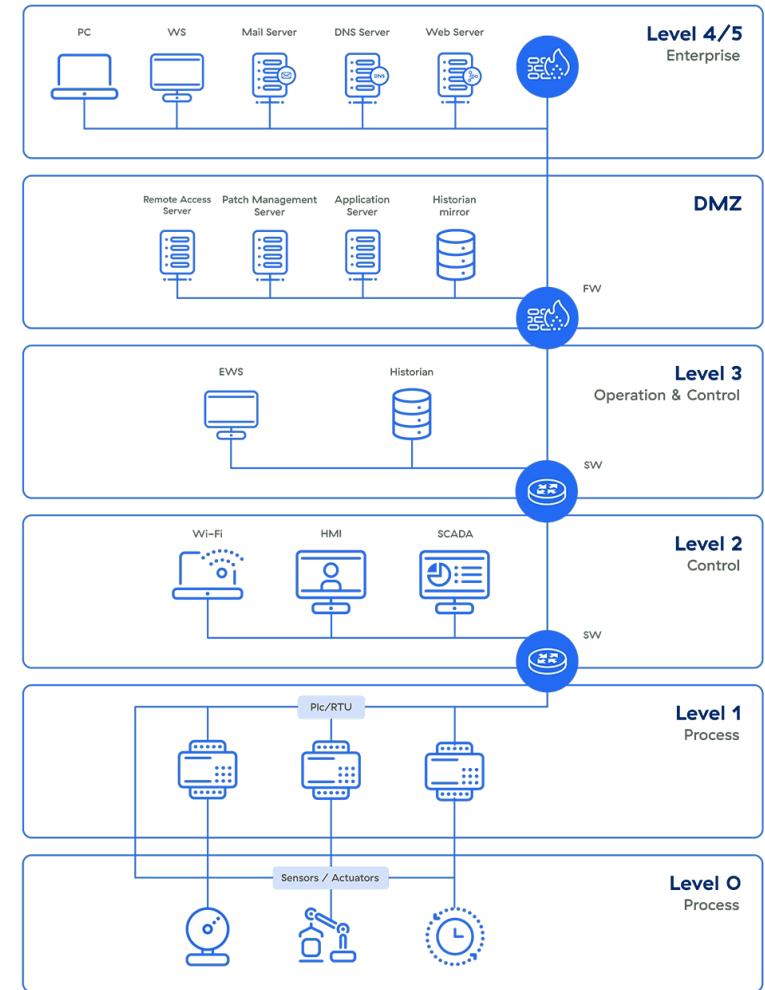
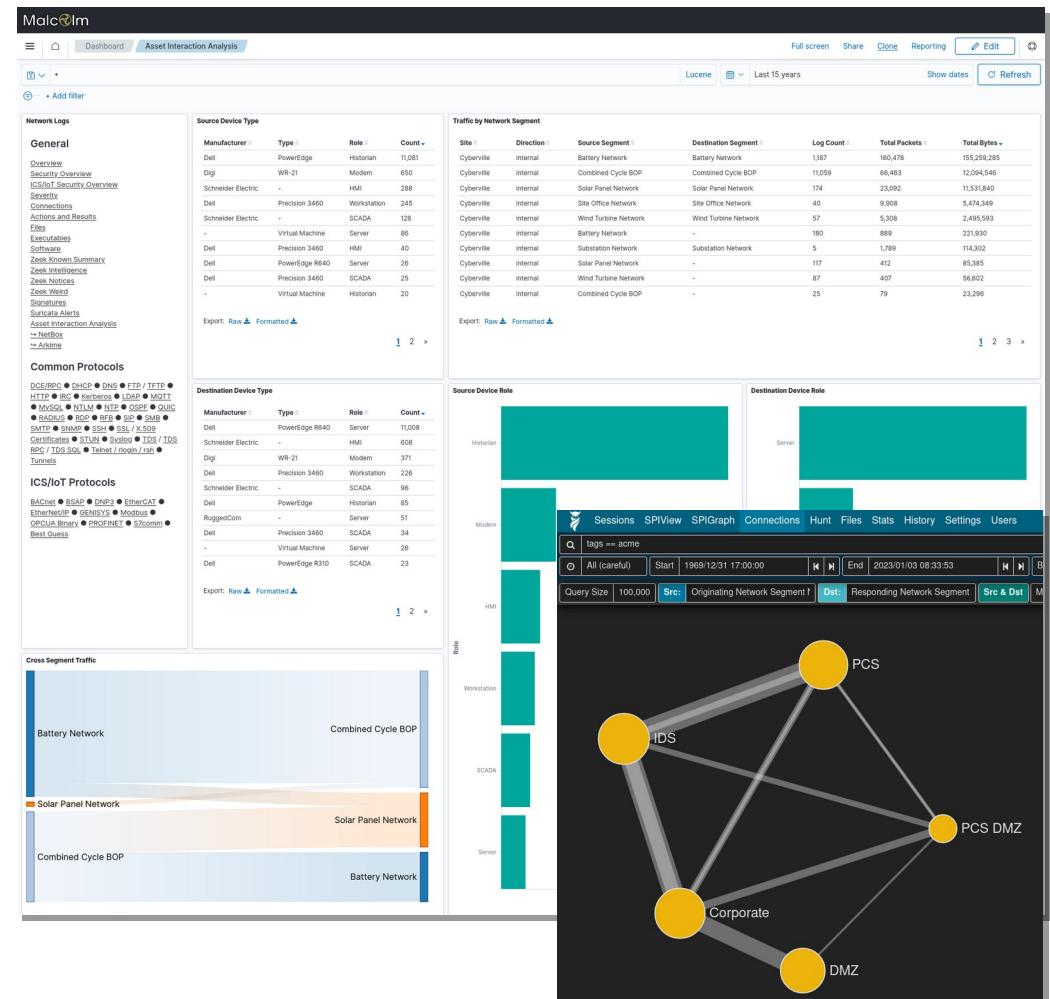


Image credit: Zscaler.com

Verify Network Segmentation with NetBox and Asset Interaction Analysis

- Network prefixes are identified by name and CIDR subnet in NetBox
- Observed network traffic is cross-referenced against inventory
- Malcolm dashboards such as **Asset Interaction Analysis** highlight relationships among network segments



Pinpoint Rogue Devices and Services

- Networked devices can be inventoried in NetBox
 - Name
 - IP address
 - Manufacturer
 - SKU or model number
 - Role
- Expected services (application protocols) can be specified for the devices that provide them

Devices

Results 52 Filters 1

× Site: Cyberville Save

Quick search

<input type="checkbox"/>	NAME	STATUS	SITE	ROLE	MANUFACTURER	TYPE	IP ADDRESS	TAGS
<input type="checkbox"/>	4yH2O0Z20tmgmH8v	Active	Cyberville	Unspecified	Microsoft Corporation	Unspecified	192.168.0.129/32	Autopopulated
<input type="checkbox"/>	Battery HMI	Active	Cyberville	HMI	Schneider Electric	Unspecified	10.10.10.3/32	—
<input type="checkbox"/>	Battery Historian	Active	Cyberville	Historian	Dell	PowerEdge	10.10.10.5/32	—
<input type="checkbox"/>	Cellular Modem	Active	Cyberville	Modem	Digi	WR-21	10.10.10.11/32	—
<input type="checkbox"/>	Combined Cycle BOP Historian	Active	Cyberville	Historian	Dell	PowerEdge	10.10.20.5/32	—
<input type="checkbox"/>	DROP200	Active	Cyberville	Server	Dell	PowerEdge R640	10.10.20.10/32	—
<input type="checkbox"/>	ENG_WORKSTATION	Active	Cyberville	Workstation	Dell	Precision 3460	10.10.20.8/32	—
<input type="checkbox"/>	Microsoft Corporation @ 10.0.0.130	Active	Cyberville	Unspecified	Microsoft Corporation	Unspecified	10.0.0.130/32	Autopopulated
<input type="checkbox"/>	Modbus Client 12	Active	Cyberville	SCADA	Schneider Electric	Unspecified	10.10.10.12/32	—
<input type="checkbox"/>	Modbus Client 55	Active	Cyberville	SCADA	Dell	Precision 3460	10.10.10.55/32	—
<input type="checkbox"/>	Modbus Client 64	Active	Cyberville	SCADA	Schneider Electric	Unspecified	10.10.10.64/32	—

Pinpoint Rogue Devices and Services

- Dashboards highlight “uninventoried” devices and services (observed but not accounted for in NetBox)
 - Zeek Known Summary
 - Asset Interaction Analysis
- Premade Arkime views
 - Uninventoried Internal Assets
 - Uninventoried Observed Services

Uninventoried Internal Source IPs			
IP Address	Site	Segment	Count
10.10.10.5	Cyberville	Battery Network	295
192.168.95.128	-	-	30
10.10.30.129	Cyberville	Wind Turbine Network	11
192.168.0.129	Cyberville	Solar Panel Network	10
10.10.100.50	Cyberville	Substation Network	10
192.168.95.1	-	-	9
10.0.0.133	Cyberville	Site Office Network	9
10.0.0.120	Cyberville	Site Office Network	8
10.0.0.40	Cyberville	Site Office Network	8
192.168.95.134	-	-	6

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

1 2 »

Uninventoried Internal Destination IPs			
IP Address	Site	Segment	Count
192.168.95.1	-	-	31
192.168.95.254	-	-	26
192.168.0.255	Cyberville	Solar Panel Network	24
10.10.10.255	Cyberville	Battery Network	18
192.168.0.129	Cyberville	Solar Panel Network	11
10.10.30.131	Cyberville	Wind Turbine Network	11
10.0.0.128	Cyberville	Site Office Network	9
10.0.0.40	Cyberville	Site Office Network	8
10.0.0.120	Cyberville	Site Office Network	6
10.10.255.255	-	-	5

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

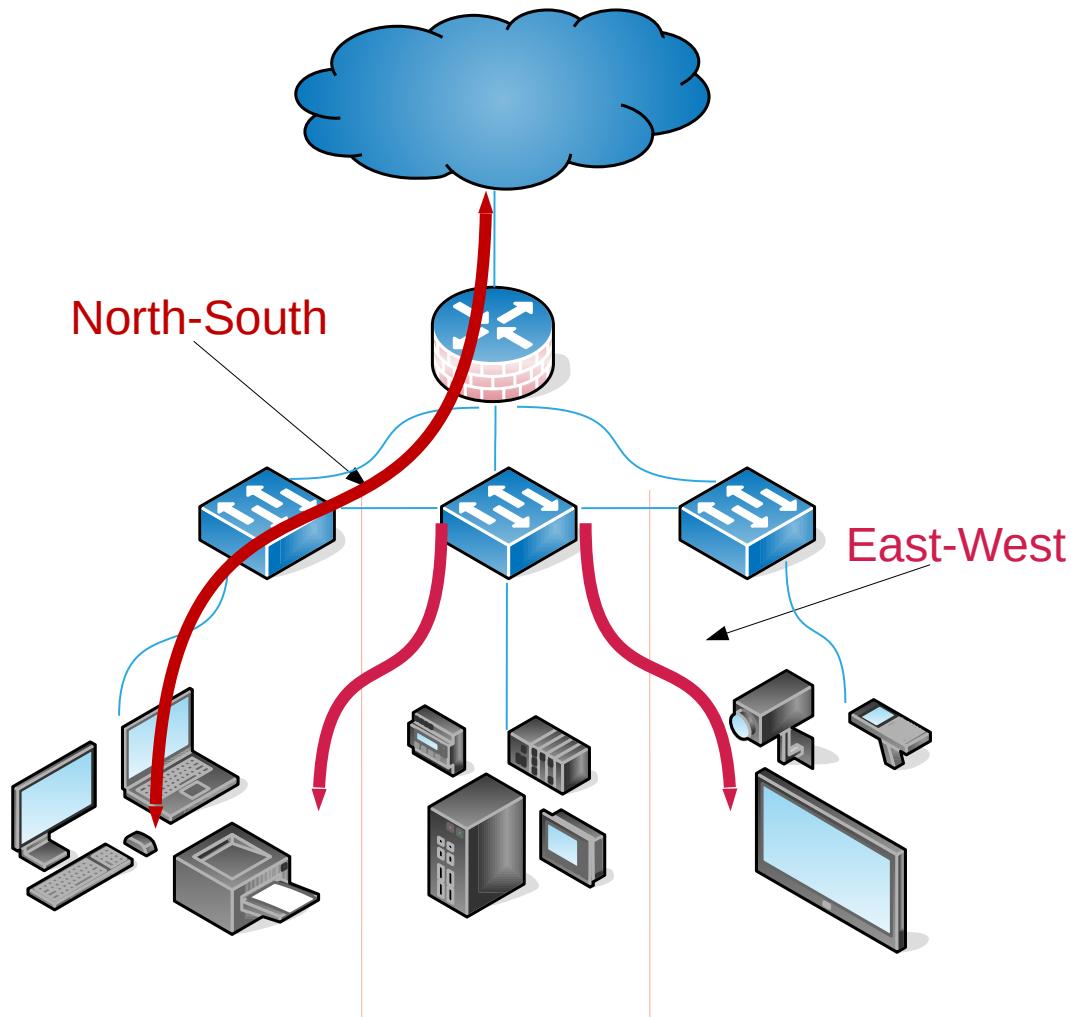
1 2 »

Uninventoried Internal Assets - Logs											
Time	network.transport	network.protocol	event.provider	event.dataset	source.segment.name	source.oui	source.ip	destination.segment.name	destination.oui	destination.ip	event.id
> Apr 28, 2020 @ 03:20:09.973	tcp	smb	suricata	alert	Battery Network	-	10.10.10.5	Battery Network	-	10.10.10.10	83713027856898
> Apr 28, 2020 @ 03:20:09.973	tcp	smb	suricata	alert	Battery Network	-	10.10.10.5	Battery Network	-	10.10.10.10	83713027856898
> Apr 28, 2020 @ 03:20:09.973	tcp	-	zeek	conn	Battery Network	VMware, Inc.	10.10.10.5	Battery Network	RuadedCom Inc.	10.10.10.10	CcfCMn2Jff01JL

1–50 of 1175 < >

The screenshot shows the Arkime interface with the title bar "Uninventoried Internal Assets". Below the title, there's a "New View" button and a list of available views: "None", "Arkime Sessions", "Public IP Addresses", "Suricata Alerts", "Suricata Logs", "Uninventoried Internal Assets" (which is currently selected and highlighted in dark blue), "Uninventoried Observed Services", "Zeek Exclude conn.log", and "Zeek Logs". To the right of the list, there are several small blue circular icons with arrows, likely for navigating between tabs or switching between different parts of the interface.

Survey East-West and North-South Traffic



network.direction field

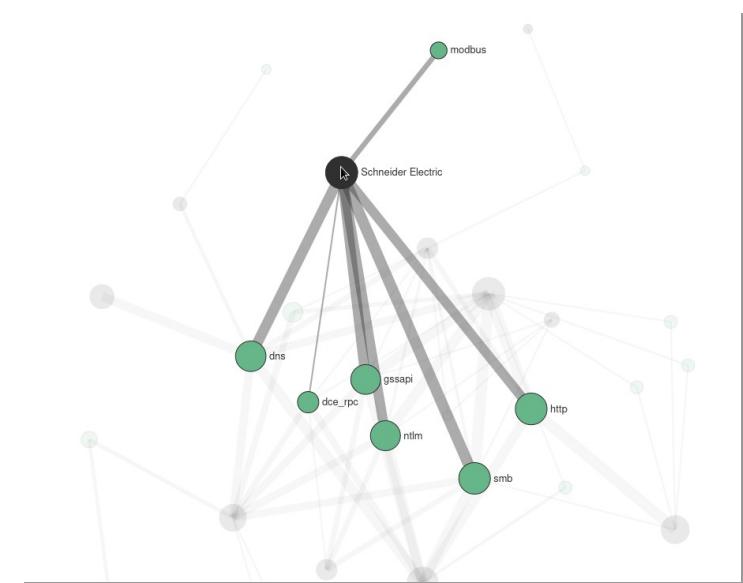
- internal: East-West
- inbound: “Southbound”
- outbound: “Northbound”

Review Observed Network Protocols

- Unsecure or outdated protocols
 - SSH < v2
 - TLS < v1.2
 - NTP < v4
 - VNC < v3.8
 - RDP < v6
 - LDAP < v3
 - FTP/TFTP
 - SMB < v3
 - telnet/rlogin/rsh
 - etc.
- Unexpected protocols (traffic that should not be present in a network segment)
 - IPv6
 - DNS
 - DHCP
 - Remote desktop
 - Automatic software updates
 - etc.

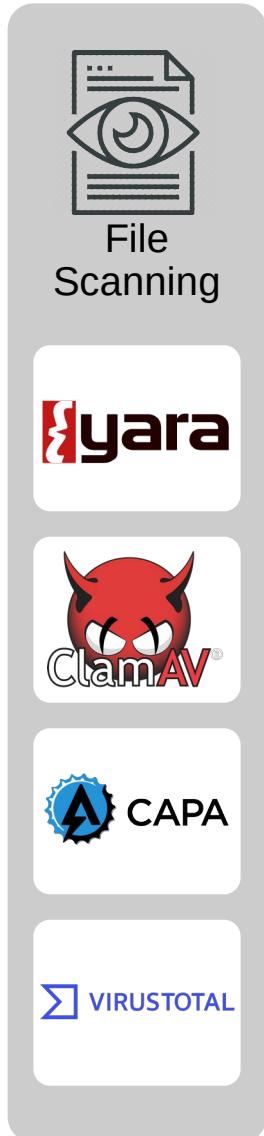
Outdated/Insecure Application Protocols		
Application Protocol	Protocol Version	Count
smb	1	124,835
ftp	-	3,099
tls	TLSv10	422
tls	TLSv11	253
tls	-	239
ntp	3	90
tftp	-	84
snmp	1	59
snmp	2c	31
telnet	-	10

Vulnerabilities	
Data Source	Log Type
zeek	notice
zeek	notice
suricata	alert
suricata	alert
zeek	notice
suricata	alert



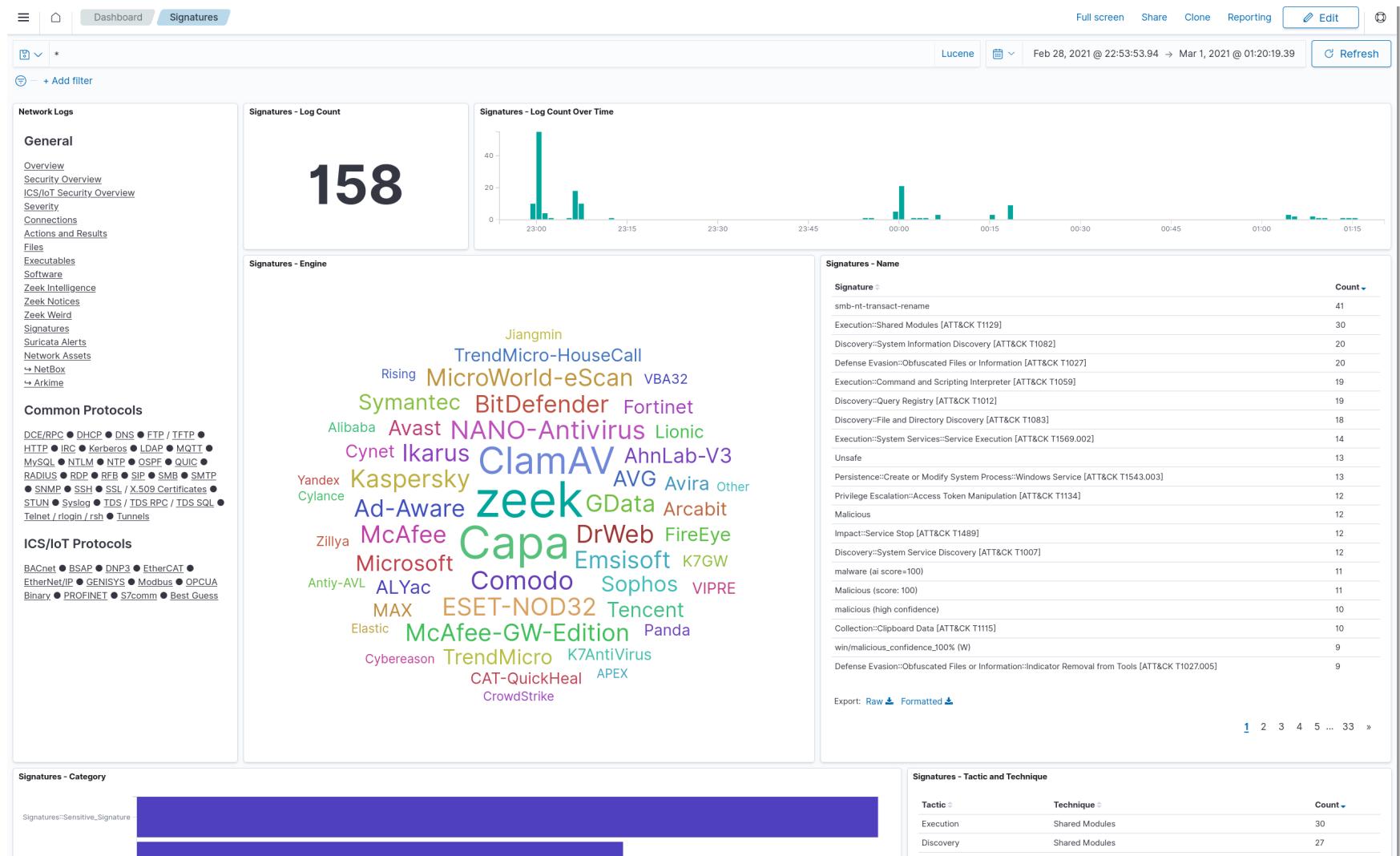
Investigate Suspicious File Transfers

- File transfers are detected in network traffic
 - SMB/CIFS (Windows file shares)
 - SMTP (Email)
 - HTTP (Web downloads)
 - etc.
- Files automatically extracted and scanned by a variety of tools
 - YARA - general-purpose scanner to identify and classify malware
 - ClamAV - antimalware toolkit able to detect many types of malware, including viruses
 - capa - executable capabilities analyzer
 - VirusTotal - an online service that can identify malicious files based on file hashes submitted to it
 - Assemblyline – a file triage and malware analysis system “integrating the cyber security community’s best tools”
 - Malcolm integration coming in 2025



File Transfers: Signatures Dashboard

Hits from file scanning engines against files carved from network traffic



File Transfers: Preserved Files

Files identified as potentially malicious are flagged and optionally preserved for further investigation

HTTP-F0joG01ZpxOl9fc...	text/plain	9.1KiB	HTTP	Cbr24h1HZxRCaZTFkcF0joG01ZpxOl9fcSN8	2024-08-01 17:11:09	
HTTP-F0xURA1fatUCF6x...	application/octet-stream	200.0B	HTTP	CP83GU157AS9a3MMG5 F0xURA1fatUCF6xke6	2024-08-01 17:06:49	
HTTP-F1AUDG3YVEB45Pl...	application/x-executable	2.5MiB	HTTP	CKI3cA387MUBTT3hG9 F1AUDG3YVEB45Plu3	2024-08-01 17:19:48	
HTTP-F1CbYm1PWuPjHV3...	text/html	6.9KiB	HTTP	CSxEQV3MW30sI3U5sh F1CbYm1PWuPjHV3hEc	2024-08-01 17:09:40	
HTTP-F1GA1N2fe1qQo4p...	application/octet-stream	160.0B	HTTP	C8gQXv40SN245pNi4d F1GA1N2fe1qQo4pd93	2024-08-01 17:38:03	

200 packets natural ▲ Packet Options ▾ Src Dst UnXOR Brute GZip Header UnXOR Unbase64

Source (10.10.10.5:3453)

GET /__utm.gif HTTP/1.0
Accept: */*
Cookie: tJDITvSBUfmkbv6PCGOUWJ8BLelApsLN4/NnjAtbb1cCAso8DUXqXgxUEjENFDZkSYkWM/Udc9Hk9Z1+3heBnnGwJftTDeuEgdLkts52v/XIKITHiEXVvakotKJYevVY4tWeQjH5waD8mFQYS1YeZ2v7Ws7DdHyOCzKooJaEwg8=

Host: {F14G-c2-host-header}

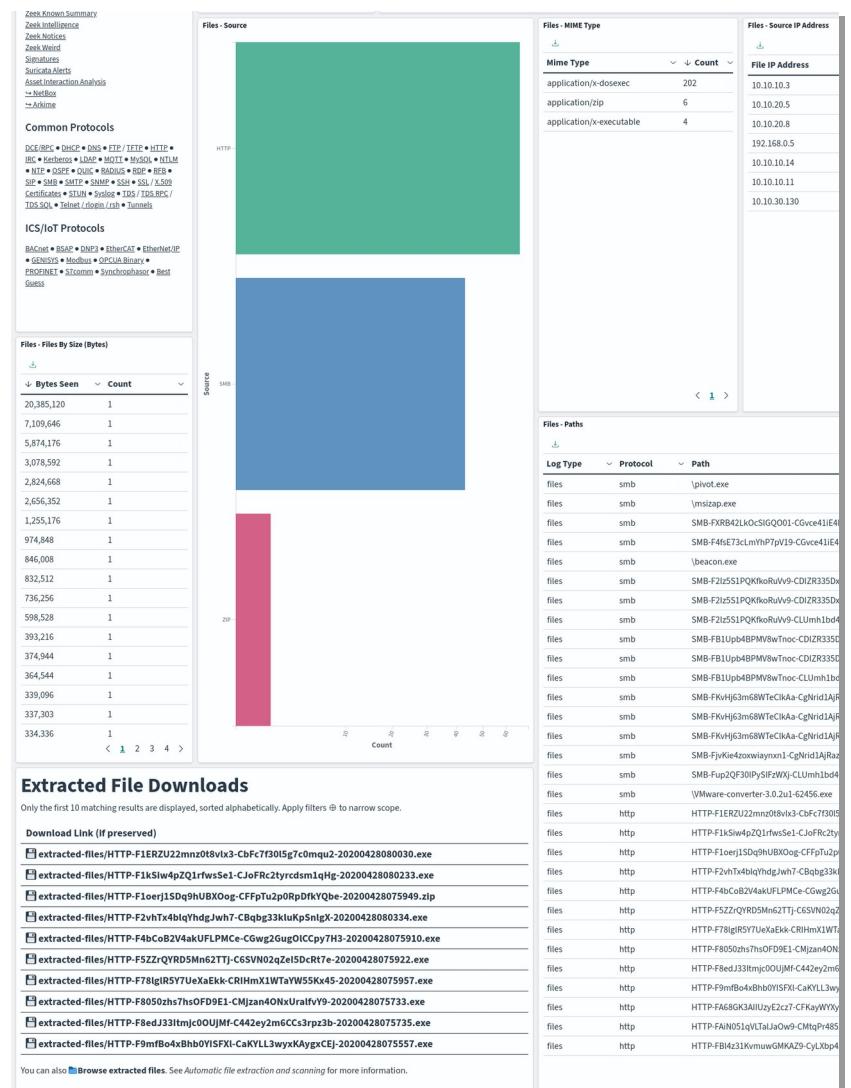
User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 5.1)

Connection: Keep-Alive

Pragma: no-cache

Destination (10.10.10.11:80)

HTTP/1.1 200 OK
Date: Mon, 20 Apr 2020 20:22:44 GMT
Content-Type: application/octet-stream
Content-Length: 160



Examine Sensitive Unencrypted Data

- Observed cleartext credentials are normalized (e.g., **related.user** and **related.password**) for ease of locating them across protocols
- Arkime Hunt feature can deep-packet search session payloads for strings, hex codes, or regular expression matches

The screenshot shows the Arkime application interface. At the top, there's a navigation bar with links: Sessions, SPIView, SPIGraph, Connections, Hunt, Files, Stats, History, Settings. Below the navigation bar is a search bar with the query "protocols == http". Underneath the search bar are buttons for "All (careful)", "Start" (set to 1969/12/31 17:00:00), "End" (set to 2023/01/12 10:24:28), "Bounding", and "Last Packet". A message says "Creating a new packet search job will search the packets of 6,511 sessions." There's a green button labeled "Create a packet search job". On the right side of the interface, there are page navigation buttons (1, 2, 3, »).

The main area displays two windows:

- Hunt Job Queue:** A table showing a single running hunt job.

Status	Matches	Name	User	Search text	Notify	Created	ID
Running	88.7%	137	HTTP with password	tlacuache	password (ascii)	2023/01/12 10:24:55	ezADp4UBDnyT0P6PZAK4

A progress bar indicates the search is at 88.7% completion. Below the table, there are status messages: "This hunt is running", "No description", "Found 137 sessions matching password (ascii) of 5,773 sessions searched", and "Still need to search 738 of 6511 total sessions".
- Clear-text Transmission of Passwords:** A table listing various protocols, usernames, and their counts.

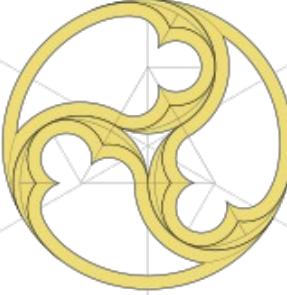
Application Protocol	Username	Count
ftp	anonymous	1,414
http	Login	102
http	maint	88
ftp	ind@psg420.com	20
ftp	DWSDataXfer	8
ftp	salesxfer	5
ldap	nginx_bind_dn@localdomain.lan	4
http	Unknown	4
ftp	sean@infosecs.cf	3
telnet	fake	2

Clear-text Transmission of Passwords		
Application Protocol	Username	Count
ftp	anonymous	1,414
http	Login	102
http	maint	88
ftp	ind@psg420.com	20
ftp	DWSDataXfer	8
ftp	salesxfer	5
ldap	nginx_bind_dn@localdomain.lan	4
http	Unknown	4
ftp	sean@infosecs.cf	3
telnet	fake	2

Towards the Future

- 
- Expand training via prepackaged modules, videos, and hands-on exercises
 - Vulnerability/IOC sharing and identification (CSAF) and exploitation visibility (KEV)
 - Fine-grained access control with user roles
 - Support generic (Sigma) rules and analytics
 - Distributed rule/policy management for network sensors
 - Improve cloud deployment
 - Evaluate/integrate NLP/LLM plugins in development at PNNL and explore other ML/AI integrations
 - Integration of a more powerful and flexible malware analysis platform (Strelka)
 - Improve integration of third-party and host logs
 - Increase OT/ICS protocol support
 - IEC 104, ANSI C12.22, Omron FINS, and more

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 © 2025 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.