Bangladesh University of Engineering and Technology

ZEEK

CSE 406 - Computer Security Sessional

Team -

1905075 Nahida Marzan
1905086 Sushmita Paul

Supervised by - Dr. Md. Shohrab Hossain

Contents

1 Introduction

1.1 Overview

Zeek is a passive, open-source network traffic analyzer extensively utilized in the realm of network security monitoring (NSM) and a plethora of other traffic analysis tasks. Originally developed for security investigations, Zeek's capabilities transcend to performance measurement and network troubleshooting, offering users a comprehensive view of their network activity through detailed logs and analysis features.

1.1.1 Comprehensive Logging

One of the primary advantages of using Zeek is its ability to produce extensive logs that capture a detailed record of network activities. These logs include:

- Application-layer transcripts such as HTTP sessions, with details like URIs, headers, MIME types, and server responses.
- DNS requests and responses.
- SSL certificate information.
- Key content from SMTP sessions.

Zeek formats these logs in structured, tab-separated, or JSON files, making them ideal for post-processing with external software or integration with databases and SIEM products for further analysis.

1.1.2 Built-in Analysis and Detection Capabilities

Zeek is not just about logging; it offers built-in functions for various analysis and detection tasks, such as:

- Extracting files from HTTP sessions.
- Detecting malware by interfacing with external registries.
- Identifying vulnerable software versions and popular web applications on the network.
- Detecting SSH brute-force attacks and validating SSL certificate chains.

These functionalities are readily available "out of the box," enhancing Zeek's utility as a network analysis tool.

1.1.3 Customization and Extensibility

Zeek stands out for its customizability and extensibility, thanks to its domain-specific scripting language. This language allows users to:

- Express arbitrary analysis tasks.

- Write custom code for unique analysis requirements, extending the pre-built "standard library" that comes with Zeek.

This feature positions Zeek as a versatile tool that can adapt to various network monitoring and analysis needs beyond its default offerings.

1.1.4 High-Performance and Scalability

Designed to run on commodity hardware, Zeek presents a cost-effective alternative to proprietary network monitoring solutions. Its performance and scalability make it suitable for:

- High-speed, high-volume network monitoring, including 10GE and 100GE networks.
- Deployment in large-scale environments using Zeek Clusters for load-balancing and centralized management.

Zeek's scalability is one of its core strengths, allowing it to accommodate growing network demands without compromising on performance.

1.1.5 Comparison with Other Tools

While Zeek excels in generating high-fidelity network logs and analysis, it is distinct from:

- Signature-based IDS systems like Suricata, as Zeek adopts a broader approach to detecting malicious activity through scripting.
- Protocol analyzers like Wireshark, which focus on detailed frame-level traffic representation, unlike Zeek's focus on compact, high-fidelity logs.

Zeek offers a balanced and efficient solution for network traffic analysis, combining comprehensive logging, built-in analysis capabilities, and extensive customization options. Its adaptability to both small-scale and large-scale environments, along with its open-source nature, makes Zeek a valuable tool for a wide range of users looking to gain deeper insights into their network activity.

1.2 History of Zeek

Zeek, initially known as Bro, was developed by Vern Paxson in 1995 at the Lawrence Berkeley National Laboratory (LBNL). Designed as a network monitoring tool, it was coined ”Bro” to serve as an Orwellian reminder of the surveillance capabilities inherent in network monitoring, with a nod to the potential for privacy violations.

Deployed at LBNL in 1996, Zeek's foundational paper won the Best Paper Award at the USENIX Security Symposium in 1998, reflecting its early recognition and significance in the field of network security.

With support from the National Science Foundation (NSF) and the Department of Energy (DOE), the project flourished through a blend of academic research and practical application. As Zeek's user base grew, the need for a more user-friendly interface became apparent, leading to a substantial update with the 2.0 release in 2012, in collaboration with the National Center for Supercomputing Applications (NCSA).

Subsequent years saw significant enhancements to Zeek, including native IPv6 support, the creation of the Bro Center of Expertise, the launch of try.zeek.org, and the introduction of the Broker communication framework and Zeek package manager.

In 2018, to better reflect its community values and avoid the negative connotations of ”bro culture,” the project was renamed from Bro to Zeek. Version 3.0, released in 2019, was the first to bear the new name. The year 2020 marked a period of renewed focus on community engagement, with increased outreach efforts aimed at expanding the Zeek community.

1.3 Optimal Locations for Zeek Deployment

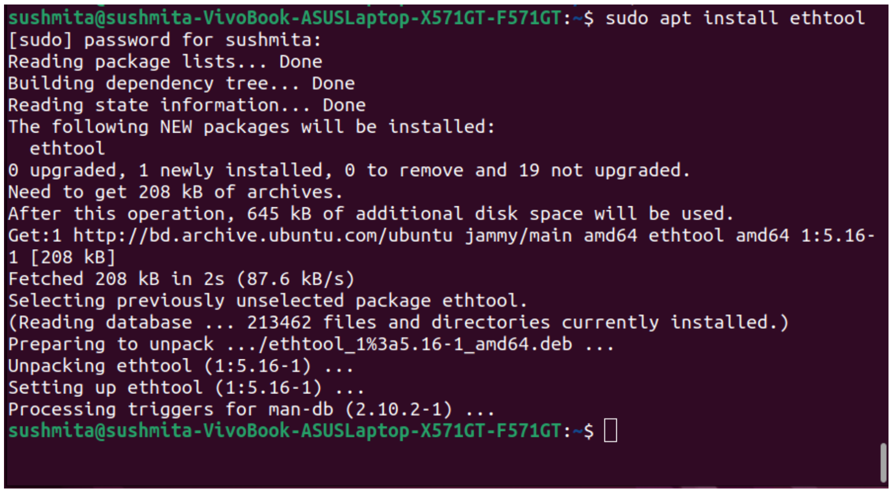
For effective network monitoring and security analysis, it is crucial to deploy Zeek in strategic locations within the network infrastructure. Below are key points outlining the recommended deployment locations for Zeek:

1. **At Network Perimeters:** Deploy Zeek at the boundaries of the network to monitor all incoming and outgoing traffic, enabling the identification of external threats and unauthorized data exfiltration.
2. **Core Network Infrastructure:** Position Zeek near central routing and switching devices to capture a broad spectrum of internal network traffic, providing insights into the core network's interactions.
3. **Data Center Entrances:** Install Zeek at the entry and exit points of data centers to oversee traffic to critical servers and storage systems, safeguarding sensitive information and key assets.
4. **Network Segmentation Gateways:** Place Zeek at the junctions between different network segments or VLANs to monitor cross-segment traffic, essential for detecting lateral movements and internal threats.
5. **VPN Concentrators and Access Points:** Deploy Zeek where VPN connections converge or near wireless access points to analyze encrypted and wireless traffic, respectively, ensuring secure remote access and wireless communication.
6. **Demilitarized Zones (DMZs):** Position Zeek within or adjacent to DMZs to scrutinize traffic between the internet and internal network, protecting services exposed to the outside world.
7. **Branch Offices:** Extend Zeek deployment to branch or remote offices to monitor localized traffic, ensuring comprehensive network security coverage across all organizational units.
8. **High Availability Clusters:** In critical network environments, deploy Zeek in a high availability configuration with load balancing to ensure continuous monitoring and redundancy, preventing single points of failure.

2 Installation

1. **Install ethtool**

```
      sudo apt install ethtool

```

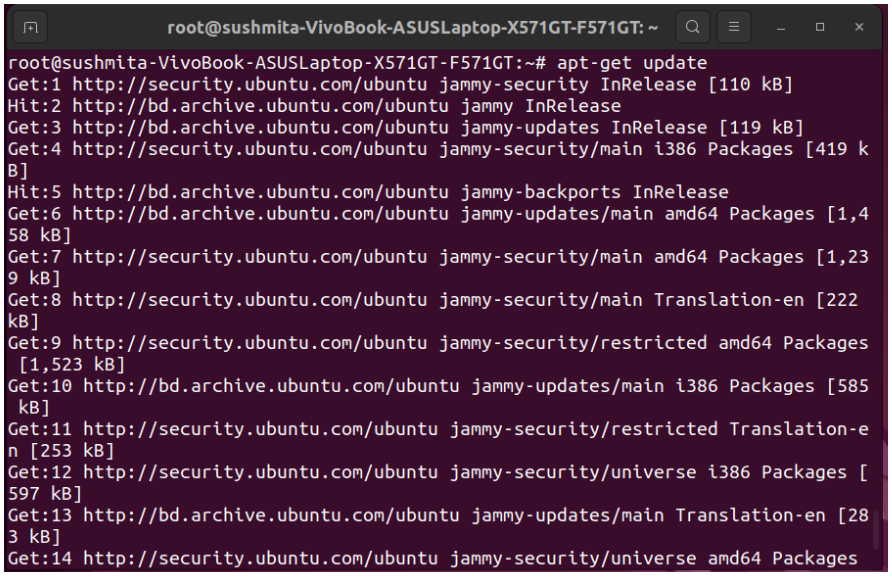


```
sushmita@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~$ sudo apt install ethtool
[sudo] password for sushmita:
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  ethtool
0 upgraded, 1 newly installed, 0 to remove and 19 not upgraded.
Need to get 208 kB of archives.
After this operation, 645 kB of additional disk space will be used.
Get:1 http://bd.archive.ubuntu.com/ubuntu jammy/main amd64 ethtool amd64 1:5.16-
1 [208 kB]
Fetched 208 kB in 2s (87.6 kB/s)
Selecting previously unselected package ethtool.
(Reading database ... 213462 files and directories currently installed.)
Preparing to unpack .../ethtool_1%3a5.16-1_amd64.deb ...
Unpacking ethtool (1:5.16-1) ...
Setting up ethtool (1:5.16-1) ...
Processing triggers for man-db (2.10.2-1) ...
sushmita@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~$ 
```

Figure 1: install ethtool

2. **Update your system**

```
      sudo apt-get update

```

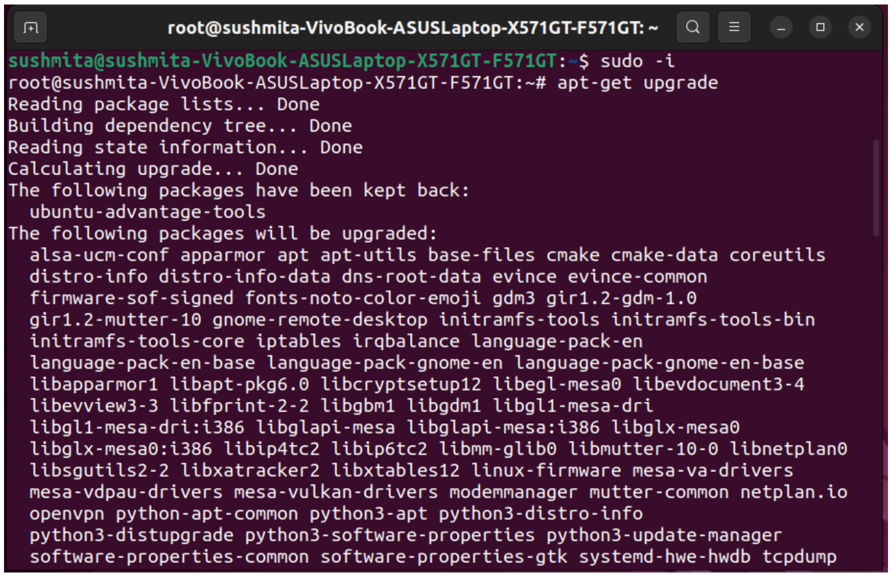


```
root@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~# apt-get update
Get:1 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:2 http://bd.archive.ubuntu.com/ubuntu jammy InRelease
Get:3 http://bd.archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Get:4 http://security.ubuntu.com/ubuntu jammy-security/main i386 Packages [419 k
B]
Hit:5 http://bd.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:6 http://bd.archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1,4
58 kB]
Get:7 http://security.ubuntu.com/ubuntu jammy-security/main amd64 Packages [1,23
9 kB]
Get:8 http://security.ubuntu.com/ubuntu jammy-security/main Translation-en [222
kB]
Get:9 http://security.ubuntu.com/ubuntu jammy-security/restricted amd64 Packages
 [1,523 kB]
Get:10 http://bd.archive.ubuntu.com/ubuntu jammy-updates/main i386 Packages [585
 kB]
Get:11 http://security.ubuntu.com/ubuntu jammy-security/restricted Translation-e
n [253 kB]
Get:12 http://security.ubuntu.com/ubuntu jammy-security/universe i386 Packages [
597 kB]
Get:13 http://bd.archive.ubuntu.com/ubuntu jammy-updates/main Translation-en [28
3 kB]
Get:14 http://security.ubuntu.com/ubuntu jammy-security/universe amd64 Packages
```

Figure 2: system update

3. **Upgrade your system**

```
        sudo apt-get upgrade

```

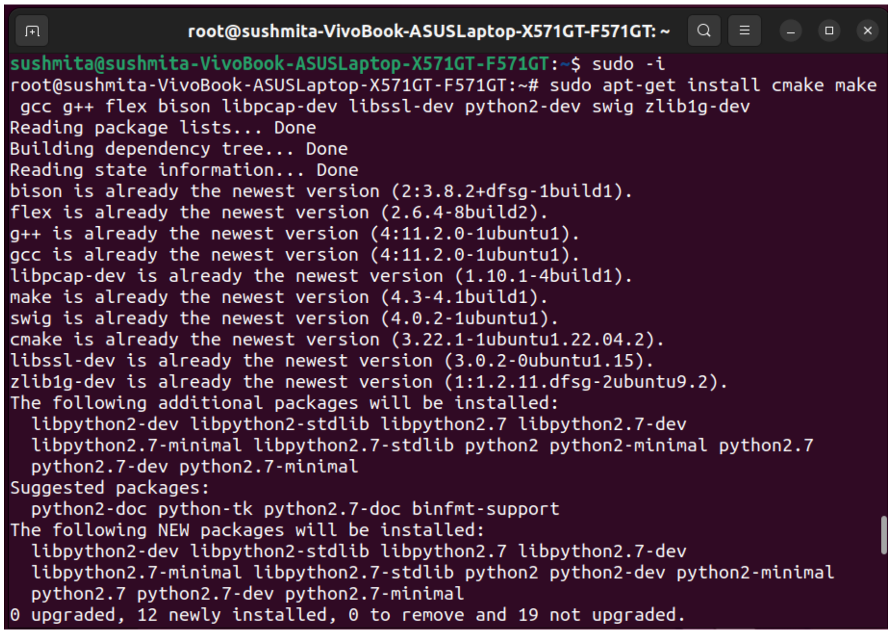


```
sushmita@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~$ sudo -i
root@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~# apt-get upgrade
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
Calculating upgrade... Done
The following packages have been kept back:
  ubuntu-advantage-tools
The following packages will be upgraded:
  alsa-ucm-conf apparmor apt apt-utils base-files cmake cmake-data coreutils
  distro-info distro-info-data dns-root-data evince evince-common
  firmware-sof-signed fonts-noto-color-emoji gdm3 gir1.2-gdm-1.0
  gir1.2-mutter-10 gnome-remote-desktop initramfs-tools initramfs-tools-bin
  initramfs-tools-core iptables irqbalance language-pack-en
  language-pack-en-base language-pack-gnome-en language-pack-gnome-en-base
  libapparmor1 libapt-pkg6.0 libcryptsetup12 libegl-mesa0 libevdocument3-4
  libevview3-3 libfprint-2-2 libgbm1 libgdm1 libgl1-mesa-dri
  libgl1-mesa-dri:i386 libglapi-mesa libglapi-mesa:i386 libglx-mesa0
  libglx-mesa0:i386 libip4tc2 libip6tc2 libmm-glib0 libmutter-10-0 libnetplan0
  libsgutils2-2 libxatracker2 libxtables12 linux-firmware mesa-va-drivers
  mesa-vdpau-drivers mesa-vulkan-drivers modemmanager mutter-common netplan.io
  openvpn python-apt-common python3-apt python3-distro-info
  python3-distupgrade python3-software-properties python3-update-manager
  software-properties-common software-properties-gtk systemd-hwe-hwdb tcpdump
```

Figure 3: system upgrade

4. **Install Zeek pre-req**

```
    sudo apt-get install cmake make gcc g++ flex bison libpcap-dev
  ↪  libssl-dev python2-dev swig zlib1g-dev
```

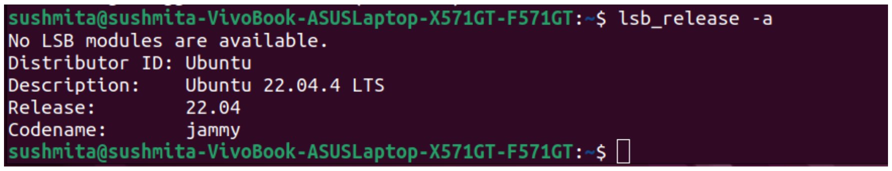


```
root@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT: ~
sushmita@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~$ sudo -i
root@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~# sudo apt-get install cmake make
 gcc g++ flex bison libpcap-dev libssl-dev python2-dev swig zlib1g-dev
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
bison is already the newest version (2:3.8.2+dfsg-1build1).
flex is already the newest version (2.6.4-8build2).
g++ is already the newest version (4:11.2.0-1ubuntu1).
gcc is already the newest version (4:11.2.0-1ubuntu1).
libpcap-dev is already the newest version (1.10.1-4build1).
make is already the newest version (4.3-4.1build1).
swig is already the newest version (4.0.2-1ubuntu1).
cmake is already the newest version (3.22.1-1ubuntu1.22.04.2).
libssl-dev is already the newest version (3.0.2-0ubuntu1.15).
zlib1g-dev is already the newest version (1:1.2.11.dfsg-2ubuntu9.2).
The following additional packages will be installed:
  libpython2-dev libpython2-stdlib libpython2.7 libpython2.7-dev
  libpython2.7-minimal libpython2.7-stdlib python2 python2-minimal python2.7
  python2.7-dev python2.7-minimal
Suggested packages:
  python2-doc python-tk python2.7-doc binfmt-support
The following NEW packages will be installed:
  libpython2-dev libpython2-stdlib libpython2.7 libpython2.7-dev
  libpython2.7-minimal libpython2.7-stdlib python2 python2-dev python2-minimal
  python2.7 python2.7-dev python2.7-minimal
0 upgraded, 12 newly installed, 0 to remove and 19 not upgraded.
```

Figure 4: Zeek pre-req

5. **Check ubuntu version**

```
    lsb_release -a
```

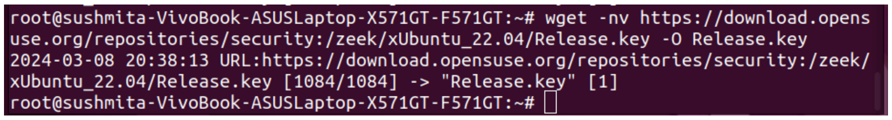


```
sushmita@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~$ lsb_release -a
No LSB modules are available.
Distributor ID: Ubuntu
Description:    Ubuntu 22.04.4 LTS
Release:        22.04
Codename:       jammy
sushmita@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~$ 
```

Figure 5: ubuntu version

6. **Add Zeek repositories to local repositories**

```
    wget -nv https://download.opensuse.org/repositories/security:/
  ↪ zeek/xUbuntu_22.04/Release.key -O Release.key
```

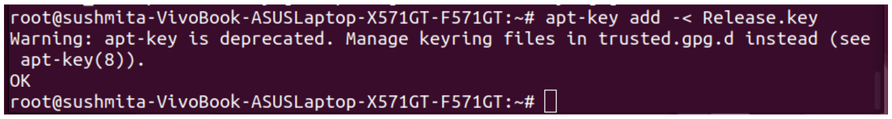


```
root@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~# wget -nv https://download.opens
use.org/repositories/security:/zeek/xUbuntu_22.04/Release.key -O Release.key
2024-03-08 20:38:13 URL:https://download.opensuse.org/repositories/security:/zeek/
xUbuntu_22.04/Release.key [1084/1084] -> "Release.key" [1]
root@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~# 
```

Figure 6: add zeek repo

7. **Add key**

```
    sudo apt-key add -< Release.key

```

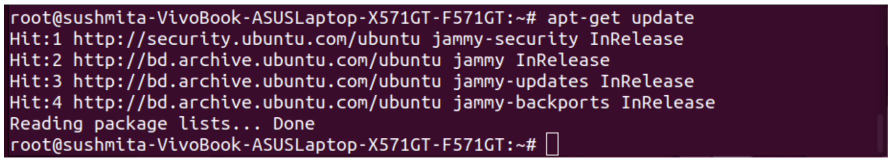


```
root@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~# apt-key add -< Release.key
Warning: apt-key is deprecated. Manage keyring files in trusted.gpg.d instead (see
 apt-key(8)).
OK
root@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~# 
```

Figure 7: add key

8. **Update package to see if there's a new repo**

```
    sudo apt-get update

```

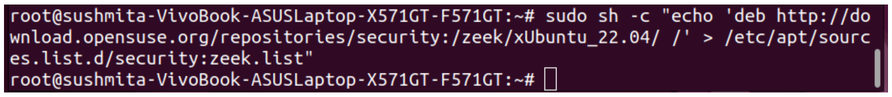


```
root@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~# apt-get update
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://bd.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://bd.archive.ubuntu.com/ubuntu jammy-updates InRelease
Hit:4 http://bd.archive.ubuntu.com/ubuntu jammy-backports InRelease
Reading package lists... Done
root@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~# 
```

Figure 8: update after add key

9. **Add repo**

```
    sudo sh -c "echo 'deb http://download.opensuse.org/
↪ repositories/security:/zeek/xUbuntu_22.04/ /' > /etc/apt/
↪ sources.list.d/security:zeek.list"

```

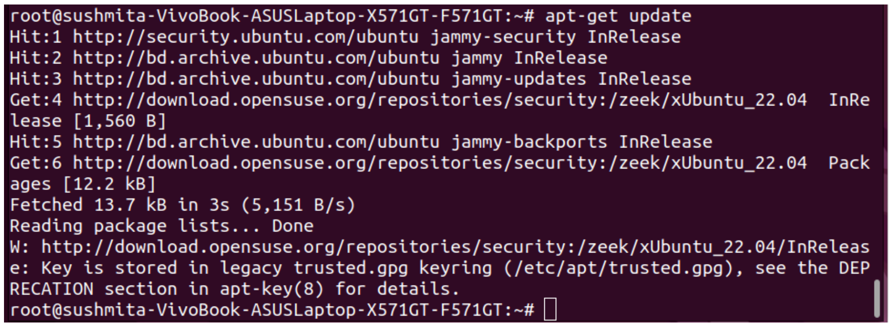


```
root@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~# sudo sh -c "echo 'deb http://do
wnload.opensuse.org/repositories/security:/zeek/xUbuntu_22.04/ /' > /etc/apt/sourc
es.list.d/security:zeek.list"
root@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~# 
```

Figure 9: add repo

10. **Update to see if we have any new repo**

```
    sudo apt-get update

```



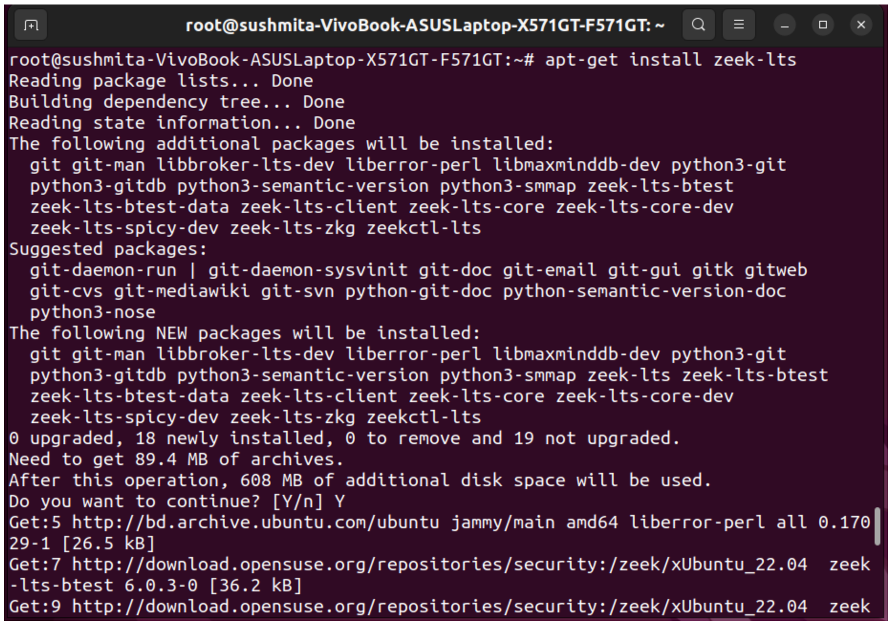
```
root@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~# apt-get update
Hit:1 http://security.ubuntu.com/ubuntu jammy-security InRelease
Hit:2 http://bd.archive.ubuntu.com/ubuntu jammy InRelease
Hit:3 http://bd.archive.ubuntu.com/ubuntu jammy-updates InRelease
Get:4 http://download.opensuse.org/repositories/security:/zeek/xUbuntu_22.04  InRe
lease [1,560 B]
Hit:5 http://bd.archive.ubuntu.com/ubuntu jammy-backports InRelease
Get:6 http://download.opensuse.org/repositories/security:/zeek/xUbuntu_22.04  Pack
ages [12.2 kB]
Fetched 13.7 kB in 3s (5,151 B/s)
Reading package lists... Done
W: http://download.opensuse.org/repositories/security:/zeek/xUbuntu_22.04/InReleas
e: Key is stored in legacy trusted.gpg keyring (/etc/apt/trusted.gpg), see the DEP
RECATION section in apt-key(8) for details.
root@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~# 
```

Figure 10: update

11. **Install Zeek**

```
    sudo apt-get install zeek-lts

```

12. **Zeek is now installed!**



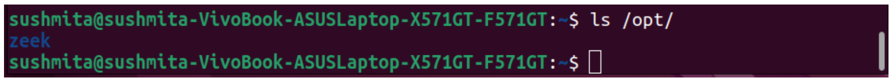
```
root@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~# apt-get install zeek-lts
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  git git-man libbroker-lts-dev liberror-perl libmaxminddb-dev python3-git
  python3-gitdb python3-semantic-version python3-smmap zeek-lts-btest
  zeek-lts-btest-data zeek-lts-client zeek-lts-core zeek-lts-core-dev
  zeek-lts-spicy-dev zeek-lts-zkg zeekctl-lts
Suggested packages:
  git-daemon-run | git-daemon-sysvinit git-doc git-email git-gui gitk gitweb
  git-cvs git-mediawiki git-svn python-git-doc python-semantic-version-doc
  python3-nose
The following NEW packages will be installed:
  git git-man libbroker-lts-dev liberror-perl libmaxminddb-dev python3-git
  python3-gitdb python3-semantic-version python3-smmap zeek-lts zeek-lts-btest
  zeek-lts-btest-data zeek-lts-client zeek-lts-core zeek-lts-core-dev
  zeek-lts-spicy-dev zeek-lts-zkg zeekctl-lts
0 upgraded, 18 newly installed, 0 to remove and 19 not upgraded.
Need to get 89.4 MB of archives.
After this operation, 608 MB of additional disk space will be used.
Do you want to continue? [Y/n] Y
Get:5 http://bd.archive.ubuntu.com/ubuntu jammy/main amd64 liberror-perl all 0.170
29-1 [26.5 kB]
Get:7 http://download.opensuse.org/repositories/security:/zeek/xUbuntu_22.04  zeek
-lts-btest 6.0.3-0 [36.2 kB]
Get:9 http://download.opensuse.org/repositories/security:/zeek/xUbuntu_22.04  zeek
```

Figure 11: install zeek

13. **To verify if we have installed**

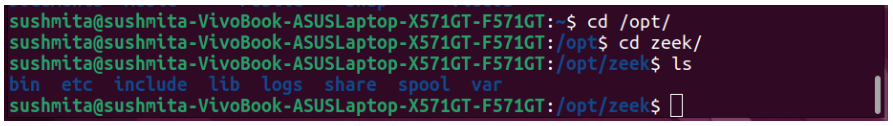
```
      ls /opt/
      cd /opt/zeek
      ls
      cd bin/
      ./zeek -h
      ./zeekctl -h

```

14. **Installation is verified!**



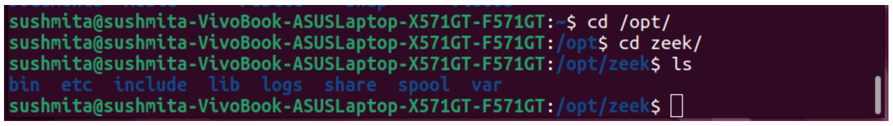
```
sushmita@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~$ ls /opt/
zeek
sushmita@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~$ 
```

Figure 12: Zeek Installation Verification 1



```
sushmita@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~$ cd /opt/
sushmita@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:/opt$ cd zeek/
sushmita@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:/opt/zeek$ ls
bin  etc  include  lib  logs  share  spool  var
sushmita@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:/opt/zeek$ 
```

Figure 13: Zeek Installation Verification 2



```
sushmita@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:~$ cd /opt/
sushmita@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:/opt$ cd zeek/
sushmita@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:/opt/zeek$ ls
bin  etc  include  lib  logs  share  spool  var
sushmita@sushmita-VivoBook-ASUSLaptop-X571GT-F571GT:/opt/zeek$ 
```

Figure 14: Zeek Installation Verification 3

3 Architecture of Zeek

Zeek, formerly known as Bro, is a powerful network analysis tool that is distinguished by its unique architecture. This architecture is primarily divided into two major components: the event engine, which processes incoming network traffic, and the script interpreter, which applies user-defined scripts to the processed data. This high-level overview aims to elucidate the roles and functionalities of these components within Zeek's architecture.

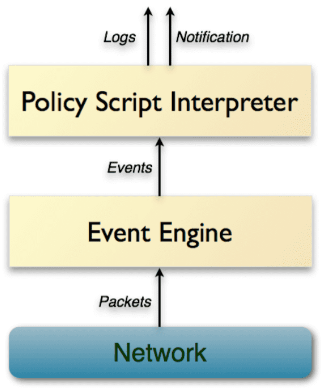


Figure 15: Architecture of Zeek

3.1 Event Engine

The event engine, or core, of Zeek plays a pivotal role in transforming the raw stream of network packets into structured, high-level events. These events are designed to be policy-neutral, focusing solely on the activities observed within the network without imparting any judgment or significance.

3.1.1 Subcomponents of the Event Engine

The event engine is further divided into several key subcomponents, each responsible for a specific aspect of packet processing:

- **Input Sources:** This subcomponent is tasked with ingesting incoming network traffic from various network interfaces.
- **Packet Analysis:** Starting at the link layer, this process involves the analysis of lower-level protocols to dissect the network traffic.
- **Session Analysis:** Focused on application-layer protocols (e.g., HTTP, FTP), session analysis examines the content and context of communication sessions.
- **File Analysis:** This involves the inspection of file content being transferred over the network, providing insights into the data being exchanged.

The modularity and plugin architecture of the event engine enable the integration and expansion of Zeek's capabilities, allowing for customized enhancements beyond the core code base.

3.2 Script Interpreter

Complementing the event engine is the script interpreter, a critical component that leverages Zeek's custom scripting language to process the events generated by the event engine. This interpreter facilitates the implementation of a site's security policies and the derivation of network statistics.

3.2.1 Functionality of the Script Interpreter

The script interpreter offers a wide range of capabilities, including but not limited to:

- **Security Policy Enforcement:** Through the execution of event handlers, scripts can define and enforce security measures in response to detected network activities.
- **Statistical Analysis:** Scripts are capable of extracting and analyzing various properties and statistics from the network traffic, enriching the understanding of network behavior.
- **State Management:** Zeek's scripting language allows for the maintenance of state over time, enabling the tracking and correlation of network events across different connections and hosts.
- **Alert Generation:** The script interpreter can produce real-time alerts and trigger external programs, facilitating proactive responses to potential security threats.

Zeek's architecture, characterized by its event engine and script interpreter, provides a robust foundation for the processing and analysis of network traffic. This design enables Zeek to deliver comprehensive insights into network activities, supporting a broad spectrum of security and monitoring applications.

4 Functionality of Zeek

Zeek stands out as a versatile platform for network security and analysis for several compelling reasons:

1. **Focused on Network Data:** Specialization in collecting and analyzing network data, a critical component for security teams.
2. **Diverse Data Collection:** Support for various data types including transaction data, extracted content, and alert data, as defined by the network security monitoring paradigm.
3. **Rich Transaction Logs:** Default generation of detailed, high-fidelity, and richly-annotated transaction logs, offering a neutral view of network activities.
4. **File Extraction Capabilities:** Robust features for carving files from network traffic, facilitating further analysis with various tools.
5. **Alert Mechanism:** Provision of an alert system through the notice mechanism, enabling judgment on network traffic.
6. **Customization and Extensibility:** Availability of a domain-specific scripting language for tailoring functionalities to specific analysis needs.
7. **Cost-Effective Deployment:** Ability to run on commodity hardware, making it an accessible solution for organizations of all sizes.
8. **Comprehensive Analysis and Detection:** Inclusion of built-in functionalities for a wide array of analysis and detection tasks, enhancing its utility as an analysis tool.

5 Log files in Zeek

5.1 conn.log

The conn.log in Zeek is crucial for tracking network connections, covering both stateful protocols like TCP and stateless ones like UDP. It provides a comprehensive view of network interactions, detailing each connection's aspects. Analysts use this log to understand network behavior, identify anomalies, and investigate incidents.

```
{
  "ts": 1591367999.430166,
  "uid": "C5bLoe2Mvxqhawzqqd",
  "id.orig_h": "192.168.4.76",
  "id.orig_p": 46378,
  "id.resp_h": "31.3.245.133",
  "id.resp_p": 80,
  "proto": "tcp",
  "service": "http",
  "duration": 0.25411510467529297,
  "orig_bytes": 77,
  "resp_bytes": 295,
  "conn_state": "SF",
  "missed_bytes": 0,
  "history": "ShADadFf",
  "orig_pkts": 6,
  "orig_ip_bytes": 397,
  "resp_pkts": 4,
  "resp_ip_bytes": 511
}
```

Listing 1: conn.log entry

A conversation between 192.168.4.76 and 31.3.245.133 occurred on Jun 5, 2020, at 14:39:59 UTC, lasting only 0.254 seconds. They communicated via HTTP on TCP port 80, with 192.168.4.76 sending 77 bytes at the application layer and 397 bytes at the IP layer.

5.2 dns.log

Zeek's dns.log captures DNS queries and responses, recording details such as the querying host, the DNS server, the requested domain, and the response. It's used for security and network analysis, helping to identify normal and malicious DNS activities, including domain resolutions and potential misuse of DNS for command-and-control or data exfiltration purposes.

```
{
  "ts": 1591367999.305988,
  "uid": "CMdzit1AMNsmfAIiQc",
  "id.orig_h": "192.168.4.76",
  "id.orig_p": 36844,
  "id.resp_h": "192.168.4.1",
  "id.resp_p": 53,
  "proto": "udp",
  "trans_id": 19671,
  "rtt": 0.06685185432434082,
  "query": "testmyids.com",
  "qclass": 1,
  "qclass_name": "C_INTERNET",
  "qtype": 1,
  "qtype_name": "A",
  "rcode": 0,
  "rcode_name": "NOERROR",
  "AA": false,
  "TC": false,
  "RD": true,
  "RA": true,
  "Z": 0,
  "answers": [
    "31.3.245.133"
  ],
  "TTLs": [
    3600
  ],
  "rejected": false
}
```

Listing 2: dns.log entry

According to this log entry, 192.168.4.76 asked 192.168.4.1 for the A record of the host testmyids.com, and received the answer 31.3.245.133.

5.3 http.log

Zeek's http.log captures details of HTTP and exposed HTTPS traffic, recording request and response information like methods, URIs, status codes, and MIME types. It's valuable for analyzing web activities, identifying suspicious or malicious behavior, and correlating with other logs like conn.log for comprehensive network insight.

```
{
  "ts": 1591367999.512593,
  "uid": "C5bLoe2Mvxqhawzqqd",
  "id.orig_h": "192.168.4.76",
  "id.orig_p": 46378,
  "id.resp_h": "31.3.245.133",
  "id.resp_p": 80,
  "trans_depth": 1,
  "method": "GET",
  "host": "testmyids.com",
  "uri": "/",
  "version": "1.1",
  "user_agent": "curl/7.47.0",
  "request_body_len": 0,
  "response_body_len": 39,
  "status_code": 200,
  "status_msg": "OK",
  "tags": [],
  "resp_fuids": [
    "FEEsZS1w0Z0VJIb5x4"
  ],
  "resp_mime_types": [
    "text/plain"
  ]
}
```

Listing 3: http.log entry

A request from 192.168.4.76 to 31.3.245.133 was logged by Zeek, showing a HTTP GET request to testmyids.com. The response was ”200 OK” with MIME type ”text/plain.” Zeek assigned a file ID to this exchange, allowing content review if configured. The unique connection ID links this HTTP interaction to the corresponding conn.log entry, facilitating comprehensive network analysis.

5.4 files.log

Zeek’s files.log records observed files in network traffic, detailing their types and transactions. Analysts can configure Zeek to extract and save specific file types, like executables, for further analysis. This process is tracked through conn.log and http.log to understand file transfer contexts.

```json
{
  "ts": 1596820191.969902,
  "fuid": "FBbQxG1GXLXgmWhbk9",
  "uid": "CzoFRWTQ6YIzfFXHk",
  "id.orig_h": "192.168.4.37",
  "id.orig_p": 58264,
  "id.resp_h": "23.195.64.241",
  "id.resp_p": 80,
  "source": "HTTP",
  "depth": 0,
  "analyzers": [
    "EXTRACT",
    "PE"
  ],
  "mime_type": "application/x-dosexec",
  "duration": 0.015498876571655273,
  "is_orig": false,
  "seen_bytes": 179272,
  "total_bytes": 179272,
  "missing_bytes": 0,
  "overflow_bytes": 0,
  "timedout": false,
  "extracted": "HTTP-FBbQxG1GXLXgmWhbk9.exe",
  "extracted_cutoff": false
}
```

Listing 4: files.log entry

The files.log entry in Zeek includes a UID from conn.log, linking the file activity to a specific connection. Zeek activated file extraction and analysis, successfully handling 179272 bytes without loss, and saved the file as HTTP-FBbQxG1GXLXgmWhbk9.exe. The is orig field indicates the file’s sender; here, the HTTP server at 23.195.64.241 sent the file to 192.168.4.37.

5.5 ftp.log

Zeek’s ftp.log records FTP communications, detailing client-server interactions including login credentials, commands, and file transfers. It captures both control channel activities and file transfer specifics, highlighting FTP’s use of separate TCP connections for commands and data.

5.6 ssl.log

Zeek's ssl.log captures metadata for TLS-encrypted sessions, including TLS versions, cipher suites, and elliptic curves. It's critical for understanding the security posture of encrypted communications, offering insights into protocol usage and potential vulnerabilities without decrypting traffic.

```
{
  "ts": 1598377391.921726,
  "uid": "CsukF91Bx9mrqdEaH9",
  "id.orig_h": "192.168.4.49",
  "id.orig_p": 56718,
  "id.resp_h": "13.32.202.10",
  "id.resp_p": 443,
  "version": "TLSv12",
  "cipher": "TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256",
  "curve": "secp256r1",
  "server_name": "www.taosecurity.com",
  "resumed": false,
  "next_protocol": "h2",
  "established": true,
  "cert_chain_fuids": [
    "F2XEvj1CahhdhtfvT4",
    "FZ7ygD3ERPfEVVohG9",
    "F7vklpOKI4yX9wmvh",
    "FAnbnR32nIIr2j9XV"
  ],
  "client_cert_chain_fuids": [],
  "subject": "CN=www.taosecurity.com",
  "issuer": "CN=Amazon,OU=Server CA 1B,O=Amazon,C=US"
}
```

Listing 5: ssl.log entry

This log entry captures a TLS 1.2 handshake between a client and a server, specifying the use of the ECDHE_RSA key exchange with AES_128_GCM_SHA256 encryption and the secp256r1 elliptic curve. It indicates a connection to "www.taosecurity.com", authenticated with a certificate chain issued by Amazon. The connection upgrades to HTTP/2 ("h2"). No client certificates are used in this session. The log is part of a series, with certificate details to be analyzed in an upcoming x509 log entry.

5.7 x509.log

Zeek's x509.log records details from TLS certificates exchanged during secure connections, providing insights into the security of these connections, including the certificate chain and key details. This log is particularly valuable for analyzing certificate validity, issuer authenticity, and encryption standards used in network communications.

5.8 smtp.log

Zeek logs SMTP traffic, providing insights into email transactions over various ports, including encrypted ones. It identifies key elements like sender, receiver, and authentication details, even in encrypted sessions, aiding in network security analysis.

5.9 ssh.log

Zeek's SSH log provides insights into SSH sessions, detailing aspects like authentication success, encryption types, and client-server interactions, aiding in the analysis of secure remote connections and potential lateral movements within networks.

5.10 pe.log

pe.log entry provides details about a portable executable file, including its architecture, operating system compatibility, security features, and structural components. It helps analysts understand the essential characteristics of the executable file quickly.

5.11 dhcp.log

dhcp.log provides detailed information about a DHCP (Dynamic Host Configuration Protocol) exchange between a client and a server. It captures the Discover-Offer-Request-Acknowledge (DORA) process, showing the messages exchanged between the client and server, including requests for IP addresses, server responses, assigned IP addresses, lease durations, and other configuration parameters. Additionally, it explains how tools like tcpdump, tshark, and Zeek (formerly known as Bro) can be used to analyze and interpret DHCP exchanges for network and security purposes.

5.12 weird.log

weird.log is various random stuff where analyzers ran into trouble understanding the traffic in terms of their protocols; basically whenever there's something unexpected at the protocol level, that's a weird (for a lack of anything better to do with it). That means that "weirds" are also essentially hardcoded by whoever wrote that analyzer. They can also be generated by scripts, but that's rarer.

5.13 notice.log

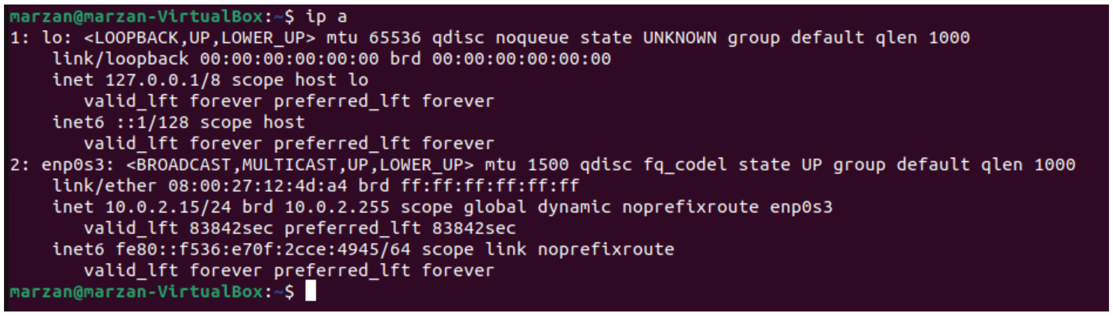
notice.log on the other hand are situations explicitly detected and reported by Zeek scripts as inspection-worthy. It's usually not protocol errors, but something semantically higher (like a self-signed cert). Notices are part of the script-level analysis and can be raised by Zeek packages as well.

6 Network Monitoring :

6.1 Live Network Monitoring:

6.1.1 Simulation :

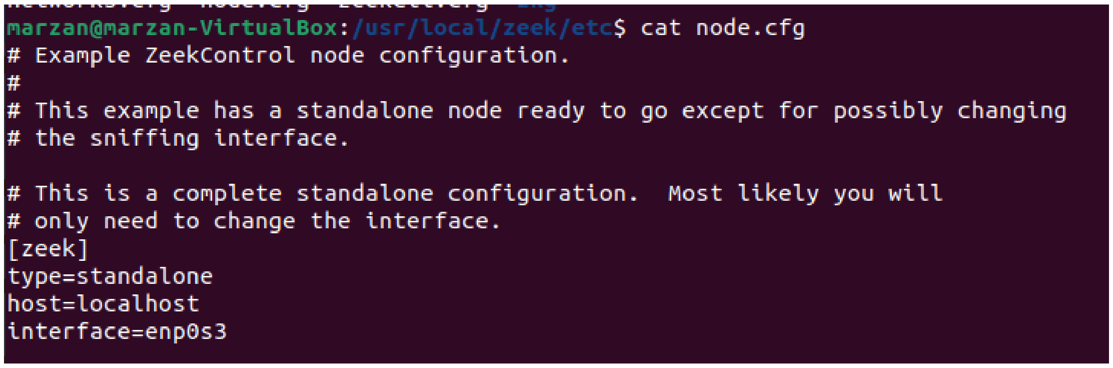
- Determine the active network interfaces on the system by executing the ip a command. This step is essential for identifying the interface through which network traffic will be monitored.



```
marzan@marzan-VirtualBox:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:12:4d:a4 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 83842sec preferred_lft 83842sec
    inet6 fe80::f536:e70f:2cce:4945/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
marzan@marzan-VirtualBox:~$
```

Figure 16: live network interfaces

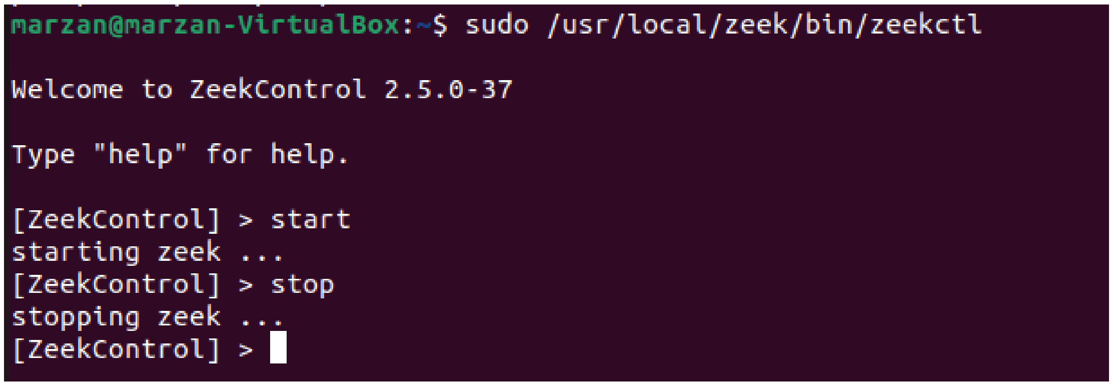
- Configure the Zeek node by editing the node.cfg file. Specify the network interface to be monitored by adding the line interface=enp0s3.



```
marzan@marzan-VirtualBox:/usr/local/zeek/etc$ cat node.cfg
# Example ZeekControl node configuration.
#
# This example has a standalone node ready to go except for possibly changing
# the sniffing interface.

# This is a complete standalone configuration.  Most likely you will
# only need to change the interface.
[zeek]
type=standalone
host=localhost
interface=enp0s3
```

Figure 17: node.cfg update

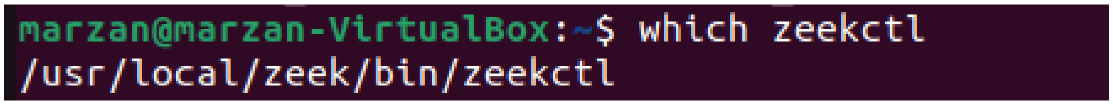
- Utilize the zeekctl command-line utility to manage the Zeek instance. Initiate Zeek by executing zeekctl start, and when necessary, halt the monitoring process using zeekctl stop.



```
marzan@marzan-VirtualBox:~$ sudo /usr/local/zeek/bin/zeekctl

Welcome to ZeekControl 2.5.0-37

Type "help" for help.

[ZeekControl] > start
starting zeek ...
[ZeekControl] > stop
stopping zeek ...
[ZeekControl] >
```

Figure 18: zeekctl

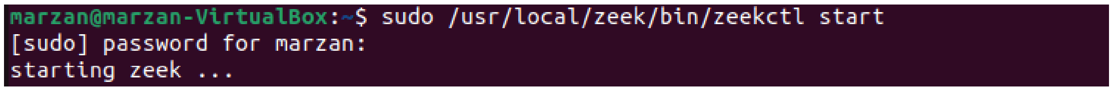
- To ascertain the installation path of the zeekctl utility, employ the command which zeekctl. This command reveals the full path to the zeekctl executable, facilitating its execution.



```
marzan@marzan-VirtualBox:~$ which zeekctl
/usr/local/zeek/bin/zeekctl
```

Figure 19: zeekctl path

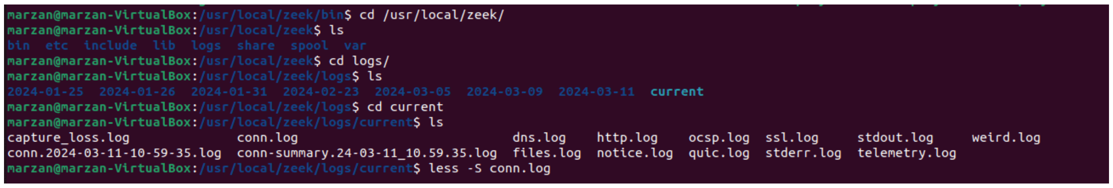
- Activate the Zeek control application by executing start zeekctl, which initiates the live monitoring process.



```
marzan@marzan-VirtualBox:~$ sudo /usr/local/zeek/bin/zeekctl start
[sudo] password for marzan:
starting zeek ...
```

Figure 20: zeekctl start

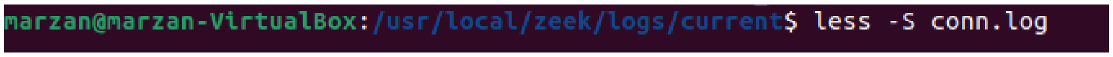
- Upon commencement of live monitoring, Zeek begins to capture and analyze network traffic, systematically storing the results in a series of log files. These logs, encompassing various protocols and activities, include but are not limited to dns.log, http.log, and dhcp.log.



```
marzan@marzan-VirtualBox:/usr/local/zeek/bin$ cd /usr/local/zeek/
marzan@marzan-VirtualBox:/usr/local/zeek$ ls
bin  etc  include  lib  logs  share  spool  var
marzan@marzan-VirtualBox:/usr/local/zeek$ cd logs/
marzan@marzan-VirtualBox:/usr/local/zeek/logs$ ls
2024-01-25  2024-01-26  2024-01-31  2024-02-23  2024-03-05  2024-03-09  2024-03-11  current
marzan@marzan-VirtualBox:/usr/local/zeek/logs$ cd current
marzan@marzan-VirtualBox:/usr/local/zeek/logs/current$ ls
capture_loss.log             conn.log                            dns.log    http.log    ocsp.log  ssl.log     stdout.log     weird.log
conn.2024-03-11-10-59-35.log  conn-summary.24-03-11_10.59.35.log  files.log  notice.log  quic.log  stderr.log  telemetry.log
marzan@marzan-VirtualBox:/usr/local/zeek/logs/current$ less -S conn.log
```

Figure 21: Log files

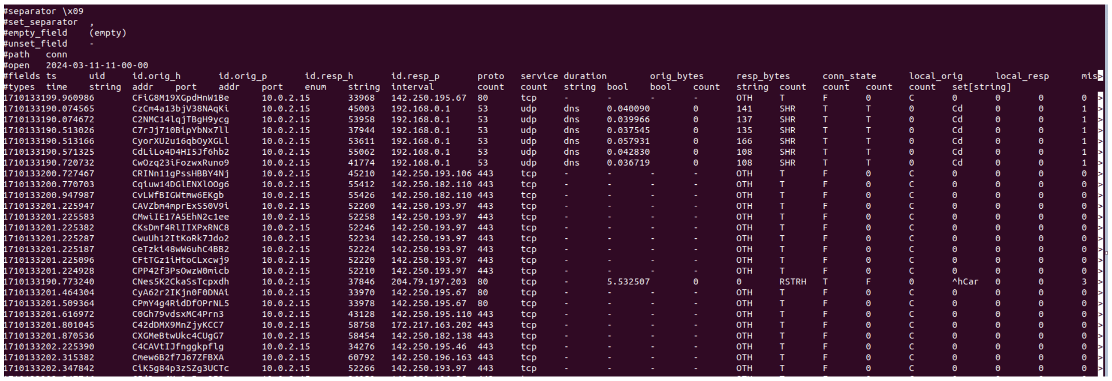
- To examine the connection log (conn.log), the command less -S conn.log is employed. This command provides a paginated view of the conn.log file, allowing for detailed inspection of recorded network connections.



```
marzan@marzan-VirtualBox:/usr/local/zeek/logs/current$ less -S conn.log
```

Figure 22: conn.log

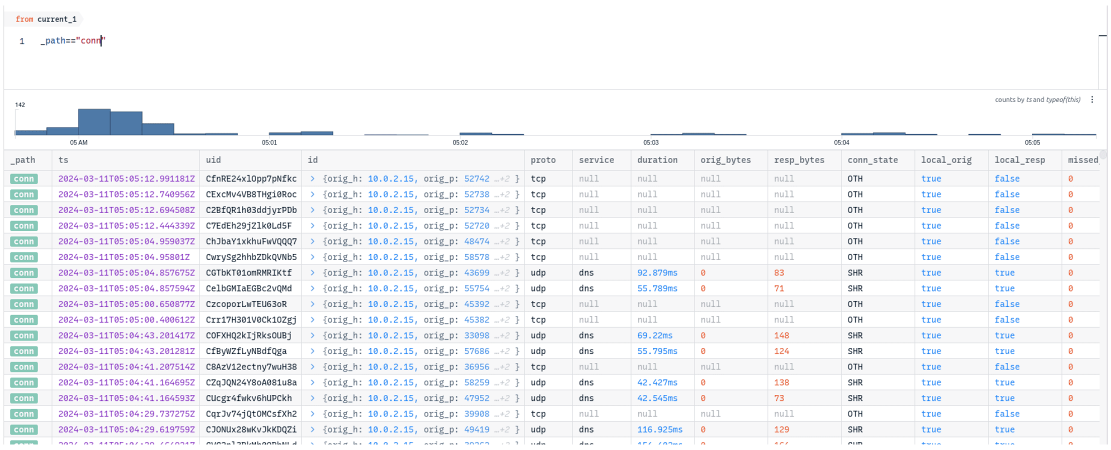
- Further scrutiny of the conn.log file can be achieved through direct observation, facilitating a deeper understanding of the network interactions and anomalies.



```
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field    -
#path   conn
#open   2024-03-11-11-00-00
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp_p       proto   service duration        orig_bytes      resp_bytes      conn_state      local_orig      local_resp      mis>
#types  time    string  addr    port    addr    port    enum    string  interval        count   count   string  bool    bool    count   set[string]
1710133199.960986       CFiG8M19XGpdHnW1Be      10.0.2.15       33968   142.250.195.67  80      tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0   >
1710133190.074565       CzCm4a13bjV38NAqKi      10.0.2.15       45003   192.168.0.1     53      udp     dns     0.040090        0       141     SHR     T       T       0       Cd      0       0       1   >
1710133190.074672       C2NMC14lqjTBgH9ycg      10.0.2.15       53958   192.168.0.1     53      udp     dns     0.039966        0       137     SHR     T       T       0       Cd      0       0       1   >
1710133190.513026       C7rJj710BipYbNx7ll      10.0.2.15       37944   192.168.0.1     53      udp     dns     0.037545        0       135     SHR     T       T       0       Cd      0       0       1   >
1710133190.513166       CyorXU2u16qbOyXGLl      10.0.2.15       53611   192.168.0.1     53      udp     dns     0.057931        0       166     SHR     T       T       0       Cd      0       0       1   >
1710133190.571325       CdLiLo4D4HI5Jf6hb2      10.0.2.15       55062   192.168.0.1     53      udp     dns     0.042830        0       108     SHR     T       T       0       Cd      0       0       1   >
1710133190.720732       CwOzq23iFozwxRuno9      10.0.2.15       41774   192.168.0.1     53      udp     dns     0.036719        0       108     SHR     T       T       0       Cd      0       0       1   >
1710133200.727467       CRINn11gPssHBBY4Nj      10.0.2.15       45210   142.250.193.106 443     tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0   >
1710133200.770703       Cqiuw14DGlENXlOOg6      10.0.2.15       55412   142.250.182.110 443     tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0   >
1710133200.947987       CvLWfBIGWtmw6EKgb       10.0.2.15       55426   142.250.182.110 443     tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0   >
1710133201.225947       CAVZbm4mprExS50V9i      10.0.2.15       52260   142.250.193.97  443     tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0   >
1710133201.225583       CMwiIE17A5EhN2c1ee      10.0.2.15       52258   142.250.193.97  443     tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0   >
1710133201.225382       CKsDmf4RlIIXPxRNC8      10.0.2.15       52246   142.250.193.97  443     tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0   >
1710133201.225287       CwuUh12ItKoRk7Jdo2      10.0.2.15       52234   142.250.193.97  443     tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0   >
1710133201.225187       CeTzki48wW6uhC4BB2      10.0.2.15       52224   142.250.193.97  443     tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0   >
1710133201.225096       CFtTGz1iHtoCLxcwj9      10.0.2.15       52220   142.250.193.97  443     tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0   >
1710133201.224928       CPP42f3PsOwzW0micb      10.0.2.15       52210   142.250.193.97  443     tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0   >
1710133190.773240       CNes5K2CkaSsTcpxdh      10.0.2.15       37846   204.79.197.203  80      tcp     -       5.532507        0       0       RSTRH   T       F       0       ^hCar   0       0       3   >
1710133201.464304       CyA62r2IKjn0F0DNAi      10.0.2.15       33970   142.250.195.67  80      tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0   >
1710133201.509364       CPmY4g4RidDfOPrNL5      10.0.2.15       33978   142.250.195.67  80      tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0   >
1710133201.616972       C0Gh79vdsxMC4Prn3       10.0.2.15       43128   142.250.195.110 443     tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0   >
1710133201.801045       C42dDMX9MnZjyKCC7       10.0.2.15       58758   172.217.163.202 443     tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0   >
1710133201.870536       CXGMeBtwUkc4CUgG7       10.0.2.15       58454   142.250.182.138 443     tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0   >
1710133202.225390       C4CAVtIJfnggkpflg       10.0.2.15       34276   142.250.195.46  443     tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0   >
1710133202.315382       Cmew6B2f7J67ZFBXA       10.0.2.15       60792   142.250.196.163 443     tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0   >
1710133202.347842       CiK5g84p3zSZg3UCTc      10.0.2.15       52266   142.250.193.97  443     tcp     -       -       -       -       OTH     T       F       0       C       0       0       0       0   >
```

Figure 23: conn.log

- For an enhanced analysis experience, the conn.log file may be imported into the Brim application. Brim provides a graphical interface for interacting with Zeek logs, offering advanced search capabilities and visualizations to aid in the interpretation of network data.

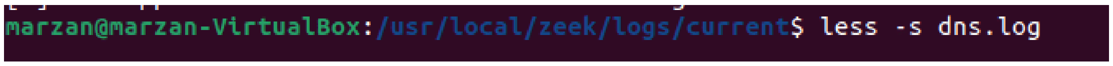


```
from current_1
_path=="conn"
```

_path	ts	uid	id	proto	service	duration	orig_bytes	resp_bytes	conn_state	local_orig	local_resp	missed
conn	2024-03-11T05:05:12.991181Z	CfnRE24xlOpp7pNfkc	› {orig_h: 10.0.2.15, orig_p: 52742 …+2 }	tcp	null	null	null	null	OTH	true	false	0
conn	2024-03-11T05:05:12.740956Z	CExcMv4VB8THgi0Roc	› {orig_h: 10.0.2.15, orig_p: 52738 …+2 }	tcp	null	null	null	null	OTH	true	false	0
conn	2024-03-11T05:05:12.694508Z	C2BfQR1h03ddjyrPDb	› {orig_h: 10.0.2.15, orig_p: 52734 …+2 }	tcp	null	null	null	null	OTH	true	false	0
conn	2024-03-11T05:05:12.444339Z	C7EdEh29jZlk0Ld5F	› {orig_h: 10.0.2.15, orig_p: 52720 …+2 }	tcp	null	null	null	null	OTH	true	false	0
conn	2024-03-11T05:05:04.959037Z	ChJbaY1xkhuFwVQQQ7	› {orig_h: 10.0.2.15, orig_p: 48474 …+2 }	tcp	null	null	null	null	OTH	true	false	0
conn	2024-03-11T05:05:04.95801Z	CwrySg2hhbZDkQVNb5	› {orig_h: 10.0.2.15, orig_p: 58578 …+2 }	tcp	null	null	null	null	OTH	true	false	0
conn	2024-03-11T05:05:04.857675Z	CGTbKT01omRMRIKtf	› {orig_h: 10.0.2.15, orig_p: 43699 …+2 }	udp	dns	92.879ms	0	83	SHR	true	true	0
conn	2024-03-11T05:05:04.857594Z	CelbGMIaEGBc2vQMd	› {orig_h: 10.0.2.15, orig_p: 55754 …+2 }	udp	dns	55.789ms	0	71	SHR	true	true	0
conn	2024-03-11T05:05:00.650877Z	CzcoporLwTEU63oR	› {orig_h: 10.0.2.15, orig_p: 45392 …+2 }	tcp	null	null	null	null	OTH	true	false	0
conn	2024-03-11T05:05:00.400612Z	Crr17H301V0Ck1OZgj	› {orig_h: 10.0.2.15, orig_p: 45382 …+2 }	tcp	null	null	null	null	OTH	true	false	0
conn	2024-03-11T05:04:43.201417Z	COFXHQ2kIjRksOUBj	› {orig_h: 10.0.2.15, orig_p: 33098 …+2 }	udp	dns	69.22ms	0	148	SHR	true	true	0
conn	2024-03-11T05:04:43.201281Z	CfByWZfLyNBdfQga	› {orig_h: 10.0.2.15, orig_p: 57686 …+2 }	udp	dns	55.795ms	0	124	SHR	true	true	0
conn	2024-03-11T05:04:41.207514Z	C8AzV12ectny7wuH38	› {orig_h: 10.0.2.15, orig_p: 36956 …+2 }	tcp	null	null	null	null	OTH	true	false	0
conn	2024-03-11T05:04:41.164695Z	CZqJQN24Y8oA081u8a	› {orig_h: 10.0.2.15, orig_p: 58259 …+2 }	udp	dns	42.427ms	0	138	SHR	true	true	0
conn	2024-03-11T05:04:41.164593Z	CUcgr4fwkv6hUPCkh	› {orig_h: 10.0.2.15, orig_p: 47952 …+2 }	udp	dns	42.545ms	0	73	SHR	true	true	0
conn	2024-03-11T05:04:29.737275Z	CqrJv74jQtOMCsfXh2	› {orig_h: 10.0.2.15, orig_p: 39908 …+2 }	tcp	null	null	null	null	OTH	true	false	0
conn	2024-03-11T05:04:29.619759Z	CJONUx28wKvJkKDQZi	› {orig_h: 10.0.2.15, orig_p: 49419 …+2 }	udp	dns	116.925ms	0	129	SHR	true	true	0

Figure 24: conn.log in BRIM

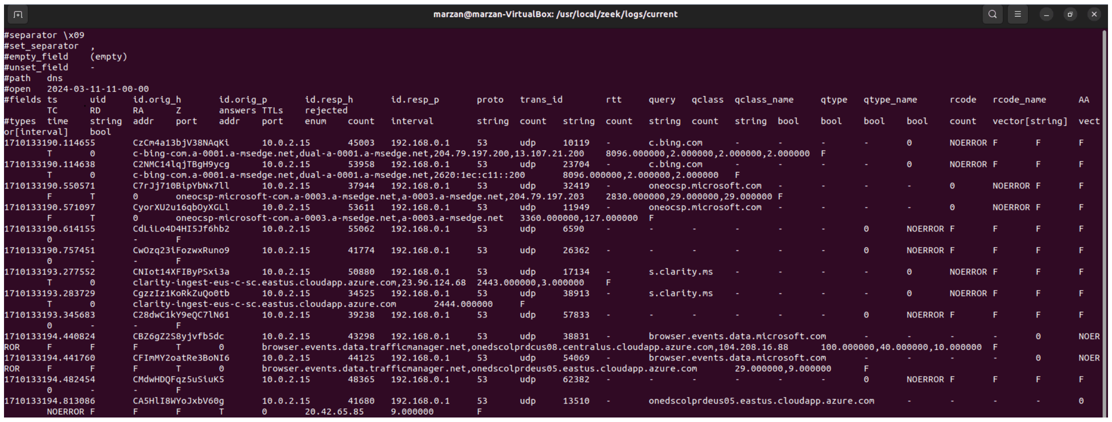
- To check out the DNS activity,we have opened the dns.log file using less -S dns.log. This shows the DNS details page by page, making it easier to see what's happening.



```
marzan@marzan-VirtualBox:/usr/local/zeek/logs/current$ less -s dns.log
```

Figure 25: dns.log

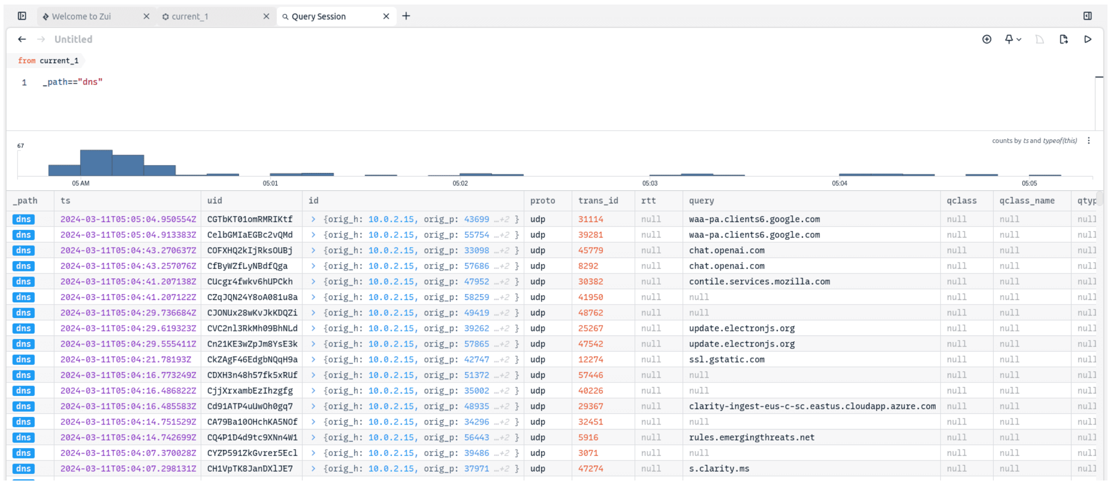
- Direct analysis of dns.log elucidates DNS traffic patterns and anomalies.



```
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field    -
#path   dns
#open   2024-03-11-11-00-00
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp_p       proto   trans_id        rtt     query   qclass  qclass_name     qtype   qtype_name      rcode   rcode_name      AA
        TC      RD      RA      Z       answers TTLs    rejected
#types  time    string  addr    port    addr    port    enum    count   interval        string  count   string  count   string  bool    bool    bool    bool    count   vector[string]  vect
or[interval]    bool
1710133190.114655       CzCm4a13bjV38NAqKi      10.0.2.15       45003   192.168.0.1     53      udp     10119   -       c.bing.com      -       -       -       -       0       NOERROR F       F       F
        T       0       c-bing-com.a-0001.a-msedge.net,dual-a-0001.a-msedge.net,204.79.197.200,13.107.21.200    8096.000000,2.000000,2.000000,2.000000  F
1710133190.114638       C2NMC14lqjTBgH9ycg      10.0.2.15       53958   192.168.0.1     53      udp     23704   -       c.bing.com      -       -       -       -       0       NOERROR F       F       F
        T       0       c-bing-com.a-0001.a-msedge.net,dual-a-0001.a-msedge.net,2620:1ec:c11::200       8096.000000,2.000000,2.000000   F
1710133190.550571       C7rJj710BipYbNx7ll      10.0.2.15       37944   192.168.0.1     53      udp     32419   -       oneocsp.microsoft.com   -       -       -       -       0       NOERROR F       F
        F       T       0       oneocsp-microsoft-com.a-0003.a-msedge.net,a-0003.a-msedge.net,204.79.197.203    2830.000000,29.000000,29.000000 F
1710133190.571097       CyorXU2u16qbOyXGLl      10.0.2.15       53611   192.168.0.1     53      udp     11949   -       oneocsp.microsoft.com   -       -       -       -       0       NOERROR F       F
        F       T       0       oneocsp-microsoft-com.a-0003.a-msedge.net,a-0003.a-msedge.net   3360.000000,127.000000  F
1710133190.614155       CdLiLo4D4HI5Jf6hb2      10.0.2.15       55062   192.168.0.1     53      udp     6590    -       -       -       -       -       -       0       NOERROR F       F       F       F
        0       -       -       F
1710133190.757451       CwOzq23iFozwxRuno9      10.0.2.15       41774   192.168.0.1     53      udp     26362   -       -       -       -       -       -       0       NOERROR F       F       F       F
        0       -       -       F
1710133193.277552       CNIot14XFIByPSxi3a      10.0.2.15       50880   192.168.0.1     53      udp     17134   -       s.clarity.ms    -       -       -       -       0       NOERROR F       F       F
        T       0       clarity-ingest-eus-c-sc.eastus.cloudapp.azure.com,23.96.124.68  2443.000000,3.000000    F
1710133193.283729       CgzzIz1KoRkZuQo0tb      10.0.2.15       34525   192.168.0.1     53      udp     38913   -       s.clarity.ms    -       -       -       -       0       NOERROR F       F       F
        T       0       clarity-ingest-eus-c-sc.eastus.cloudapp.azure.com       2444.000000     F
1710133193.345683       C28dwC1kY9eQC7lN61      10.0.2.15       39238   192.168.0.1     53      udp     57833   -       -       -       -       -       -       0       NOERROR F       F       F       F
        0       -       -       F
1710133194.440824       CBZ6gZ2S8yjvfb5dc       10.0.2.15       43298   192.168.0.1     53      udp     38831   -       browser.events.data.microsoft.com       -       -       -       -       0       NOER
ROR     F       F       F       T       0       browser.events.data.trafficmanager.net,onedscolprdcus08.centralus.cloudapp.azure.com,104.208.16.88      100.000000,40.000000,10.000000  F
1710133194.441760       CFImMY2oatRe3BoNI6      10.0.2.15       44125   192.168.0.1     53      udp     54069   -       browser.events.data.microsoft.com       -       -       -       -       0       NOER
ROR     F       F       F       T       0       browser.events.data.trafficmanager.net,onedscolprdeus05.eastus.cloudapp.azure.com       29.000000,9.000000      F
1710133194.482454       CMdwHDQFqz5uSiuK5       10.0.2.15       48365   192.168.0.1     53      udp     62382   -       -       -       -       -       -       0       NOERROR F       F       F       F
        0       -       -       F
1710133194.813086       CA5HlI8WYoJxbV60g       10.0.2.15       41680   192.168.0.1     53      udp     13510   -       onedscolprdeus05.eastus.cloudapp.azure.com      -       -       -       -       0
        NOERROR F       F       F       T       0       20.42.65.85     9.000000        F
```

Figure 26: dns.log

- For advanced analysis, dns.log is imported into Brim. Brim's GUI and analytical tools enhance DNS log data interpretation, offering clarity on network DNS behavior.



```
from current_1
_path=="dns"
```

_path	ts	uid	id	proto	trans_id	rtt	query	qclass	qclass_name	qtyp
dns	2024-03-11T05:05:04.950554Z	CGTbKT01omRMRIKtf	{orig_h: 10.0.2.15, orig_p: 43699 …+2 }	udp	31114	null	waa-pa.clients6.google.com	null	null	null
dns	2024-03-11T05:05:04.913383Z	CelbGMIaEGBc2vQMd	{orig_h: 10.0.2.15, orig_p: 55754 …+2 }	udp	39281	null	waa-pa.clients6.google.com	null	null	null
dns	2024-03-11T05:04:43.270637Z	COFXHQ2kIjRksOUBj	{orig_h: 10.0.2.15, orig_p: 33098 …+2 }	udp	45779	null	chat.openai.com	null	null	null
dns	2024-03-11T05:04:43.257076Z	CfByWZfLyNBdfQga	{orig_h: 10.0.2.15, orig_p: 57686 …+2 }	udp	8292	null	chat.openai.com	null	null	null
dns	2024-03-11T05:04:41.207138Z	CUcgr4fwkv6hUPCkh	{orig_h: 10.0.2.15, orig_p: 47952 …+2 }	udp	30382	null	contile.services.mozilla.com	null	null	null
dns	2024-03-11T05:04:41.207122Z	CZqJQN24Y8oA081u8a	{orig_h: 10.0.2.15, orig_p: 58259 …+2 }	udp	41950	null	null	null	null	null
dns	2024-03-11T05:04:29.736684Z	CJONUx28wKvJkKDQZi	{orig_h: 10.0.2.15, orig_p: 49419 …+2 }	udp	48762	null	null	null	null	null
dns	2024-03-11T05:04:29.619323Z	CVC2nl3RkMh09BhNLd	{orig_h: 10.0.2.15, orig_p: 39262 …+2 }	udp	25267	null	update.electronjs.org	null	null	null
dns	2024-03-11T05:04:29.555411Z	Cn21KE3wZpJm8YsE3k	{orig_h: 10.0.2.15, orig_p: 57865 …+2 }	udp	47542	null	update.electronjs.org	null	null	null
dns	2024-03-11T05:04:21.78193Z	CkZAgF46EdgbNQqH9a	{orig_h: 10.0.2.15, orig_p: 42747 …+2 }	udp	12274	null	ssl.gstatic.com	null	null	null
dns	2024-03-11T05:04:16.773249Z	CDXH3n48h57fk5xRUf	{orig_h: 10.0.2.15, orig_p: 51372 …+2 }	udp	57446	null	null	null	null	null
dns	2024-03-11T05:04:16.486822Z	CjjXrxambEzIhzgfg	{orig_h: 10.0.2.15, orig_p: 35002 …+2 }	udp	40226	null	null	null	null	null
dns	2024-03-11T05:04:16.485583Z	Cd91ATP4uUwOh0gq7	{orig_h: 10.0.2.15, orig_p: 48935 …+2 }	udp	29367	null	clarity-ingest-eus-c-sc.eastus.cloudapp.azure.com	null	null	null
dns	2024-03-11T05:04:14.751529Z	CA79Ba10OHchKA5NOf	{orig_h: 10.0.2.15, orig_p: 34296 …+2 }	udp	32451	null	null	null	null	null
dns	2024-03-11T05:04:14.742699Z	CQ4P1D4d9tc9XNn4W1	{orig_h: 10.0.2.15, orig_p: 56443 …+2 }	udp	5916	null	rules.emergingthreats.net	null	null	null
dns	2024-03-11T05:04:07.370028Z	CYZP591ZkGvrer5Ecl	{orig_h: 10.0.2.15, orig_p: 39486 …+2 }	udp	3071	null	null	null	null	null
dns	2024-03-11T05:04:07.298131Z	CH1VpTK8JanDXlJE7	{orig_h: 10.0.2.15, orig_p: 37971 …+2 }	udp	47274	null	s.clarity.ms	null	null	null

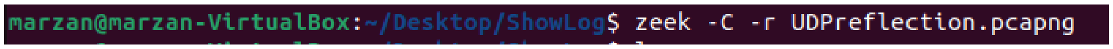
Figure 27: dns.log in BRIM

6.2 Network Monitoring from PCAP file:

Here is the link to PCAP file: malware-traffic-analysis

6.2.1 Simulation :

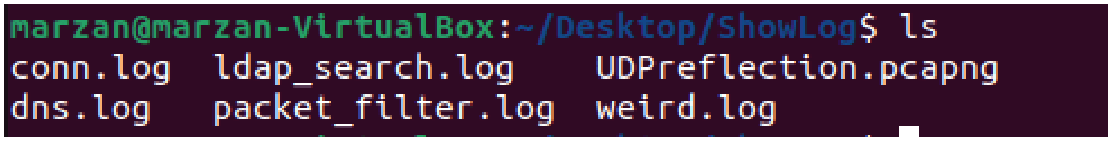
- Begin by executing the command zeek -C -r a.pcap, which initiates the analysis of the provided PCAP file. The -C flag is used to ignore checksums, ensuring that all packets are analyzed irrespective of checksum validity.



```
marzan@marzan-VirtualBox:~/Desktop/ShowLog$ zeek -C -r UDPreflection.pcapng
```

Figure 28: zeek analysis of pcap file

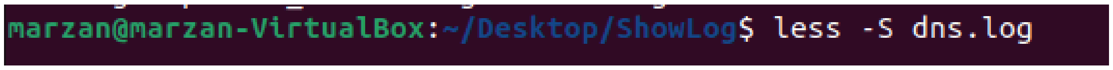
- Upon completion of the analysis, Zeek will generate a series of log files, including but not limited to conn.log, dns.log, and weird.log, each containing detailed information pertinent to their respective network protocols and events.



```
marzan@marzan-VirtualBox:~/Desktop/ShowLog$ ls
conn.log  ldap_search.log    UDPreflection.pcapng
dns.log   packet_filter.log  weird.log
```

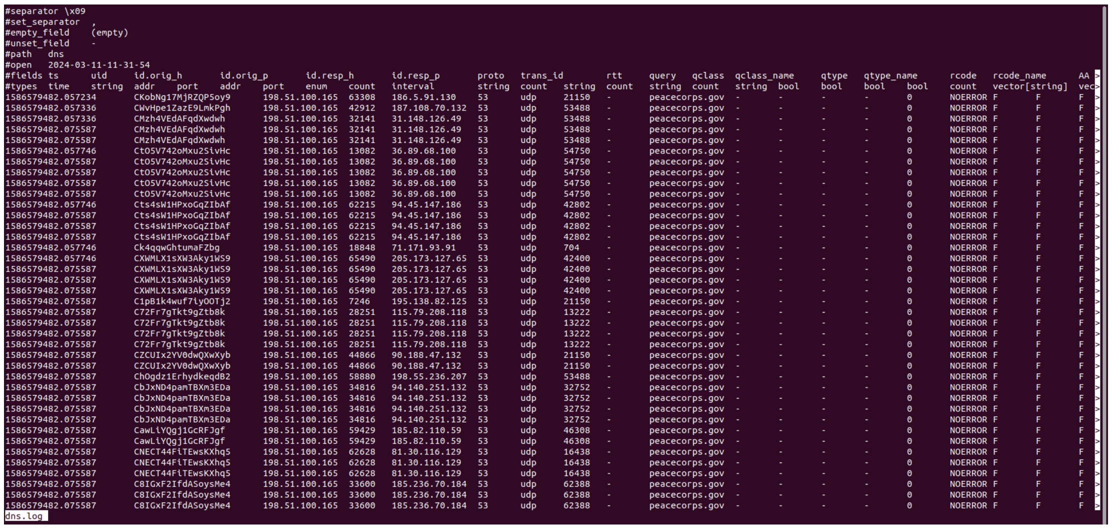
Figure 29: Log files

- To review the DNS activity captured during the analysis, employ the command less -S dns.log. This command will present the contents of the dns.log file in a paginated manner, allowing for a detailed examination of DNS transactions.



```
marzan@marzan-VirtualBox:~/Desktop/ShowLog$ less -S dns.log
```

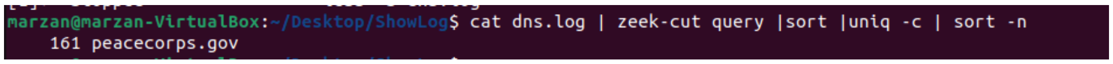
Figure 30: dns.log



```
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field    -
#path   dns
#open   2024-03-11-11-31-54
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp_p       proto   trans_id        rtt     query   qclass  qclass_name     qtype   qtype_name      rcode   rcode_name      AA >
#types  time    string  addr    port    addr    port    enum    count   interval        string  count   string  count   string  bool    bool    bool    count   vector[string]  vec>
1586579482.057234       CKobNg17MjRZQP5oy9      198.51.100.165  63308   186.5.91.130    53      udp     21150   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.057336       CWvHpe1ZazE9LmkPgh      198.51.100.165  42912   187.108.70.132  53      udp     53488   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.057336       CMzh4VEdAFqdXwdwh       198.51.100.165  32141   31.148.126.49   53      udp     53488   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       CMzh4VEdAFqdXwdwh       198.51.100.165  32141   31.148.126.49   53      udp     53488   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       CMzh4VEdAFqdXwdwh       198.51.100.165  32141   31.148.126.49   53      udp     53488   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.057746       CtO5V742oMxu2SivHc      198.51.100.165  13082   36.89.68.100    53      udp     54750   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       CtO5V742oMxu2SivHc      198.51.100.165  13082   36.89.68.100    53      udp     54750   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       CtO5V742oMxu2SivHc      198.51.100.165  13082   36.89.68.100    53      udp     54750   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       CtO5V742oMxu2SivHc      198.51.100.165  13082   36.89.68.100    53      udp     54750   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       CtO5V742oMxu2SivHc      198.51.100.165  13082   36.89.68.100    53      udp     54750   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.057746       Cts4sW1HPxoGqZIbAf      198.51.100.165  62215   94.45.147.186   53      udp     42802   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       Cts4sW1HPxoGqZIbAf      198.51.100.165  62215   94.45.147.186   53      udp     42802   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       Cts4sW1HPxoGqZIbAf      198.51.100.165  62215   94.45.147.186   53      udp     42802   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       Cts4sW1HPxoGqZIbAf      198.51.100.165  62215   94.45.147.186   53      udp     42802   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.057746       Ck4qqwGhtumaFZbg        198.51.100.165  18848   71.171.93.91    53      udp     704     -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.057746       CXWMLX1sXW3Aky1WS9      198.51.100.165  65490   205.173.127.65  53      udp     42400   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       CXWMLX1sXW3Aky1WS9      198.51.100.165  65490   205.173.127.65  53      udp     42400   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       CXWMLX1sXW3Aky1WS9      198.51.100.165  65490   205.173.127.65  53      udp     42400   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       CXWMLX1sXW3Aky1WS9      198.51.100.165  65490   205.173.127.65  53      udp     42400   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       C1pB1k4wuf7iyOOTj2      198.51.100.165  7246    195.138.82.125  53      udp     21150   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       C72Fr7gTkt9gZtb8k       198.51.100.165  28251   115.79.208.118  53      udp     13222   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       C72Fr7gTkt9gZtb8k       198.51.100.165  28251   115.79.208.118  53      udp     13222   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       C72Fr7gTkt9gZtb8k       198.51.100.165  28251   115.79.208.118  53      udp     13222   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       C72Fr7gTkt9gZtb8k       198.51.100.165  28251   115.79.208.118  53      udp     13222   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       CZCUIx2YV0dwQXwXyb      198.51.100.165  44866   90.188.47.132   53      udp     21150   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       CZCUIx2YV0dwQXwXyb      198.51.100.165  44866   90.188.47.132   53      udp     21150   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       ChOgdz1ErhydkeqdB2      198.51.100.165  58880   198.55.236.207  53      udp     53488   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       CbJxND4pamTBXm3EDa      198.51.100.165  34816   94.140.251.132  53      udp     32752   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       CbJxND4pamTBXm3EDa      198.51.100.165  34816   94.140.251.132  53      udp     32752   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       CbJxND4pamTBXm3EDa      198.51.100.165  34816   94.140.251.132  53      udp     32752   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       CbJxND4pamTBXm3EDa      198.51.100.165  34816   94.140.251.132  53      udp     32752   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       CawLiYQgj1GcRFJgf       198.51.100.165  59429   185.82.110.59   53      udp     46308   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       CawLiYQgj1GcRFJgf       198.51.100.165  59429   185.82.110.59   53      udp     46308   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       CNECT44FiTEwsKXhq5      198.51.100.165  62628   81.30.116.129   53      udp     16438   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       CNECT44FiTEwsKXhq5      198.51.100.165  62628   81.30.116.129   53      udp     16438   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       CNECT44FiTEwsKXhq5      198.51.100.165  62628   81.30.116.129   53      udp     16438   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       C8IGxF2IfdASoysMe4      198.51.100.165  33600   185.236.70.184  53      udp     62388   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       C8IGxF2IfdASoysMe4      198.51.100.165  33600   185.236.70.184  53      udp     62388   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
1586579482.075587       C8IGxF2IfdASoysMe4      198.51.100.165  33600   185.236.70.184  53      udp     62388   -       peacecorps.gov  -       -       -       -       0       NOERROR F       F       F >
dns.log
```

Figure 31: show dns.log

- The execution of `cat dns.log | zeek-cut query | sort | uniq -c | sort -n` serves a critical function in aggregating and quantifying unique DNS queries. This pipeline first extracts the query field from the dns.log using zeek-cut, then sorts and counts unique instances, and finally orders them numerically.
  The output of the final command reveals the frequency of individual DNS queries, exemplified by 161 peacecorps.gov. This indicates that the domain peacecorps.gov was queried 161 times, highlighting it as a significant point of interest in the network traffic analyzed.



```
marzan@marzan-VirtualBox:~/Desktop/ShowLog$ cat dns.log | zeek-cut query |sort |uniq -c | sort -n
    161 peacecorps.gov
```

Figure 32: zeek-cut for unique query

7 Detecting DNS Amplification Attacks with Zeek

Zeek stands out as a powerful network security tool, capable of real-time monitoring and analysis of network traffic. It is particularly effective in detecting and mitigating DNS Amplification attacks, a type of Distributed Denial of Service (DDoS) attack that leverages public DNS servers to overwhelm a target with disproportionate DNS response traffic.

7.1 How It Works

The following points outline Zeek's approach to identifying DNS Amplification attacks:

1. **Monitoring DNS Traffic**: Zeek meticulously observes network traffic, focusing on DNS activities to capture both query requests and their responses, thereby monitoring the exchanges between clients and DNS servers.
2. **Analyzing Response Sizes**: Zeek assesses the size of DNS response packets, identifying potential amplification behavior when response sizes significantly exceed those of the query requests.
3. **Establishing Baseline Metrics**: Zeek establishes baseline metrics for standard DNS traffic, including typical sizes for requests and responses and normal query rates, which assists in spotting deviations that might indicate an attack.
4. **Threshold Configuration**: Within Zeek's configuration, customizable thresholds are set for factors such as response size limits and query rate caps. Surpassing these thresholds may indicate a potential DNS Amplification attack.
5. **Pattern Recognition**: Zeek detects patterns that align with DNS Amplification characteristics, like an unusual influx of large response packets directed at a single target or increased queries to open DNS resolvers.
6. **Incident Logging and Alerts**: When Zeek identifies anomalies consistent with DNS Amplification attacks, it logs the incidents and can also be configured to issue real-time alerts, facilitating prompt intervention and attack mitigation.

7.2 DNS Script :

```
@load base/protocols/dns

module Notice;

export {
    redef enum Notice::Type += {DNS_Amplification_Suspected};
}
global dns_query_threshold: count = 20;
global suspicious_dns_activity: table[addr] of count = {};

event dns_request(c: connection, msg: dns_msg, query: string, qtype: count,
    qclass: count) {
    local src = c$id$orig_h;
    if ( src in suspicious_dns_activity ) {
```

```
14         suspicious_dns_activity[src] += 1;
15     } else {
16         suspicious_dns_activity[src] = 1;
17     }
18 
19     if ( suspicious_dns_activity[src] > dns_query_threshold ) {
20         NOTICE([$note=DNS_Amplification_Suspected,
21                 $msg=fmt("High volume of DNS requests from %s, potential DNS
       Amplification.", src),
22                 $src=src,
23                 $conn=c]);
24 
25         delete suspicious_dns_activity[src];
26     }
27 }
28 
29 event dns_query_reply(c: connection, msg: dns_msg, query: string, qtype: count,
       qclass: count) {
30     local src = c$id$orig_h;
31     if ( src in suspicious_dns_activity ) {
32         suspicious_dns_activity[src] += 1;
33     } else {
34         suspicious_dns_activity[src] = 1;
35     }
36 
37     if ( suspicious_dns_activity[src] > dns_query_threshold ) {
38         NOTICE([$note=DNS_Amplification_Suspected,
39                 $msg=fmt("High volume of DNS requests from %s, potential DNS
       Amplification.", src),
40                 $src=src,
41                 $conn=c]);
42 
43         delete suspicious_dns_activity[src];
44     }
45 }
```

7.3 DNS Script Explanation

1. **DNS Script Inclusion** (Line 1): The script initiates by incorporating DNS protocol analysis functionalities through '`@load base/protocols/dns`', enabling inspection of DNS traffic.

2. **Notice Module Extension** (Lines 2-6): The script enhances the '`Notice`' module by appending a new notice type called '`DNS_Amplification_Suspected`', designed for alerting upon detection of suspicious DNS activities.

3. **Global Variables Declaration** (Lines 8-9):

   - '`dns_query_threshold`': Sets a threshold (20 in this context) to delineate the normal count of DNS queries from a single IP before it is deemed suspicious.
   - '`suspicious_dns_activity`': Initializes a table to record the number of DNS queries made by each source IP address, facilitating the tracking of DNS request frequencies.

4. **DNS Request Event Handling** (`dns_request`)(Lines 11-27):

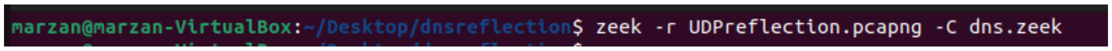
- The script monitors ‘`dns_request`‘ events, triggered by DNS queries.
- Extracts the source IP (‘`src`‘) from each DNS request’s connection details.
- Updates the ‘`suspicious_dns_activity`‘ table with the query count for the source IP, adding the IP with a count of 1 if it’s not already present.
- Generates a ‘`DNS_Amplification_Suspected`‘ notice if the query count for a source IP surpasses the ‘`dns_query_threshold`‘, indicating potential DNS amplification attack behavior.
- Resets the query count for the source IP in the ‘`suspicious_dns_activity`‘ table by removing its entry post-notice generation to avoid repetitive alerts.

5. **DNS Query Reply Event (`dns_query_reply`) (Lines 29-45):**
    - Triggered for each DNS query reply observed in the monitored traffic.
    - Increments a counter in `suspicious_dns_activity` for the source IP address of the query (`c$id$orig_h`).
    - If the count for an IP address exceeds `dns_query_threshold`, a `DNS_Amplification_Suspected` notice is generated, indicating potential amplification activity from that IP address.
    - The entry for the IP address in `suspicious_dns_activity` is deleted after generating the notice, resetting its count.

7.4 Simulation :

Here is the link to PCAP file: malware-traffic-analysis

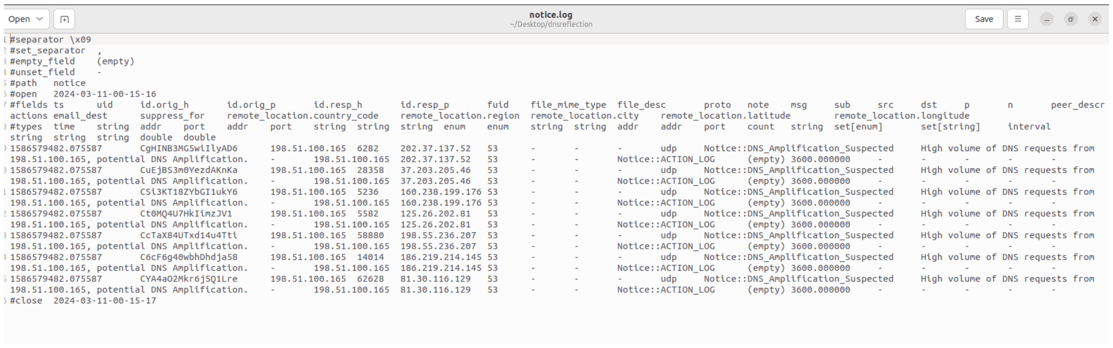
- The process starts by analyzing a PCAP file with dns.zeek, which uses DNS protocol analysis to detect potential DNS Amplification attacks by identifying abnormal DNS query volumes, defining a specific notice type DNS_Amplification_Suspected for such events.



```
marzan@marzan-VirtualBox:~/Desktop/dnsreflection$ zeek -r UDPreflection.pcapng -C dns.zeek
```

Figure 33: Script Execution

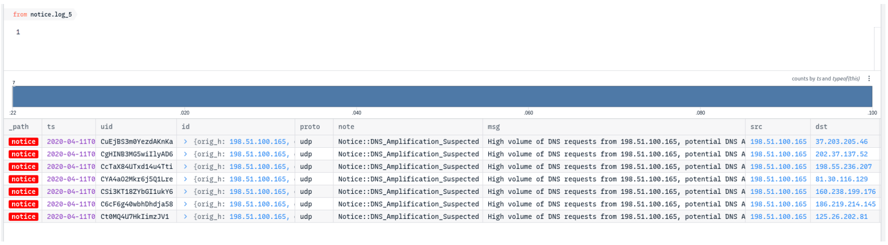
- The notice.log entries indicate the IP 198.51.100.165’s excessive DNS requests, suggesting potential DNS Amplification attacks. These entries document the incident details, including the involved IPs and ports, and mark each event for logging with a one-hour suppression to mitigate repeat alerts.



```
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field    -
#path   notice
#open   2024-03-11-00-15-16
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp_p       fuid    file_mime_type  file_desc       proto   note    msg     sub     src     dst     p       n       peer_descr
actions email_dest      suppress_for    remote_location.country_code    remote_location.region  remote_location.city    remote_location.latitude        remote_location.longitude
#types  time    string  addr    port    addr    port    string  string  string  enum    enum    string  string  addr    addr    port    count   string  set[enum]       set[string]     interval
string  string  string  double  double
1586579482.075587       CgHINB3MG5wiIlyAD6      198.51.100.165  6282    202.37.137.52   53      -       -       -       udp     Notice::DNS_Amplification_Suspected     High volume of DNS requests from
198.51.100.165, potential DNS Amplification.    -       198.51.100.165  202.37.137.52   53      -       -       Notice::ACTION_LOG      (empty) 3600.000000     -       -       -       -       -
1586579482.075587       CuEjBS3m0YezdAKnKa      198.51.100.165  28358   37.203.205.46   53      -       -       -       udp     Notice::DNS_Amplification_Suspected     High volume of DNS requests from
198.51.100.165, potential DNS Amplification.    -       198.51.100.165  37.203.205.46   53      -       -       Notice::ACTION_LOG      (empty) 3600.000000     -       -       -       -       -
1586579482.075587       CSi3KT18ZYbGI1ukY6      198.51.100.165  5236    160.238.199.176 53      -       -       -       udp     Notice::DNS_Amplification_Suspected     High volume of DNS requests from
198.51.100.165, potential DNS Amplification.    -       198.51.100.165  160.238.199.176 53      -       -       Notice::ACTION_LOG      (empty) 3600.000000     -       -       -       -       -
1586579482.075587       Ct0MQ4U7HkIimzJV1       198.51.100.165  5582    125.26.202.81   53      -       -       -       udp     Notice::DNS_Amplification_Suspected     High volume of DNS requests from
198.51.100.165, potential DNS Amplification.    -       198.51.100.165  125.26.202.81   53      -       -       Notice::ACTION_LOG      (empty) 3600.000000     -       -       -       -       -
1586579482.075587       CcTaX84UTxd14u4Tti      198.51.100.165  58880   198.55.236.207  53      -       -       -       udp     Notice::DNS_Amplification_Suspected     High volume of DNS requests from
198.51.100.165, potential DNS Amplification.    -       198.51.100.165  198.55.236.207  53      -       -       Notice::ACTION_LOG      (empty) 3600.000000     -       -       -       -       -
1586579482.075587       C6cF6g40wbhDhdja58      198.51.100.165  14014   186.219.214.145 53      -       -       -       udp     Notice::DNS_Amplification_Suspected     High volume of DNS requests from
198.51.100.165, potential DNS Amplification.    -       198.51.100.165  186.219.214.145 53      -       -       Notice::ACTION_LOG      (empty) 3600.000000     -       -       -       -       -
1586579482.075587       CYA4aO2Mkr6j5Q1Lre      198.51.100.165  62628   81.30.116.129   53      -       -       -       udp     Notice::DNS_Amplification_Suspected     High volume of DNS requests from
198.51.100.165, potential DNS Amplification.    -       198.51.100.165  81.30.116.129   53      -       -       Notice::ACTION_LOG      (empty) 3600.000000     -       -       -       -       -
#close  2024-03-11-00-15-17
```

Figure 34: notice.log

- In BRIM, the notice.log is organized into a formal tabular representation, systematically displaying instances of potential DNS Amplification attacks originating from the IP 198.51.100.165.



_path	ts	uid	id	proto	note	msg	src	dst
notice	2020-04-11T0	CuEjBS3m0YezdAKnKa	> {orig_h: 198.51.100.165,	udp	Notice::DNS_Amplification_Suspected	High volume of DNS requests from 198.51.100.165, potential DNS A	198.51.100.165	37.203.205.46
notice	2020-04-11T0	CgHINB3MG5wiIlyAD6	> {orig_h: 198.51.100.165,	udp	Notice::DNS_Amplification_Suspected	High volume of DNS requests from 198.51.100.165, potential DNS A	198.51.100.165	202.37.137.52
notice	2020-04-11T0	CcTaX84UTxd14u4Tti	> {orig_h: 198.51.100.165,	udp	Notice::DNS_Amplification_Suspected	High volume of DNS requests from 198.51.100.165, potential DNS A	198.51.100.165	198.55.236.207
notice	2020-04-11T0	CYA4aO2Mkr6j5Q1Lre	> {orig_h: 198.51.100.165,	udp	Notice::DNS_Amplification_Suspected	High volume of DNS requests from 198.51.100.165, potential DNS A	198.51.100.165	81.30.116.129
notice	2020-04-11T0	CSi3KT18ZYbGI1ukY6	> {orig_h: 198.51.100.165,	udp	Notice::DNS_Amplification_Suspected	High volume of DNS requests from 198.51.100.165, potential DNS A	198.51.100.165	160.238.199.176
notice	2020-04-11T0	C6cF6g40wbhDhdja58	> {orig_h: 198.51.100.165,	udp	Notice::DNS_Amplification_Suspected	High volume of DNS requests from 198.51.100.165, potential DNS A	198.51.100.165	186.219.214.145
notice	2020-04-11T0	Ct0MQ4U7HkIimzJV1	> {orig_h: 198.51.100.165,	udp	Notice::DNS_Amplification_Suspected	High volume of DNS requests from 198.51.100.165, potential DNS A	198.51.100.165	125.26.202.81

Figure 35: notice.log in BRIM

7.5 Example Script :

```
module DDosAttacks;
@load base/protocols/dns

redef enum Notice::Type += {
    DNSDDoSAmplification
};

function generate_ddos_notice(c: connection, query: string) {
    local query1: string = strip(query);
    if (query1 == "peacecorps.gov" || query1 == "pizzaseo.com") {
        NOTICE([$note = DNSDDoSAmplification,
            $msg = fmt("Possible DNS DDoS Amplification Attack"),
            $conn = c,
            $uid = c$uid
        ]);
    }
}

event dns_request(c: connection, msg: dns_msg, query: string, qtype: count,
    qclass: count, original_query: string) {
```

```
20      generate_ddos_notice(c, query);
21  }
22
23  event dns_query_reply(c: connection, msg: dns_msg, query: string, qtype: count,
        qclass: count, original_query: string) {
24      generate_ddos_notice(c, query);
25  }
```

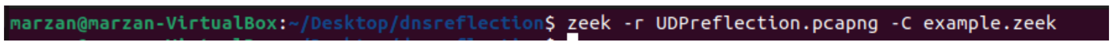
7.6 Example Script Explanation

1. **Module Declaration and DNS Script Loading** (Lines 1-2): The script commences with the declaration of a module titled ‘DDosAttacks‘. It proceeds to integrate DNS protocol analysis functionalities from the Zeek standard library through the command ‘`@load base/protocols/dns`‘.
2. **Enumeration Extension for Notice** (Lines 4-5): The script enhances the ‘`Notice::Type`‘ enumeration by appending a novel type named ‘DNSDDoSAmplification‘. This addition is aimed at facilitating the signaling of potential DNS DDoS amplification attacks.
3. **DDoS Notice Generation Function** (Lines 8-17): A function named ‘`generate_ddos_notice`‘ is introduced. It is designed to examine network connections (denoted as ‘c‘) and DNS queries (denoted as ‘query‘). The function scrutinizes the DNS query after removing any whitespace and, if it matches specific domains (‘peacecorps.gov‘ or ‘pizzaseo.com‘), it issues a notice of type ‘DNSDDoSAmplification‘.
4. **DNS Request Event Handling** (Lines 19-21): The script configures an event handler named ‘`dns_request`‘. This handler is triggered for every DNS request and it invokes the ‘`generate_ddos_notice`‘ function. This mechanism is employed to assess each DNS inquiry for indications of DNS DDoS amplification attacks.
5. **DNS Query Reply Event Handling** (Lines 23-25): Similarly, the script sets up another event handler named ‘`dns_query_reply`‘. This handler also calls the ‘`generate_ddos_notice`‘ function for DNS responses, thereby extending the surveillance to include reply messages which might be pertinent in detecting attack patterns.

7.7 Simulation :

Here is the link to PCAP file: malware-traffic-analysis

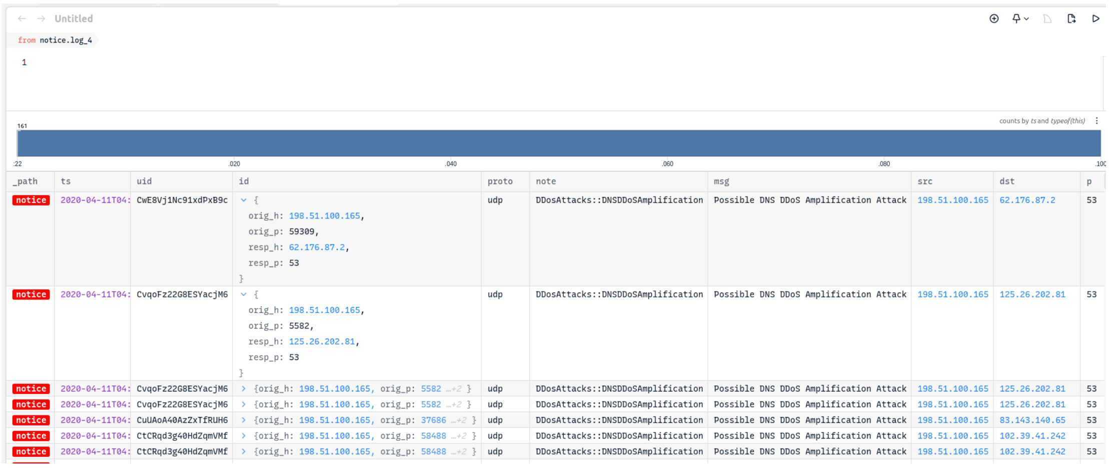
- The process begins with the execution of a PCAP file through the example.zeek script, part of the DDosAttacks module, which loads DNS protocol analyses. The script is designed to identify potential DNS DDoS Amplification attacks by inspecting DNS queries. If a query matches specific target domains, it triggers a DNSDDoSAmplification notice.



```
marzan@marzan-VirtualBox:~/Desktop/dnsreflection$ zeek -r UDPreflection.pcapng -C example.zeek
```

Figure 36: Script Execution

- The resulting notice.log entries, visualized in BRIM in a tabular format, detail instances where the script detected potential DNS DDoS Amplification attacks. Each entry includes the timestamp, source and destination IP addresses and ports, protocol used (UDP), and a message indicating a ”Possible DNS DDoS Amplification Attack”. The action logged is Notice::ACTION_LOG, and a suppression period of 3600 seconds is set to prevent immediate duplicate alerts.



```
from notice.log_4
1
```

_path	ts	uid	id	proto	note	msg	src	dst	p
notice	2020-04-11T04:	CwE8Vj1Nc91xdPxB9c	{ orig_h: 198.51.100.165, orig_p: 59309, resp_h: 62.176.87.2, resp_p: 53 }	udp	DDosAttacks::DNSDDoSAmplification	Possible DNS DDoS Amplification Attack	198.51.100.165	62.176.87.2	53
notice	2020-04-11T04:	CvqoFz22G8ESYacjM6	{ orig_h: 198.51.100.165, orig_p: 5582, resp_h: 125.26.202.81, resp_p: 53 }	udp	DDosAttacks::DNSDDoSAmplification	Possible DNS DDoS Amplification Attack	198.51.100.165	125.26.202.81	53
notice	2020-04-11T04:	CvqoFz22G8ESYacjM6	{orig_h: 198.51.100.165, orig_p: 5582 …+2 }	udp	DDosAttacks::DNSDDoSAmplification	Possible DNS DDoS Amplification Attack	198.51.100.165	125.26.202.81	53
notice	2020-04-11T04:	CvqoFz22G8ESYacjM6	{orig_h: 198.51.100.165, orig_p: 5582 …+2 }	udp	DDosAttacks::DNSDDoSAmplification	Possible DNS DDoS Amplification Attack	198.51.100.165	125.26.202.81	53
notice	2020-04-11T04:	CuUAoA40AzZxTfRUH6	{orig_h: 198.51.100.165, orig_p: 37686 …+2 }	udp	DDosAttacks::DNSDDoSAmplification	Possible DNS DDoS Amplification Attack	198.51.100.165	83.143.140.65	53
notice	2020-04-11T04:	CtCRqd3g40HdZqmVMf	{orig_h: 198.51.100.165, orig_p: 58488 …+2 }	udp	DDosAttacks::DNSDDoSAmplification	Possible DNS DDoS Amplification Attack	198.51.100.165	102.39.41.242	53
notice	2020-04-11T04:	CtCRqd3g40HdZqmVMf	{orig_h: 198.51.100.165, orig_p: 58488 …+2 }	udp	DDosAttacks::DNSDDoSAmplification	Possible DNS DDoS Amplification Attack	198.51.100.165	102.39.41.242	53

Figure 37: notice.log Analysis in BRIM

8 Detecting SSH Brute Forcing with Zeek

Zeek is a powerful network traffic analyzer that plays a crucial role in cybersecurity defenses by monitoring and analyzing network traffic in real time. It's particularly adept at identifying and mitigating security threats, such as SSH brute force attacks. An SSH brute force attack involves an attacker attempting numerous passwords or passphrase combinations to gain unauthorized access to a system. Zeek's approach to detecting such attacks hinges on its ability to scrutinize the intricacies of SSH traffic, focusing on the authentication process.

8.1 How It Works

Here's a concise breakdown of how Zeek detects SSH brute forcing:

1. **Monitoring SSH Traffic:** Zeek listens to network traffic, focusing on the SSH protocol to capture the initial handshake, authentication attempts, and the session's lifecycle.

2. **Analyzing Authentication Responses:** It examines the size of the server's response packets post-authentication attempts. A response size around 5k bytes often signifies a successful login, which Zeek uses as a heuristic measure.

3. **Setting Detection Thresholds:** Parameters like "password_guesses_limit" and "guessing_timeout" within Zeek's SSH brute forcing script help define the acceptable number of failed login attempts within a certain timeframe, beyond which an attack is presumed.

4. **Identifying Brute Force Patterns:** Zeek flags a sequence of failed logins (indicated by "auth_success" set to "F") exceeding the predefined threshold within the set timeframe as a brute force attempt.

5. **Logging and Alerting:** Following the detection of a brute force attack, Zeek logs the event and can generate alerts, facilitating prompt investigation and response.

8.2 Detect-bruteforcing.zeek Script:

```
##! Detect hosts which are doing password guessing attacks and/or password
##! bruteforcing over SSH.

@load base/protocols/ssh
@load base/frameworks/sumstats
@load base/frameworks/notice
@load base/frameworks/intel
module SSH;
export {
  redef enum Notice::Type += {
    ## Indicates that a host has been identified as crossing the
    ## :zeek:id:`SSH::password_guesses_limit` threshold with
    ## failed logins.
    Password_Guessing,
    ## Indicates that a host previously identified as a "password
    ## guesser" has now had a successful login
    ## attempt. This is not currently implemented.
    Login_By_Password_Guesser,
  };

  redef enum Intel::Where += {
    ## An indicator of the login for the intel framework.
    SSH::SUCCESSFUL_LOGIN,
  };

  ## The number of failed SSH connections before a host is designated as
  ## guessing passwords.
  const password_guesses_limit: double = 30 &redef;

  ## The amount of time to remember presumed non-successful logins to
  ## build a model of a password guesser.
  const guessing_timeout = 30 mins &redef;

  ## This value can be used to exclude hosts or entire networks from being
  ## tracked as potential "guessers". The index represents
  ## client subnets and the yield value represents server subnets.
  const ignore_guessers: table[subnet] of subnet &redef;
}

event zeek_init()
  {
  local r1: SumStats::Reducer = [$stream="ssh.login.failure", $apply=set(SumStats::SUM, SumStats::SAMPLE), $num_samples=5];
  SumStats::create([$name="detect-ssh-bruteforcing",
                    $epoch=guessing_timeout,
                    $reducers=set(r1),
                    $threshold_val(key: SumStats::Key, result: SumStats::Result) =
                      {
                      return result["ssh.login.failure"]$sum;
                      },
                    $threshold=password_guesses_limit,
                    $threshold_crossed(key: SumStats::Key, result: SumStats::Result) =
                      {
                      local r = result["ssh.login.failure"];
                      local sub_msg = fmt("Sampled servers: ");
                      local samples = r$samples;
                      for ( i in samples )
                        {
                        if ( samples[i]?$str )
                          sub_msg = fmt("%s%s %s", sub_msg, i==0 ? "":",", samples[i]$str);
                        }
                      # Generate the notice.
                      NOTICE([$note=Password_Guessing,
                              $msg=fmt("%s appears to be guessing SSH passwords (seen in %d connections).", key$host, r$num),
                              $sub=sub_msg,
                              $src=key$host,
                              $identifier=cat(key$host)]);
                      }]);
  }

event ssh_auth_successful(c: connection, auth_method_none: bool)
  {
  local id = c$id;
```

```
73
74   Intel::seen([$host=id$orig_h,
75                $conn=c,
76                $where=SSH::SUCCESSFUL_LOGIN]);
77   }
78
79 event ssh_auth_failed(c: connection)
80   {
81   local id = c$id;
82
83   # Add data to the FAILED_LOGIN metric unless this connection should
84   # be ignored.
85   if ( ! (id$orig_h in ignore_guessers &&
86           id$resp_h in ignore_guessers[id$orig_h]) )
87     SumStats::observe("ssh.login.failure", [$host=id$orig_h], [$str=cat(id$resp_h
     )]);
88   }
```

Listing 6: detect-bruteforcing.zeek

8.3 Detect-bruteforcing.zeek Script Explanation

1. **Notice Types Extension** (Lines 10-19): Extends Zeek's notice framework with 'Password_Guessing' and 'Login_By_Password_Guesser' for flagging SSH brute force activities and successful logins by flagged hosts.
2. **Intel Framework Extension** (Lines 21-24): Adds 'SSH::SUCCESSFUL_LOGIN' to the Intel framework, tracking successful SSH logins from hosts involved in brute force activities.
3. **Configuration Constants** (Lines 26-36): Sets 'password_guesses_limit' to 30 and 'guessing_timeout' to 30 minutes, defining the threshold for failed login attempts and the time window for their evaluation.
4. **SumStats Framework Integration** (Lines 38-68): Utilizes 'SumStats' for aggregating failed SSH login attempts, employing a reducer with '"ssh.login.failure"' and configuring aggregation functions with a sample size of 5. Threshold evaluation and crossing logic are based on the sum of failed login attempts.
5. **Successful Login Intel Reporting** (Lines 70-77): Logs successful SSH authentications through the 'ssh_auth_successful' event using 'Intel::seen', marking with 'SSH::SUCCESSFUL_LOGIN'.
6. **Failed Login Data Collection** (Lines 79-87): Captures and logs each failed SSH login attempt in the 'ssh_auth_failed' event, excluding those from 'ignore_guessers', contributing to brute force detection analysis.

8.4 SSHAttack Script :

```
module SShAttacks;

@load protocols/ssh/detect-bruteforcing
redef SSH::password_guesses_limit=3;

```

```
 6 hook Notice::policy(n: Notice::Info) {
 7   if ( n$note == SSH::Password_Guessing ) {
 8     add n$actions[Notice::ACTION_LOG];
 9   }
10 }
```

Listing 7: sshAttack.zeek

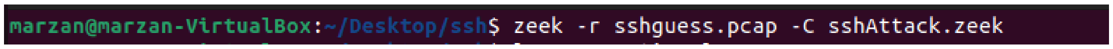
8.5 SSHAttack Script Explanation

1. **Module Declaration** (Line 1): Declares a new module `SShAttacks` for organizing script functionalities. The script's module naming might contain a capitalization typo in "SSH".
2. **Load Statement** (Line 3): Instructs Zeek to load the `detect-bruteforcing` script from the SSH protocol analysis suite, which contains logic for identifying potential brute force attempts.
3. **Variable Redefinition** (Line 4): Sets the brute force detection threshold at three failed attempts by redefining `SSH::password_guesses_limit` to 3.
4. **Notice Framework Hook** (Line 6): Implements a hook into Zeek's Notice framework with `hook Notice::policy(n: Notice::Info) { ...}`, allowing customized handling of notable events, particularly SSH password guessing.
5. **Notice Handling** (Lines 7-8): Within the hook, the script checks for notices related to SSH password guessing (`n$note == SSH::Password_Guessing`) and logs these events (`add n$actions[Notice::ACTION_LOG];`) for further analysis.

8.6 Simulation :

Here is the link to PCAP file: malware-traffic-analysis

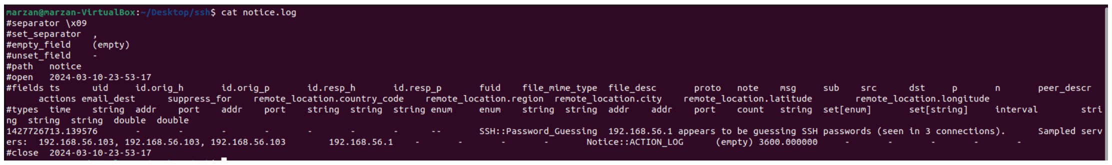
- The PCAP file is executed within Zeek using the 'ssAttack.zeek' script for analysis.



```
marzan@marzan-VirtualBox:~/Desktop/ssh$ zeek -r sshguess.pcap -C sshAttack.zeek
```

Figure 38: Executing PCAP file in Zeek with 'ssAttack.zeek' script

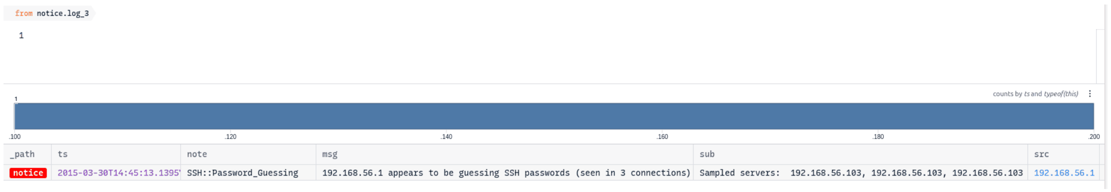
- This notice log entry from Zeek indicates that the IP address 192.168.56.1 was detected attempting SSH password guessing, observed over three connections to 192.168.56.103. The activity triggered a SSH::Password_Guessing alert, and the system took the action to log the event (Notice::ACTION_LOG). An alert suppression period of one hour (3600.000000 seconds) was set following this detection.



```
marzan@marzan-VirtualBox:~/Desktop/ssh$ cat notice.log
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field    -
#path   notice
#open   2024-03-10-23-53-17
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp_p       fuid    file_mime_type  file_desc       proto   note    msg     sub     src     dst     p       n       peer_descr
      actions email_dest      suppress_for    remote_location.country_code    remote_location.region  remote_location.city    remote_location.latitude        remote_location.longitude
#types  time    string  addr    port    addr    port    string  string  string enum     enum    string  string  addr    addr    port    count   string  set[enum]       set[string]     interval        stri
ng  string  string  double  double
1427726713.139576       -       -       -       -       -       -       -       --      SSH::Password_Guessing  192.168.56.1 appears to be guessing SSH passwords (seen in 3 connections).      Sampled serv
ers:  192.168.56.103, 192.168.56.103, 192.168.56.103        192.168.56.1    -       -       -       -       Notice::ACTION_LOG      (empty) 3600.000000     -       -       -       -       -
#close  2024-03-10-23-53-17
```

Figure 39: notice.log

- In BRIM, the notice log entry is presented in a structured tabular format, detailing the detection of an SSH password guessing attempt.



_path	ts	note	msg	sub	src
notice	2015-03-30T14:45:13.1395	SSH::Password_Guessing	192.168.56.1 appears to be guessing SSH passwords (seen in 3 connections)	Sampled servers: 192.168.56.103, 192.168.56.103, 192.168.56.103	192.168.56.1

Figure 40: notice.log in BRIM

9 JA3 Fingerprinting and Zeek Integration

The advent of encryption in network communications, while enhancing privacy and security, has also enabled malicious actors to conceal their activities. This report details a solution to this challenge through the integration of JA3 fingerprinting with Zeek, aimed at identifying and alerting on Trickbot malware communications.

9.1 How It Works

The integration of JA3 fingerprinting with Zeek to detect Trickbot malware communications works through a series of coordinated steps, leveraging the unique capabilities of both technologies to identify and alert on potential security threats in encrypted network traffic. Here's a summary of how it works:

- **JA3 Fingerprinting:** JA3 generates unique fingerprints for SSL/TLS clients based on TLS handshake details, creating a consistent hash value for client identification.
- **Custom Zeek Script:** A Zeek script is developed to capture TLS handshakes, apply JA3 hashing, and generate fingerprints for network traffic analysis.
- **Trickbot Hash Matching:** The script compares generated JA3 hashes against a database of known Trickbot-associated hashes to identify potential malware communications.
- **Detection and Alerting:** When a JA3 hash matches a known Trickbot hash, Zeek triggers an alert, signaling a potential security threat.
- **Logging and Analysis:** Detected events and their details are logged for further investigation, providing valuable data for incident response and threat intelligence.
- **Enhanced Security Posture:** Integrating JA3 with Zeek improves the detection of encrypted malware communications, enhancing the network's security monitoring capabilities.

9.2 JA3 Script :

The following listing presents the Zeek script used for JA3 fingerprinting.

```
module JA3;

export {
redef enum Log::ID += { LOG };
}

type TLSFPStorage: record {
       client_version:  count &default=0 &log;
       client_ciphers:  string &default="" &log;
       extensions:      string &default="" &log;
       e_curves:        string &default="" &log;
       ec_point_fmt:    string &default="" &log;
};
redef record connection += {
       tlsfp: TLSFPStorage &optional;
```

```
};

redef record SSL::Info += {
  ja3:             string &optional &log;
# LOG FIELD VALUES ##
#   ja3_version:  string &optional &log;
#   ja3_ciphers:  string &optional &log;
#   ja3_extensions: string &optional &log;
#   ja3_ec:         string &optional &log;
#   ja3_ec_fmt:     string &optional &log;
};

# Google. https://tools.ietf.org/html/draft-davidben-tls-grease-01

const grease: set[int] = {
    2570, 6682, 10794, 14906, 19018, 23130, 27242, 31354, 35466, 39578, 43690,
    47802, 51914, 56026, 60138, 64250
};

const sep = "-";
event zeek_init() {

    Log::create_stream(JA3::LOG,[$columns=TLSFPStorage, $path="tlsfp"]);
}

event ssl_extension(c: connection, is_orig: bool, code: count, val: string)
{
    if ( is_orig == T ) {
        if ( code in grease ) {
            return;
        }
        if ( ! c?$tlsfp ){
            c$tlsfp=TLSFPStorage();
        }
        if ( c$tlsfp$extensions == "" ) {
            c$tlsfp$extensions = cat(code);
        }
        else {
            c$tlsfp$extensions = string_cat(c$tlsfp$extensions, sep,cat(code));
        }
    }
}

event ssl_extension_ec_point_formats(c: connection, is_orig: bool, point_formats:
     index_vec)
{
    if ( is_orig == T ) {
        if ( !c?$tlsfp )
            c$tlsfp=TLSFPStorage();
        for ( i in point_formats ) {
            if ( point_formats[i] in grease ) {
            next;
            }
            if ( c$tlsfp$ec_point_fmt == "" ) {
            c$tlsfp$ec_point_fmt += cat(point_formats[i]);
            }
            else {
            c$tlsfp$ec_point_fmt += string_cat(sep,cat(point_formats[i]));
            }
        }
    }
}

event ssl_extension_elliptic_curves(c: connection, is_orig: bool, curves: index_vec)
{
    if ( !c?$tlsfp )
    c$tlsfp=TLSFPStorage();
    if ( is_orig == T  ) {
        for ( i in curves ) {
            if ( curves[i] in grease ) {
            next;
            }
            if ( c$tlsfp$e_curves == "" ) {
             c$tlsfp$e_curves += cat(curves[i]);
            }
            else {
                c$tlsfp$e_curves += string_cat(sep,cat(curves[i]));
            }
        }
   }
}

@if ( ( Version::number >= 20600 ) || ( Version::number == 20500 && Version::info$commit >= 944 ) )
event ssl_client_hello(c: connection, version: count, record_version: count, possible_ts: time, client_random: string, session_id: string, ciphers: index_vec, comp_methods: index_vec) &priority=1
@else
event ssl_client_hello(c: connection, version: count, possible_ts: time, client_random: string, session_id: string, ciphers: index_vec) &priority=1
@endif
{
    if ( !c?$tlsfp )

    c$tlsfp=TLSFPStorage();

    c$tlsfp$client_version = version;

    for ( i in ciphers ) {

        if ( ciphers[i] in grease ) {
            next;
        }
        if ( c$tlsfp$client_ciphers == "" ) {
            c$tlsfp$client_ciphers += cat(ciphers[i]);
        }
        else {
            c$tlsfp$client_ciphers += string_cat(sep,cat(ciphers[i]));
        }
    }
    local sep2 = ",";
    local ja3_string = string_cat(cat(c$tlsfp$client_version),sep2,c$tlsfp$client_ciphers,sep2,c$tlsfp$extensions,sep2,c$tlsfp$e_curves,sep2,c$tlsfp$ec_point_fmt);
    local tlsfp_1 = md5_hash(ja3_string);
    c$ssl$ja3 = tlsfp_1;

# LOG FIELD VALUES ##
#c$ssl$ja3_version = cat(c$tlsfp$client_version);
#c$ssl$ja3_ciphers = c$tlsfp$client_ciphers;
#c$ssl$ja3_extensions = c$tlsfp$extensions;
#c$ssl$ja3_ec = c$tlsfp$e_curves;
#c$ssl$ja3_ec_fmt = c$tlsfp$ec_point_fmt;

#
# FOR DEBUGGING ##
#print "JA3: "+tlsfp_1+" Fingerprint String: "+ja3_string;

}
```

Listing 8: JA3 Fingerprinting Zeek Script

9.3 JA3 Script Explanation

1. **Module Initialization:** The script declares a 'JA3' module and extends Zeek's logging capabilities to include JA3 fingerprints.
2. **Data Structures:**
   (a) **TLSFPStorage:** A record that stores TLS handshake components such as the TLS version, cipher suites, SSL extensions, elliptic curves, and EC point formats.
   (b) The Zeek 'connection' record is extended to include a 'tlsfp' field for storing TLS fingerprint data.
3. **SSL::Info Extension:** The 'SSL::Info' record is augmented with a 'ja3' field to hold the MD5 hash of the JA3 fingerprint.
4. **GREASE Handling:** The script accounts for GREASE values in the TLS handshake, ensuring they are ignored to maintain fingerprint integrity.
5. **Event Handlers:**
   (a) **ssl_extension:** Captures SSL extensions, excluding GREASE values, for JA3 string construction.
   (b) **ssl_extension_ec_point_formats** and **ssl_extension_elliptic_curves:** Manage elliptic curve information critical for JA3 fingerprinting.
   (c) **ssl_client_hello:** Processes the client's initial handshake message, extracting key information to assemble the JA3 string, which is then hashed to produce the JA3 fingerprint.
6. **JA3 Hash Generation:** The JA3 string, a concatenation of selected handshake properties, is hashed using MD5 to generate the fingerprint, uniquely identifying SSL/TLS clients.

9.4 Notice Script:

```
@load base/frameworks/notice
module JA3;

redef enum Notice::Type += {
    TrickBotDetection
};

export {
    global target_ja3_hash = "6734f37431670b3ab4292b8f60f29984";
}


event ssl_client_hello(c: connection, version: count, record_version: count,
    possible_ts: time, client_random: string, session_id: string, ciphers:
    index_vec, comp_methods: index_vec) &priority=1
{
    if (c$ssl$ja3 == target_ja3_hash) {
        local msg = fmt("Detected target JA3 hash (%s) for connection: %s (Orig:
    %s, Resp: %s)", target_ja3_hash, c$id, c$id$orig_h, c$id$resp_h);
        NOTICE([$note=TrickBotDetection, $msg=msg, $src=c$id$orig_h, $dst=
    c$id$resp_h]);
    }
}
```

Listing 9: JA3 notice.zeek

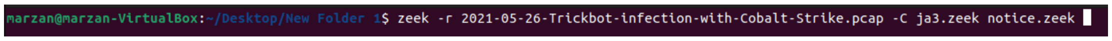
9.5 Notice Script Explanation

1. **Notice Framework Integration**: Utilizes Zeek's `base/frameworks/notice` module for alert generation.
2. **JA3 Module Declaration**: Establishes a JA3 module to encapsulate JA3 hashing logic.
3. **Notice Type Extension**: Introduces `TrickBotDetection` as a new notice type for specific categorization of alerts.
4. **Target JA3 Hash Definition**: Sets `target_ja3_hash` with the JA3 hash signature of Trickbot malware for comparison.
5. **SSL Client Hello Analysis**:
   (a) Extracts parameters from `ssl_client_hello` messages necessary for JA3 hash computation.
   (b) Computes the JA3 hash based on these parameters.
6. **Detection and Alerting**:
   (a) Compares the computed JA3 hash with the `target_ja3_hash`.
   (b) Generates a `TrickBotDetection` notice upon hash match, indicating potential Trickbot activity.

9.6 Simulation

Here is the link to PCAP file: malware-traffic-analysis

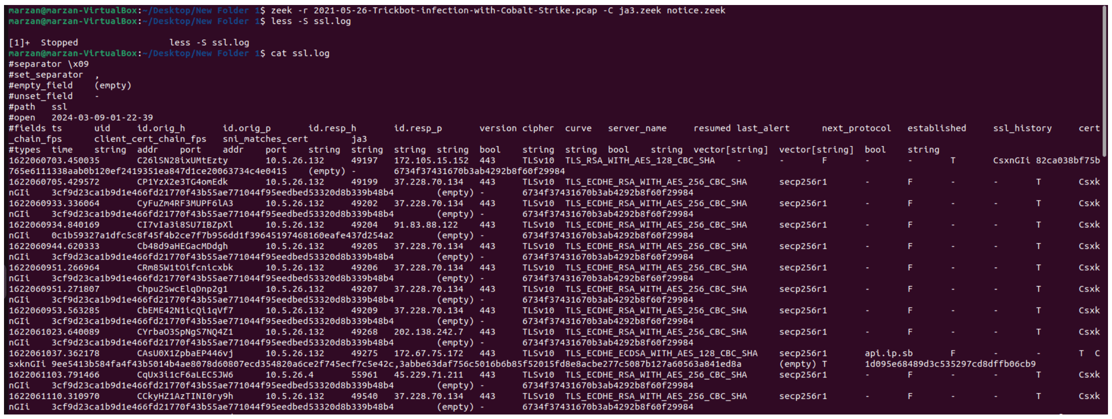
- The PCAP file is processed using ja3.zeek and notice.zeek scripts in Zeek. The ja3.zeek script generates JA3 hashes for TLS handshakes, while notice.zeek identifies hashes matching Trickbot signatures, creating alerts in notice.log.



```
marzan@marzan-VirtualBox:~/Desktop/New Folder 1$ zeek -r 2021-05-26-Trickbot-infection-with-Cobalt-Strike.pcap -C ja3.zeek notice.zeek
```

Figure 41: run pcap file with ja3.zeek and notice.zeek script

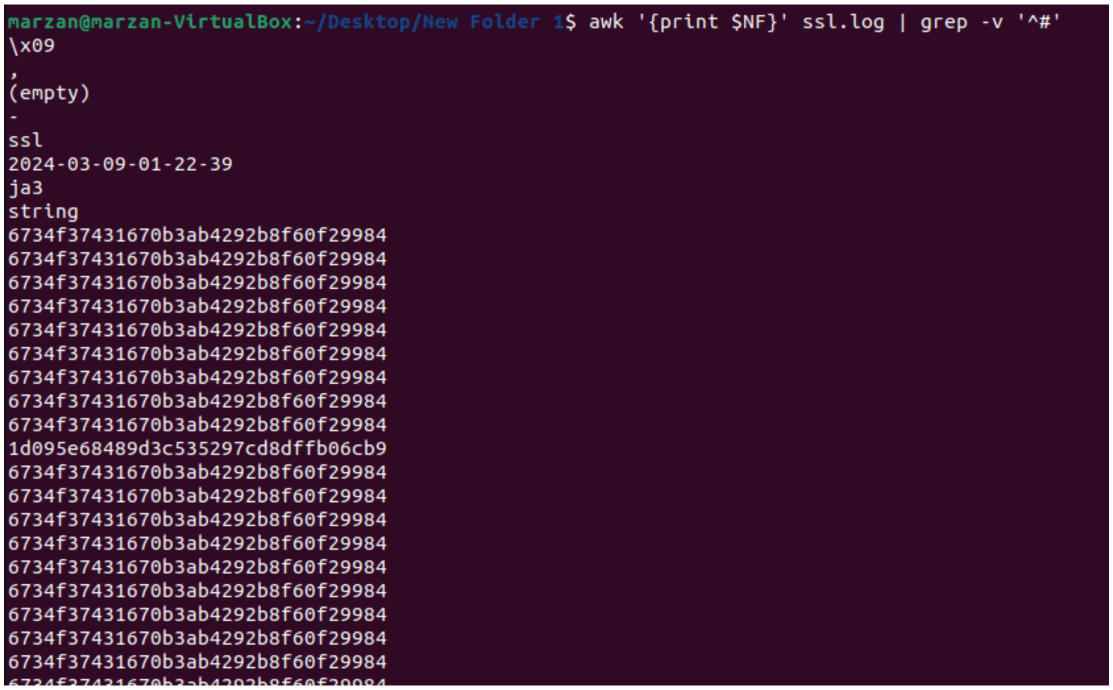
- The ssl.log file is examined to review detailed TLS handshake records and associated JA3 hashes generated during the simulation.



```
marzan@marzan-VirtualBox:~/Desktop/New Folder 1$ zeek -r 2021-05-26-Trickbot-infection-with-Cobalt-Strike.pcap -C ja3.zeek notice.zeek
marzan@marzan-VirtualBox:~/Desktop/New Folder 1$ less -S ssl.log

[1]+  Stopped                 less -S ssl.log
marzan@marzan-VirtualBox:~/Desktop/New Folder 1$ cat ssl.log
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field    -
#path   ssl
#open   2024-03-09-01-22-39
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp_p       version cipher  curve   server_name     resumed last_alert      next_protocol   established     ssl_history     cert
_chain_fps      client_cert_chain_fps   sni_matches_cert        ja3
#types  time    string  addr    port    addr    port    string  string  string  string  bool    string  string  bool    string  vector[string]  vector[string]  bool    string
1622060703.450035       C26lSN28ixUMtEzty       10.5.26.132     49197   172.105.15.152  443     TLSv10  TLS_RSA_WITH_AES_128_CBC_SHA    -       -       F       -       -       T       CsxnGIi 82ca038bf75b
765e6111338aab0b120ef2419351ea847d1ce20063734c4e0415    (empty) -       6734f37431670b3ab4292b8f60f29984
1622060705.429572       CP1YzX2e3TG4omEdk       10.5.26.132     49199   37.228.70.134   443     TLSv10  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA      secp256r1       -       F       -       -       T       Csxk
nGIi    3cf9d23ca1b9d1e466fd21770f43b55ae771044f95eedbed53320d8b339b48b4        (empty) -       6734f37431670b3ab4292b8f60f29984
1622060933.336064       CyFuZm4RF3MUPF6lA3      10.5.26.132     49202   37.228.70.134   443     TLSv10  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA      secp256r1       -       F       -       -       T       Csxk
nGIi    3cf9d23ca1b9d1e466fd21770f43b55ae771044f95eedbed53320d8b339b48b4        (empty) -       6734f37431670b3ab4292b8f60f29984
1622060934.840169       CI7vIa3i8SU7IBZpXl      10.5.26.132     49204   91.83.88.122    443     TLSv10  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA      secp256r1       -       F       -       -       T       Csxk
nGIi    0c1b59327a1dfc5c8f45f4b2ce7f7b956dd1f39645197468160eafe437d254a2        (empty) -       6734f37431670b3ab4292b8f60f29984
1622060944.620333       Cb48d9aHEGacMDdgh       10.5.26.132     49205   37.228.70.134   443     TLSv10  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA      secp256r1       -       F       -       -       T       Csxk
nGIi    3cf9d23ca1b9d1e466fd21770f43b55ae771044f95eedbed53320d8b339b48b4        (empty) -       6734f37431670b3ab4292b8f60f29984
1622060951.266964       CRm85W1tOifcnicxbk      10.5.26.132     49206   37.228.70.134   443     TLSv10  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA      secp256r1       -       F       -       -       T       Csxk
nGIi    3cf9d23ca1b9d1e466fd21770f43b55ae771044f95eedbed53320d8b339b48b4        (empty) -       6734f37431670b3ab4292b8f60f29984
1622060951.271807       Chpu2SwcElqDnp2g1       10.5.26.132     49207   37.228.70.134   443     TLSv10  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA      secp256r1       -       F       -       -       T       Csxk
nGIi    3cf9d23ca1b9d1e466fd21770f43b55ae771044f95eedbed53320d8b339b48b4        (empty) -       6734f37431670b3ab4292b8f60f29984
1622060953.563285       CbEME42N1icQi1qVf7      10.5.26.132     49209   37.228.70.134   443     TLSv10  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA      secp256r1       -       F       -       -       T       Csxk
nGIi    3cf9d23ca1b9d1e466fd21770f43b55ae771044f95eedbed53320d8b339b48b4        (empty) -       6734f37431670b3ab4292b8f60f29984
1622061023.640089       CYrbaO3SpNgS7NQ4Z1      10.5.26.132     49268   202.138.242.7   443     TLSv10  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA      secp256r1       -       F       -       -       T       Csxk
nGIi    3cf9d23ca1b9d1e466fd21770f43b55ae771044f95eedbed53320d8b339b48b4        (empty) -       6734f37431670b3ab4292b8f60f29984
1622061037.362178       CAsU0X1ZpbaEP446vj      10.5.26.132     49275   172.67.75.172   443     TLSv10  TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA    secp256r1       api.ip.sb       F       -       -       T  C
sxknGIi 9ee5413b584fa4f43b5014b4ae8078d60807ecd354820a6ce2f745ecf7c5e42c,3abbe63daf756c5016b6b85f52015fd8e8acbe277c5087b127a60563a841ed8a        (empty) T       1d095e68489d3c535297cd8dffb06cb9
1622061103.791466       CqUx3i1cF6aLEC5JW6      10.5.26.4       55961   45.229.71.211   443     TLSv10  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA      secp256r1       -       F       -       -       T       Csxk
nGIi    3cf9d23ca1b9d1e466fd21770f43b55ae771044f95eedbed53320d8b339b48b4        (empty) -       6734f37431670b3ab4292b8f60f29984
1622061110.310970       CCkyHZ1AzTINI0ry9h      10.5.26.132     49540   37.228.70.134   443     TLSv10  TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA      secp256r1       -       F       -       -       T       Csxk
nGIi    3cf9d23ca1b9d1e466fd21770f43b55ae771044f95eedbed53320d8b339b48b4        (empty) -       6734f37431670b3ab4292b8f60f29984
```

Figure 42: ssl.log

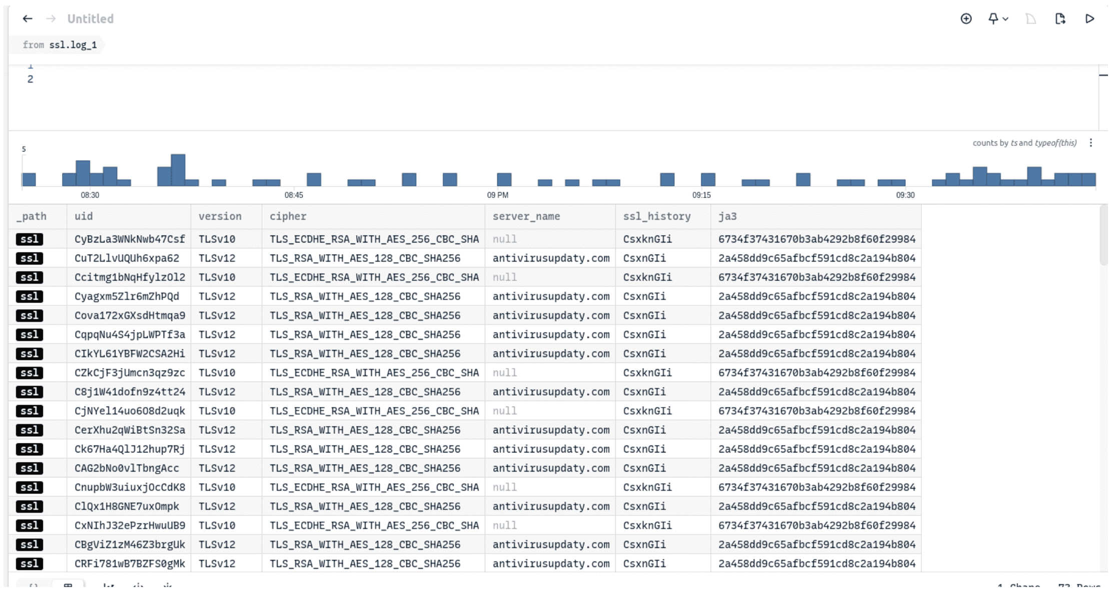
- Specific JA3 hash fields are extracted from the ssl.log for focused analysis.



```
marzan@marzan-VirtualBox:~/Desktop/New Folder 1$ awk '{print $NF}' ssl.log | grep -v '^#'
\x09
,
(empty)
-
ssl
2024-03-09-01-22-39
ja3
string
6734f37431670b3ab4292b8f60f29984
6734f37431670b3ab4292b8f60f29984
6734f37431670b3ab4292b8f60f29984
6734f37431670b3ab4292b8f60f29984
6734f37431670b3ab4292b8f60f29984
6734f37431670b3ab4292b8f60f29984
6734f37431670b3ab4292b8f60f29984
6734f37431670b3ab4292b8f60f29984
6734f37431670b3ab4292b8f60f29984
1d095e68489d3c535297cd8dffb06cb9
6734f37431670b3ab4292b8f60f29984
6734f37431670b3ab4292b8f60f29984
6734f37431670b3ab4292b8f60f29984
6734f37431670b3ab4292b8f60f29984
6734f37431670b3ab4292b8f60f29984
6734f37431670b3ab4292b8f60f29984
6734f37431670b3ab4292b8f60f29984
6734f37431670b3ab4292b8f60f29984
6734f37431670b3ab4292b8f60f29984
```

Figure 43: ja3 entries

- The ssl.log entries are visualized in BRIM, offering a tabular representation of SSL transactions and JA3 hashes



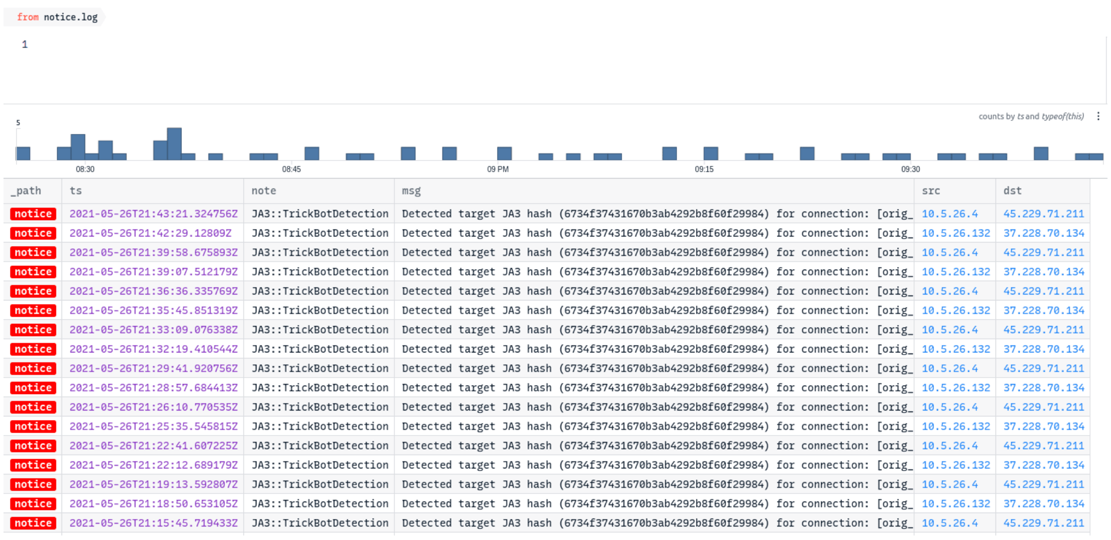
```
from ssl.log_1
```

_path	uid	version	cipher	server_name	ssl_history	ja3
ssl	CyBzLa3WNkNwb47Csf	TLSv10	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	null	CsxknGIi	6734f37431670b3ab4292b8f60f29984
ssl	CuT2LlvUQUh6xpa62	TLSv12	TLS_RSA_WITH_AES_128_CBC_SHA256	antivirusupdaty.com	CsxnGIi	2a458dd9c65afbcf591cd8c2a194b804
ssl	Ccitmg1bNqHfylzOl2	TLSv10	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	null	CsxknGIi	6734f37431670b3ab4292b8f60f29984
ssl	Cyagxm5Zlr6mZhPQd	TLSv12	TLS_RSA_WITH_AES_128_CBC_SHA256	antivirusupdaty.com	CsxnGIi	2a458dd9c65afbcf591cd8c2a194b804
ssl	Cova172xGXsdHtmqa9	TLSv12	TLS_RSA_WITH_AES_128_CBC_SHA256	antivirusupdaty.com	CsxnGIi	2a458dd9c65afbcf591cd8c2a194b804
ssl	CqpqNu4S4jpLWPTf3a	TLSv12	TLS_RSA_WITH_AES_128_CBC_SHA256	antivirusupdaty.com	CsxnGIi	2a458dd9c65afbcf591cd8c2a194b804
ssl	CIkYL61YBFW2CSA2Hi	TLSv12	TLS_RSA_WITH_AES_128_CBC_SHA256	antivirusupdaty.com	CsxnGIi	2a458dd9c65afbcf591cd8c2a194b804
ssl	CZkCjF3jUmcn3qz9zc	TLSv10	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	null	CsxknGIi	6734f37431670b3ab4292b8f60f29984
ssl	C8j1W41dofn9z4tt24	TLSv12	TLS_RSA_WITH_AES_128_CBC_SHA256	antivirusupdaty.com	CsxnGIi	2a458dd9c65afbcf591cd8c2a194b804
ssl	CjNYel14uo6O8d2uqk	TLSv10	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	null	CsxknGIi	6734f37431670b3ab4292b8f60f29984
ssl	CerXhu2qWiBtSn32Sa	TLSv12	TLS_RSA_WITH_AES_128_CBC_SHA256	antivirusupdaty.com	CsxnGIi	2a458dd9c65afbcf591cd8c2a194b804
ssl	Ck67Ha4QlJ12hup7Rj	TLSv12	TLS_RSA_WITH_AES_128_CBC_SHA256	antivirusupdaty.com	CsxnGIi	2a458dd9c65afbcf591cd8c2a194b804
ssl	CAG2bNo0vlTbngAcc	TLSv12	TLS_RSA_WITH_AES_128_CBC_SHA256	antivirusupdaty.com	CsxnGIi	2a458dd9c65afbcf591cd8c2a194b804
ssl	CnupbW3uiuxjOcCdK8	TLSv10	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	null	CsxknGIi	6734f37431670b3ab4292b8f60f29984
ssl	ClQx1H8GNE7uxOmpk	TLSv12	TLS_RSA_WITH_AES_128_CBC_SHA256	antivirusupdaty.com	CsxnGIi	2a458dd9c65afbcf591cd8c2a194b804
ssl	CxNIhJ32ePzrHwuUB9	TLSv10	TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	null	CsxknGIi	6734f37431670b3ab4292b8f60f29984
ssl	CBgViZ1zM46Z3brgUk	TLSv12	TLS_RSA_WITH_AES_128_CBC_SHA256	antivirusupdaty.com	CsxnGIi	2a458dd9c65afbcf591cd8c2a194b804
ssl	CRFi781wB7BZFS0gMk	TLSv12	TLS_RSA_WITH_AES_128_CBC_SHA256	antivirusupdaty.com	CsxnGIi	2a458dd9c65afbcf591cd8c2a194b804

Figure 44: ssl.log in BRIM

- The notice.log file is checked for Trickbot hash detection alerts, confirming the identification of malicious communications.

  JA3 = 6734f37431670b3ab4292b8f60f29984 (Fingerprint of Trickbot)



```
from notice.log
```

_path	ts	note	msg	src	dst
notice	2021-05-26T21:43:21.324756Z	JA3::TrickBotDetection	Detected target JA3 hash (6734f37431670b3ab4292b8f60f29984) for connection: [orig_	10.5.26.4	45.229.71.211
notice	2021-05-26T21:42:29.12809Z	JA3::TrickBotDetection	Detected target JA3 hash (6734f37431670b3ab4292b8f60f29984) for connection: [orig_	10.5.26.132	37.228.70.134
notice	2021-05-26T21:39:58.675893Z	JA3::TrickBotDetection	Detected target JA3 hash (6734f37431670b3ab4292b8f60f29984) for connection: [orig_	10.5.26.4	45.229.71.211
notice	2021-05-26T21:39:07.512179Z	JA3::TrickBotDetection	Detected target JA3 hash (6734f37431670b3ab4292b8f60f29984) for connection: [orig_	10.5.26.132	37.228.70.134
notice	2021-05-26T21:36:36.335769Z	JA3::TrickBotDetection	Detected target JA3 hash (6734f37431670b3ab4292b8f60f29984) for connection: [orig_	10.5.26.4	45.229.71.211
notice	2021-05-26T21:35:45.851319Z	JA3::TrickBotDetection	Detected target JA3 hash (6734f37431670b3ab4292b8f60f29984) for connection: [orig_	10.5.26.132	37.228.70.134
notice	2021-05-26T21:33:09.076338Z	JA3::TrickBotDetection	Detected target JA3 hash (6734f37431670b3ab4292b8f60f29984) for connection: [orig_	10.5.26.4	45.229.71.211
notice	2021-05-26T21:32:19.410544Z	JA3::TrickBotDetection	Detected target JA3 hash (6734f37431670b3ab4292b8f60f29984) for connection: [orig_	10.5.26.132	37.228.70.134
notice	2021-05-26T21:29:41.920756Z	JA3::TrickBotDetection	Detected target JA3 hash (6734f37431670b3ab4292b8f60f29984) for connection: [orig_	10.5.26.4	45.229.71.211
notice	2021-05-26T21:28:57.684413Z	JA3::TrickBotDetection	Detected target JA3 hash (6734f37431670b3ab4292b8f60f29984) for connection: [orig_	10.5.26.132	37.228.70.134
notice	2021-05-26T21:26:10.770535Z	JA3::TrickBotDetection	Detected target JA3 hash (6734f37431670b3ab4292b8f60f29984) for connection: [orig_	10.5.26.4	45.229.71.211
notice	2021-05-26T21:25:35.545815Z	JA3::TrickBotDetection	Detected target JA3 hash (6734f37431670b3ab4292b8f60f29984) for connection: [orig_	10.5.26.132	37.228.70.134
notice	2021-05-26T21:22:41.607225Z	JA3::TrickBotDetection	Detected target JA3 hash (6734f37431670b3ab4292b8f60f29984) for connection: [orig_	10.5.26.4	45.229.71.211
notice	2021-05-26T21:22:12.689179Z	JA3::TrickBotDetection	Detected target JA3 hash (6734f37431670b3ab4292b8f60f29984) for connection: [orig_	10.5.26.132	37.228.70.134
notice	2021-05-26T21:19:13.592807Z	JA3::TrickBotDetection	Detected target JA3 hash (6734f37431670b3ab4292b8f60f29984) for connection: [orig_	10.5.26.4	45.229.71.211
notice	2021-05-26T21:18:50.653105Z	JA3::TrickBotDetection	Detected target JA3 hash (6734f37431670b3ab4292b8f60f29984) for connection: [orig_	10.5.26.132	37.228.70.134
notice	2021-05-26T21:15:45.719433Z	JA3::TrickBotDetection	Detected target JA3 hash (6734f37431670b3ab4292b8f60f29984) for connection: [orig_	10.5.26.4	45.229.71.211

Figure 45: notice.log in BRIM

10 File Extraction and Hash Computation in Zeek

10.1 How It Works

The process of file extraction in Zeek can be summarized as follows:

1. **Event Detection:** Zeek scripts react to network events, such as the presence of a file in the network stream.

2. **File Type Identification:** The script identifies the type of the file using its MIME type, enabling appropriate handling.

3. **File Handling:** Based on the identified type, the script determines the handling procedure and naming convention for the file.

4. **Analyzer Usage:** File analyzers, especially the extraction analyzer, are utilized to automatically save the file to the disk.

5. **Extraction Configuration:** Specific parameters, such as the output filename, are defined for the extraction process.

10.2 File Extraction Script :

```
event file_sniff(f: fa_file, meta: fa_metadata)
{
    if (!meta?$mime_type) return;
    local filename: string;
    switch (meta$mime_type) {
        case "text/plain":
            print fmt("Detected Text file: %s", f$id);
            filename = fmt("extracted_text_%s.txt", f$id);
            break;

        case "text/html":
            print fmt("Detected HTML file: %s", f$id);
            filename = fmt("extracted_html_%s.html", f$id);
            break;

        case "image/png":
            print fmt("Detected PNG file: %s", f$id);
            filename = fmt("extracted_png_%s.png", f$id);
            break;

        default:
            return;
    }
    Files::add_analyzer(f, Files::ANALYZER_MD5);
    Files::add_analyzer(f, Files::ANALYZER_EXTRACT, [$extract_filename=filename])
    ;

    print fmt("File will be extracted to: %s", filename);
}

event file_hash(f: fa_file, kind: string, hash: string)
{
```

```
32      if (kind == "MD5")
33      {
34          print fmt("MD5 hash of file %s: %s", f$id, hash);
35      }
36  }
37
38  event zeek_init()
39  {
40      print "Zeek FILE Monitoring Script Initialized.";
41  }
```

Listing 10: File EXtraction Script

10.3 Script Explanation

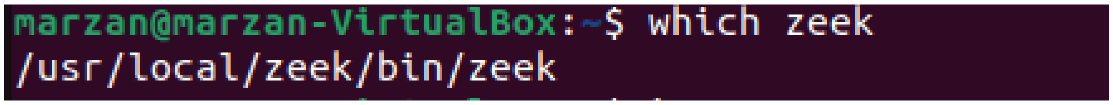
1. **Event Detection** (Line 1) : Triggered by the `file_sniff` event, the script processes each file detected in network traffic.
2. **MIME Type Check** (Lines 3-19) The script exits the function early if a file does not have a MIME type associated with it.
3. **File Classification** (Lines 3-19) : Files are classified based on their MIME type (text, HTML, PNG) and assigned unique filenames.
4. **Analyzer Addition** (Lines 24-25) : The script adds MD5 and extraction analyzers for hashing and saving the files, respectively.
5. **Extraction Notification** (Line 27) It logs the path where the file will be extracted to.
6. **Hash Event** (Lines 30-36) : During the `file_hash` event, the MD5 hash of the file is logged.
7. **Initialization** (Lines 38-40) : The `zeek_init` event signals that the script is ready for file monitoring.

10.4 Simulation

We will commence by extracting files through a script that operates on both live network monitoring data and pre-captured pcap files.

10.4.1 File Extraction from Live Network

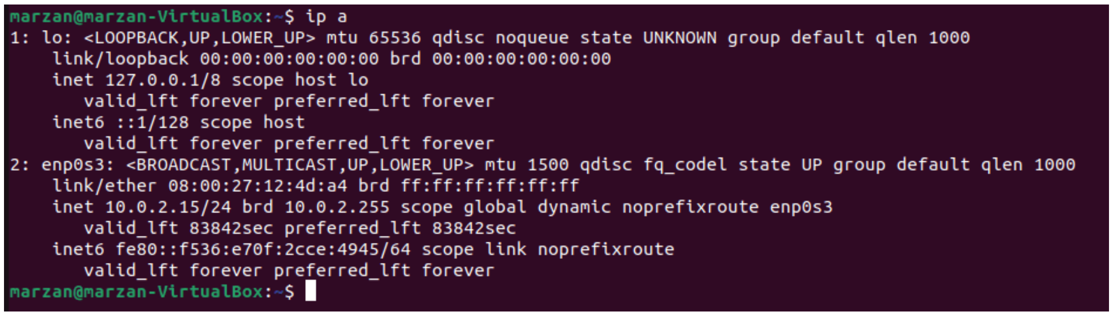
- The initial step involves locating the Zeek executable on the system.



```
marzan@marzan-VirtualBox:~$ which zeek
/usr/local/zeek/bin/zeek
```

Figure 46: zeek path

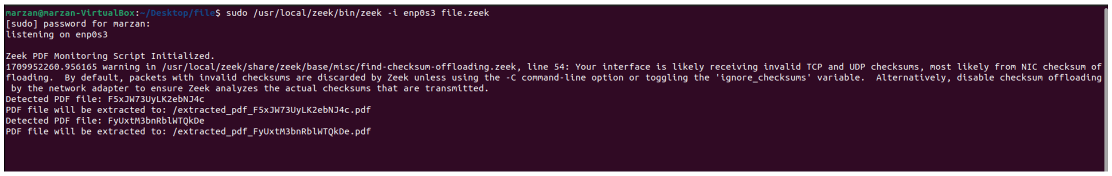
- Subsequently, we will identify the active network interface on the machine to monitor the network traffic.



```
marzan@marzan-VirtualBox:~$ ip a
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
    link/ether 08:00:27:12:4d:a4 brd ff:ff:ff:ff:ff:ff
    inet 10.0.2.15/24 brd 10.0.2.255 scope global dynamic noprefixroute enp0s3
       valid_lft 83842sec preferred_lft 83842sec
    inet6 fe80::f536:e70f:2cce:4945/64 scope link noprefixroute
       valid_lft forever preferred_lft forever
marzan@marzan-VirtualBox:~$ 
```

Figure 47: active network interfaces

- Following this, we initiate monitoring on the selected interface (enp0s3) using Zeek, which results in the extraction of 2 PDF files from the network traffic



```
marzan@marzan-VirtualBox:~/Desktop/file$ sudo /usr/local/zeek/bin/zeek -i enp0s3 file.zeek
[sudo] password for marzan:
listening on enp0s3

Zeek PDF Monitoring Script Initialized.
1709952260.956165 warning in /usr/local/zeek/share/zeek/base/misc/find-checksum-offloading.zeek, line 54: Your interface is likely receiving invalid TCP and UDP checksums, most likely from NIC checksum of
floading.  By default, packets with invalid checksums are discarded by Zeek unless using the -C command-line option or toggling the 'ignore_checksums' variable.  Alternatively, disable checksum offloading
 by the network adapter to ensure Zeek analyzes the actual checksums that are transmitted.
Detected PDF file: F5xJW73UyLK2ebNJ4c
PDF file will be extracted to: /extracted_pdf_F5xJW73UyLK2ebNJ4c.pdf
Detected PDF file: FyUxtM3bnRblWTQkDe
PDF file will be extracted to: /extracted_pdf_FyUxtM3bnRblWTQkDe.pdf
```

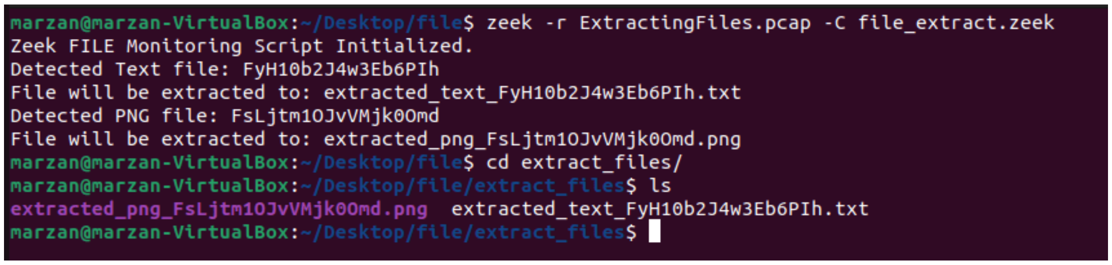
Figure 48: PDF file extraction from live network monitoring

10.4.2 File Extraction from PCAP file

- The same extraction process can be applied to pre-captured pcap files containing file transfer traffic

  For this exercise, a pcap file is available at PCAP file URL : `https://www.cloudshark.org/captures/a9472fbe700a`
  The file can be downloaded by selecting "Export — Download".

  By analyzing the pcap file with Zeek, we can extract files embedded within the network traffic. The extracted files, which include a PNG file and a TXT file, are stored in the **/extract_files** folder and are named according to their unique identifiers as logged by Zeek.



```
marzan@marzan-VirtualBox:~/Desktop/file$ zeek -r ExtractingFiles.pcap -C file_extract.zeek
Zeek FILE Monitoring Script Initialized.
Detected Text file: FyH10b2J4w3Eb6PIh
File will be extracted to: extracted_text_FyH10b2J4w3Eb6PIh.txt
Detected PNG file: FsLjtm1OJvVMjk0Omd
File will be extracted to: extracted_png_FsLjtm1OJvVMjk0Omd.png
marzan@marzan-VirtualBox:~/Desktop/file$ cd extract_files/
marzan@marzan-VirtualBox:~/Desktop/file/extract_files$ ls
extracted_png_FsLjtm1OJvVMjk0Omd.png  extracted_text_FyH10b2J4w3Eb6PIh.txt
marzan@marzan-VirtualBox:~/Desktop/file/extract_files$ 
```

Figure 49: PDF file extraction from pcap file

- The **extract_files** folder contains the two files that were extracted from the pcap file, showcasing Zeek's file extraction capabilities.

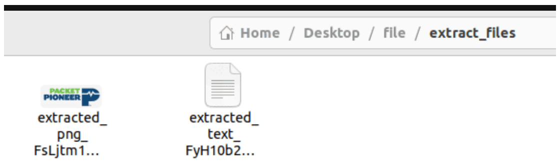

Home / Desktop / file / extract_files
extracted_png_FsLjtm1...
extracted_text_FyH10b2...

Figure 50: Files inside extract_files folder

- A PNG file has been successfully extracted from the network traffic.

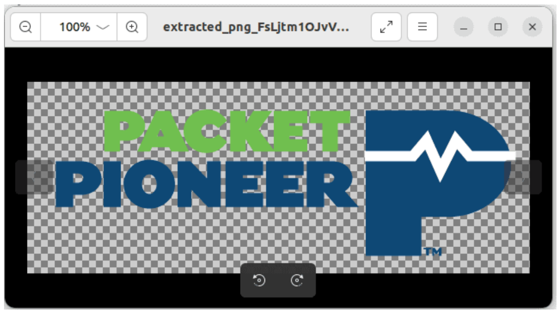
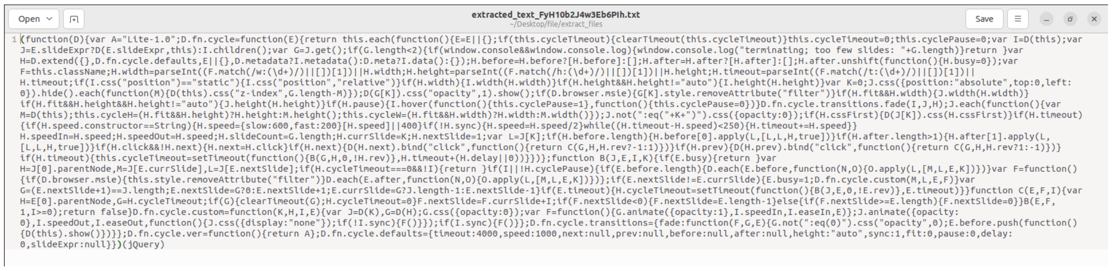


Figure 51: PNG file

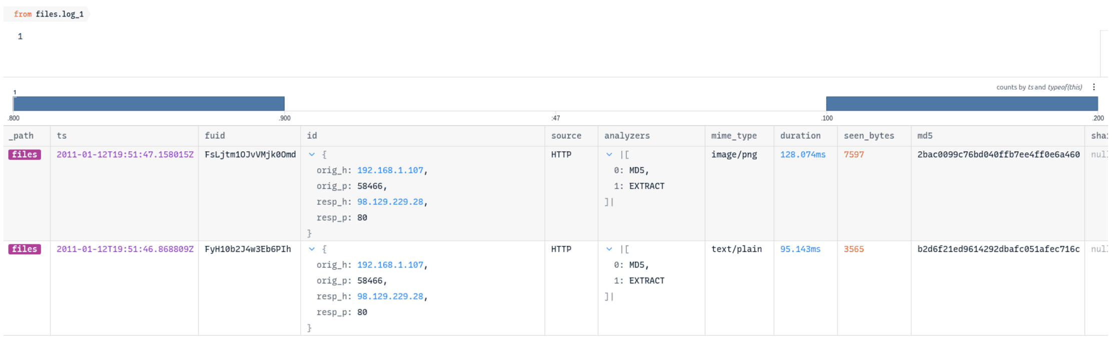
- A TXT file has been successfully extracted from the network traffic.


Open
extracted_text_FyH10b2J4w3Eb6PIh.txt
~/Desktop/file/extract_files
Save

```
(function(D){var A="Lite-1.0";D.fn.cycle=function(E){return this.each(function(){E=E||{};if(this.cycleTimeout){clearTimeout(this.cycleTimeout)}this.cycleTimeout=0;this.cyclePause=0;var I=D(this);var
J=E.slideExpr?D(E.slideExpr,this):I.children();var G=J.get();if(G.length<2){if(window.console&&window.console.log){window.console.log("terminating; too few slides: "+G.length)}return }var
H=D.extend({},D.fn.cycle.defaults,E||{},D.metadata?I.metadata():D.meta?I.data():{});H.before=H.before?[H.before]:[];H.after=H.after?[H.after]:[];H.after.unshift(function(){H.busy=0});var
F=this.className;H.width=parseInt((F.match(/w:(\d+)/)||[])[1])||H.width;H.height=parseInt((F.match(/h:(\d+)/)||[])[1])||H.height;H.timeout=parseInt((F.match(/t:(\d+)/)||[])[1])||
H.timeout;if(I.css("position")=="static"){I.css("position","relative")}if(H.width){I.width(H.width)}if(H.height&&H.height!="auto"){I.height(H.height)}var K=0;J.css({position:"absolute",top:0,left:
0}).hide().each(function(M){D(this).css("z-index",G.length-M)});D(G[K]).css("opacity",1).show();if(D.browser.msie){G[K].style.removeAttribute("filter")}if(H.fit&&H.width){J.width(H.width)}
if(H.fit&&H.height&&H.height!="auto"){J.height(H.height)}if(H.pause){I.hover(function(){this.cyclePause=1},function(){this.cyclePause=0})}D.fn.cycle.transitions.fade(I,J,H);J.each(function(){var
M=D(this);this.cycleH=(H.fit&&H.height)?H.height:M.height();this.cycleW=(H.fit&&H.width)?H.width:M.width()});J.not(":eq("+K+")").css({opacity:0});if(H.cssFirst){D(J[K]).css(H.cssFirst)}if(H.timeout)
{if(H.speed.constructor==String){H.speed={slow:600,fast:200}[H.speed]||400}if(!H.sync){H.speed=H.speed/2}while((H.timeout-H.speed)<250){H.timeout+=H.speed}}
H.speedIn=H.speed;H.speedOut=H.speed;H.slideCount=G.length;H.currSlide=K;H.nextSlide=1;var L=J[K];if(H.before.length){H.before[0].apply(L,[L,L,H,true])}if(H.after.length>1){H.after[1].apply(L,
[L,L,H,true])}if(H.click&&!H.next){H.next=H.click}if(H.next){D(H.next).bind("click",function(){return C(G,H,H.rev?-1:1)})}if(H.prev){D(H.prev).bind("click",function(){return C(G,H,H.rev?1:-1)})}
if(H.timeout){this.cycleTimeout=setTimeout(function(){B(G,H,0,!H.rev)},H.timeout+(H.delay||0))}})};function B(J,E,I,K){if(E.busy){return }var
H=J[0].parentNode,M=J[E.currSlide],L=J[E.nextSlide];if(H.cycleTimeout===0&&!I){return }if(I||!H.cyclePause){if(E.before.length){D.each(E.before,function(N,O){O.apply(L,[M,L,E,K])})}var F=function()
{if(D.browser.msie){this.style.removeAttribute("filter")}D.each(E.after,function(N,O){O.apply(L,[M,L,E,K])})};if(E.nextSlide!=E.currSlide){E.busy=1;D.fn.cycle.custom(M,L,E,F)}var
G=(E.nextSlide+1)==J.length;E.nextSlide=G?0:E.nextSlide+1;E.currSlide=G?J.length-1:E.nextSlide-1}if(E.timeout){H.cycleTimeout=setTimeout(function(){B(J,E,0,!E.rev)},E.timeout)}}function C(E,F,I){var
H=E[0].parentNode,G=H.cycleTimeout;if(G){clearTimeout(G);H.cycleTimeout=0}F.nextSlide=F.currSlide+I;if(F.nextSlide<0){F.nextSlide=E.length-1}else{if(F.nextSlide>=E.length){F.nextSlide=0}}B(E,F,
1,I>=0);return false}D.fn.cycle.custom=function(K,H,I,E){var J=D(K),G=D(H);G.css({opacity:0});var F=function(){G.animate({opacity:1},I.speedIn,I.easeIn,E)};J.animate({opacity:
0},I.speedOut,I.easeOut,function(){J.css({display:"none"});if(!I.sync){F()}});if(I.sync){F()}};D.fn.cycle.transitions={fade:function(F,G,E){G.not(":eq(0)").css("opacity",0);E.before.push(function()
{D(this).show()})}};D.fn.cycle.ver=function(){return A};D.fn.cycle.defaults={timeout:4000,speed:1000,next:null,prev:null,before:null,after:null,height:"auto",sync:1,fit:0,pause:0,delay:
0,slideExpr:null}})(jQuery)
```

Figure 52: TXT file

- Additionally, the files.log file, which can be viewed in BRIM, provides an MD5 hash value for each extracted file, facilitating further analysis and verification of the files' integrity.



```
from files.log_1
1
```

_path	ts	fuid	id	source	analyzers	mime_type	duration	seen_bytes	md5	sha
files	2011-01-12T19:51:47.158015Z	FsLjtm1OJvVMjk0Omd	{ orig_h: 192.168.1.107, orig_p: 58466, resp_h: 98.129.229.28, resp_p: 80 }	HTTP	|[0: MD5, 1: EXTRACT]|	image/png	128.074ms	7597	2bac0099c76bd040ffb7ee4ff0e6a460	nul
files	2011-01-12T19:51:46.868809Z	FyH10b2J4w3Eb6PIh	{ orig_h: 192.168.1.107, orig_p: 58466, resp_h: 98.129.229.28, resp_p: 80 }	HTTP	|[0: MD5, 1: EXTRACT]|	text/plain	95.143ms	3565	b2d6f21ed9614292dbafc051afec716c	nul

Figure 53: files.log in BRIM

11 Malware Hash Detection in Zeek

11.1 How It Works

The Zeek script integrates with Team Cymru's Malware Hash Registry to enhance network security monitoring by identifying files with hash values known to be associated with malware. The process is outlined as follows:

- **Initialization :** The script initializes by loading essential frameworks for file handling and notice management, setting the groundwork for its operations.
- **Notice Extension :** It extends Zeek's notice mechanism to include a new type specifically for malware hash matches, facilitating targeted alerts.
- **File Filtering :** File downloads are filtered by MIME type, focusing the analysis on file formats commonly associated with malware dissemination.
- **Malware Hash Lookup :** For each relevant file, the script performs a DNS TXT lookup against the Malware Hash Registry using the file's SHA-1 hash as the query, efficiently leveraging external threat intelligence.
- **Lookup Evaluation :** The lookup response, containing the detection rate and first detection date, is evaluated against a predefined threshold to determine the file's potential threat level.
- **Notice Generation :** If the detection criteria are met, a notice is generated, including the malware detection rate, the URL for further information, and the file's last seen date, thereby informing the network administrator of potential security threats.
- **Network Defense Enhancement :** This integration allows for an automated, real-time response to the transmission of potentially malicious files, enhancing the network's defensive posture against malware.

11.2 Detect-MHR Script:

```
##! Detect file downloads that have hash values matching files in Team
##! Cymru's Malware Hash Registry (https://www.team-cymru.com/mhr.html).

@load base/frameworks/files
@load base/frameworks/notice
@load frameworks/files/hash-all-files

module TeamCymruMalwareHashRegistry;

export {
  redef enum Notice::Type += {
    Match
  };

  ## File types to attempt matching against the Malware Hash Registry.
  option match_file_types = /application\/x-dosexec/ |
                            /application\/vnd\.ms-cab-compressed/ |
                           /application\/pdf/ |
                           /application\/x-shockwave-flash/ |
                           /application\/x-java-applet/ |
                           /application\/jar/ |
                           /video\/mp4/;


  option match_sub_url = "https://www.virustotal.com/gui/search/%s";
  option notice_threshold = 10;
}


function do_mhr_lookup(hash: string, fi: Notice::FileInfo)
  {
  local hash_domain = fmt("%s.malware.hash.cymru.com", hash);
  when [hash, fi, hash_domain] ( local MHR_result = lookup_hostname_txt(hash_domain) )
    {
    local MHR_answer = split_string1(MHR_result, / /);

    if ( |MHR_answer| == 2 )
      {
      local mhr_detect_rate = to_count(MHR_answer[1]);
      if ( mhr_detect_rate >= notice_threshold )
        {
        local mhr_first_detected = double_to_time(to_double(MHR_answer[0]));

        local readable_first_detected = strftime("%Y-%m-%d %H:%M:%S", mhr_first_detected);

        local message = fmt("Malware Hash Registry Detection rate: %d%%  Last seen: %s", mhr_detect_rate, readable_first_detected);

        local virustotal_url = fmt(match_sub_url, hash);

        local n: Notice::Info = Notice::Info($note=Match, $msg=message, $sub=virustotal_url);
        Notice::populate_file_info2(fi, n);
        NOTICE(n);
        }
      }
    }
  }

event file_hash(f: fa_file, kind: string, hash: string)
  {
  if ( kind == "sha1" && f?$info && f$info?$mime_type &&
       match_file_types in f$info$mime_type )
    do_mhr_lookup(hash, Notice::create_file_info(f));
  }


```

Listing 11: detect-MHR.zeek

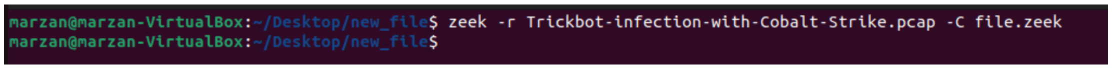
11.3 Detect-MHR Script Explanation :

1. **Module and Dependencies** (Lines 4-8) : The script introduces the ‘TeamCymruMalwareHashRegistry‘ module and includes necessary Zeek frameworks for file operations and notice generation.
2. **Notice Type Extension** (Lines 11-12) : Extends ‘Notice::Type‘ with ‘Match‘ to signify malware hash matches.
3. **File Type Filtering** (Lines 16-22) : Defines ‘match_file_types‘ to specify MIME types of interest, including executables, archives, PDFs, Flash, Java applets, and MP4 videos.
4. **Malware Information URL** (Line 25) : The ‘match_sub_url‘ option formats a URL to VirusTotal, substituting ‘
5. **Detection Threshold** (Line 26) : Sets a ‘notice_threshold‘ to determine the minimum detection rate by antivirus engines for generating a notice.
6. **Malware Hash Lookup** (Lines 30-48) : Implements ‘do_mhr_lookup‘ to query the hash in Team Cymru’s Malware Hash Registry via DNS TXT lookup.
7. **Handling Lookup Results** (Lines 50-52) : Processes lookup results, generating a notice if the detection rate exceeds the set threshold, including the rate, last seen date, and a VirusTotal link.
8. **File Hash Event** (Lines 58-62) : The ‘file_hash‘ event initiates the malware hash lookup for SHA-1 hashes of files matching the specified MIME types.

11.4 Simulation :

Here is the link to VirusTotal: VirusTotal
Here is the link to PCAP file: malware-traffic-analysis

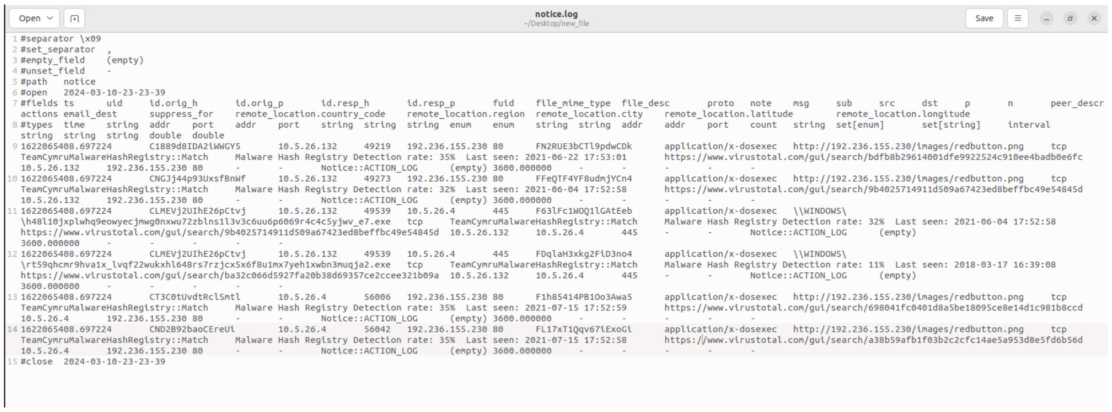
- To begin the analysis, the malicious pcap file is processed using the detect-MHR.zeek script, specifically designed for malware hash detection.



```
marzan@marzan-VirtualBox:~/Desktop/new_file$ zeek -r Trickbot-infection-with-Cobalt-Strike.pcap -C file.zeek
marzan@marzan-VirtualBox:~/Desktop/new_file$
```

Figure 54: Running PCAP with Script

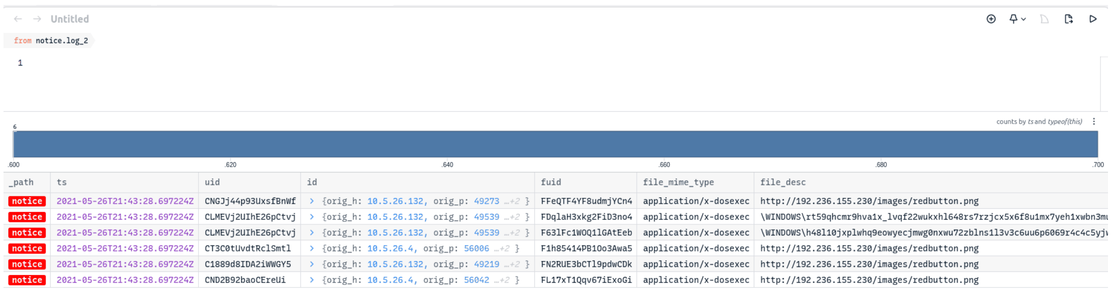
- The notice.log entries highlight malware detections with the following critical information: the detection rate, the last seen date, and a link to VirusTotal for further investigation. For instance, one entry shows a 3% detection rate for malware last seen on 2021-06-22, with a corresponding VirusTotal search link for the hash bdfb8b29614001dfe9922524c910ee4badb0e6fc. Another entry indicates a 32% detection rate for malware last seen on 2021-06-04, linked to the hash 9b4025714911d509a67423ed8beffbc49e54845d on VirusTotal. These details facilitate a deeper analysis of the detected threats.



```
#separator \x09
#set_separator  ,
#empty_field    (empty)
#unset_field    -
#path   notice
#open   2024-03-10-23-23-39
#fields ts      uid     id.orig_h       id.orig_p       id.resp_h       id.resp_p       fuid    file_mime_type  file_desc       proto   note    msg     sub     src     dst     p       n       peer_descr
actions email_dest      suppress_for    remote_location.country_code    remote_location.region  remote_location.city    remote_location.latitude        remote_location.longitude
#types  time    string  addr    port    addr    port    string  string  string  enum    enum    string  string  addr    addr    port    count   string  set[enum]       set[string]     interval
string  string  string  double  double
1622065408.697224       C1889d8IDA2iWWGY5       10.5.26.132     49219   192.236.155.230 80      FN2RUE3bCTl9pdwCDk      application/x-dosexec   http://192.236.155.230/images/redbutton.png     tcp
TeamCymruMalwareHashRegistry::Match     Malware Hash Registry Detection rate: 35%  Last seen: 2021-06-22 17:53:01       https://www.virustotal.com/gui/search/bdfb8b29614001dfe9922524c910ee4badb0e6fc
10.5.26.132     192.236.155.230 80      -       -       Notice::ACTION_LOG      (empty) 3600.000000     -       -       -       -       -
1622065408.697224       CNGJj44p93UxsfBnWf      10.5.26.132     49273   192.236.155.230 80      FFeQTF4YF8udmjYCn4      application/x-dosexec   http://192.236.155.230/images/redbutton.png     tcp
TeamCymruMalwareHashRegistry::Match     Malware Hash Registry Detection rate: 32%  Last seen: 2021-06-04 17:52:58       https://www.virustotal.com/gui/search/9b4025714911d509a67423ed8beffbc49e54845d
10.5.26.132     192.236.155.230 80      -       -       Notice::ACTION_LOG      (empty) 3600.000000     -       -       -       -       -
1622065408.697224       CLMEVj2UIhE26pCtvj      10.5.26.132     49539   10.5.26.4       445     F63lFc1WOQ1lGAtEeb      application/x-dosexec   \\WINDOWS\
\h48l10jxplwhq9eowyecjmwg0nxwu72zblns1l3v3c6uu6p6069r4c4c5yjwv_e7.exe   tcp     TeamCymruMalwareHashRegistry::Match     Malware Hash Registry Detection rate: 32%  Last seen: 2021-06-04 17:52:58
https://www.virustotal.com/gui/search/9b4025714911d509a67423ed8beffbc49e54845d  10.5.26.132     10.5.26.4       445     -       -       Notice::ACTION_LOG      (empty)
3600.000000     -       -       -       -       -
1622065408.697224       CLMEVj2UIhE26pCtvj      10.5.26.132     49539   10.5.26.4       445     FDqlaH3xkg2FiD3no4      application/x-dosexec   \\WINDOWS\
\rt59qhcmr9hva1x_lvqf22wukxhl648rs7rzjcx5x6f8u1mx7yeh1xwbn3muqja2.exe   tcp     TeamCymruMalwareHashRegistry::Match     Malware Hash Registry Detection rate: 11%  Last seen: 2018-03-17 16:39:08
https://www.virustotal.com/gui/search/ba32c066d5927fa20b38d69357ce2ccee321b09a  10.5.26.132     10.5.26.4       445     -       -       Notice::ACTION_LOG      (empty)
3600.000000     -       -       -       -       -
1622065408.697224       CT3C0tUvdtRclSmtl       10.5.26.4       56006   192.236.155.230 80      F1h85414PB1Oo3Awa5      application/x-dosexec   http://192.236.155.230/images/redbutton.png     tcp
TeamCymruMalwareHashRegistry::Match     Malware Hash Registry Detection rate: 35%  Last seen: 2021-07-15 17:52:59       https://www.virustotal.com/gui/search/698041fc0401d8a5be18095ce8e14d1c981b8ccd
10.5.26.4       192.236.155.230 80      -       -       Notice::ACTION_LOG      (empty) 3600.000000     -       -       -       -       -
1622065408.697224       CND2B92baoCEreUi        10.5.26.4       56042   192.236.155.230 80      FL17xT1Qqv67iExoGi      application/x-dosexec   http://192.236.155.230/images/redbutton.png     tcp
TeamCymruMalwareHashRegistry::Match     Malware Hash Registry Detection rate: 35%  Last seen: 2021-07-15 17:52:58       https://www.virustotal.com/gui/search/a38b59afb1f03b2c2cfc14ae5a953d8e5fd6b56d
10.5.26.4       192.236.155.230 80      -       -       Notice::ACTION_LOG      (empty) 3600.000000     -       -       -       -       -
#close  2024-03-10-23-23-39
```

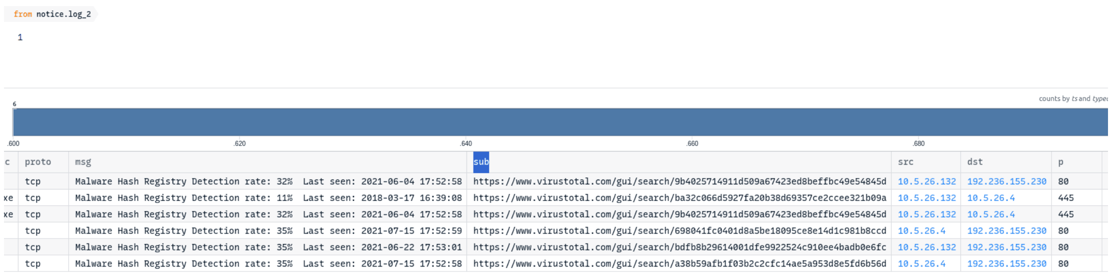
Figure 55: notice.log

- These detection logs can be viewed in a tabular format within the BRIM application, providing a user-friendly interface for detailed analysis.



_path	ts	uid	id	fuid	file_mime_type	file_desc
notice	2021-05-26T21:43:28.697224Z	CNGJj44p93UxsfBnWf	› {orig_h: 10.5.26.132, orig_p: 49273 …+2 }	FFeQTF4YF8udmjYCn4	application/x-dosexec	http://192.236.155.230/images/redbutton.png
notice	2021-05-26T21:43:28.697224Z	CLMEVj2UIhE26pCtvj	› {orig_h: 10.5.26.132, orig_p: 49539 …+2 }	FDqlaH3xkg2FiD3no4	application/x-dosexec	\WINDOWS\rt59qhcmr9hva1x_lvqf22wukxhl648rs7rzjcx5x6f8u1mx7yeh1xwbn3mu
notice	2021-05-26T21:43:28.697224Z	CLMEVj2UIhE26pCtvj	› {orig_h: 10.5.26.132, orig_p: 49539 …+2 }	F63lFc1WOQ1lGAtEeb	application/x-dosexec	\WINDOWS\h48l10jxplwhq9eowyecjmwg0nxwu72zblns1l3v3c6uu6p6069r4c4c5yjw
notice	2021-05-26T21:43:28.697224Z	CT3C0tUvdtRclSmtl	› {orig_h: 10.5.26.4, orig_p: 56006 …+2 }	F1h85414PB1Oo3Awa5	application/x-dosexec	http://192.236.155.230/images/redbutton.png
notice	2021-05-26T21:43:28.697224Z	C1889d8IDA2iWWGY5	› {orig_h: 10.5.26.132, orig_p: 49219 …+2 }	FN2RUE3bCTl9pdwCDk	application/x-dosexec	http://192.236.155.230/images/redbutton.png
notice	2021-05-26T21:43:28.697224Z	CND2B92baoCEreUi	› {orig_h: 10.5.26.4, orig_p: 56042 …+2 }	FL17xT1Qqv67iExoGi	application/x-dosexec	http://192.236.155.230/images/redbutton.png

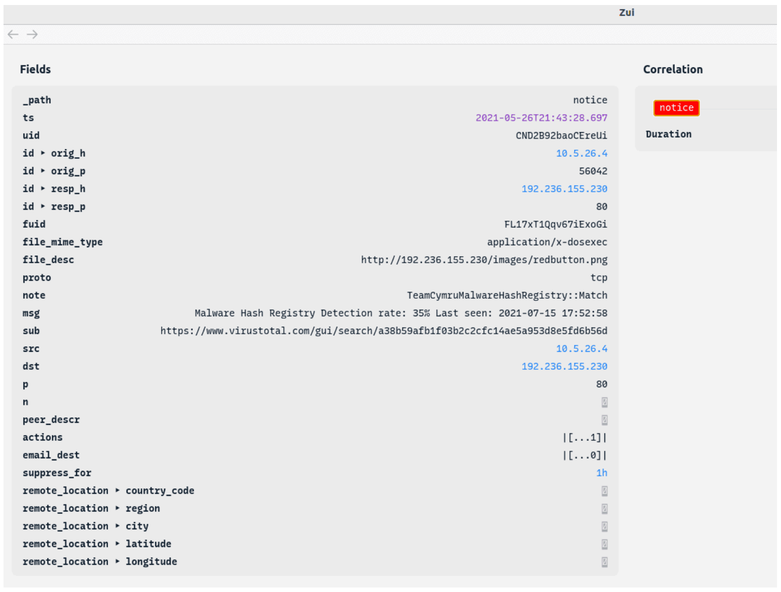
Figure 56: notice.log in BRIM



c	proto	msg	sub	src	dst	p
	tcp	Malware Hash Registry Detection rate: 32% Last seen: 2021-06-04 17:52:58	https://www.virustotal.com/gui/search/9b4025714911d509a67423ed8beffbc49e54845d	10.5.26.132	192.236.155.230	80
xe	tcp	Malware Hash Registry Detection rate: 11% Last seen: 2018-03-17 16:39:08	https://www.virustotal.com/gui/search/ba32c066d5927fa20b38d69357ce2ccee321b09a	10.5.26.132	10.5.26.4	445
xe	tcp	Malware Hash Registry Detection rate: 32% Last seen: 2021-06-04 17:52:58	https://www.virustotal.com/gui/search/9b4025714911d509a67423ed8beffbc49e54845d	10.5.26.132	10.5.26.4	445
	tcp	Malware Hash Registry Detection rate: 35% Last seen: 2021-07-15 17:52:59	https://www.virustotal.com/gui/search/698041fc0401d8a5be18095ce8e14d1c981b8ccd	10.5.26.4	192.236.155.230	80
	tcp	Malware Hash Registry Detection rate: 35% Last seen: 2021-06-22 17:53:01	https://www.virustotal.com/gui/search/bdfb8b29614001dfe9922524c910ee4badb0e6fc	10.5.26.132	192.236.155.230	80
	tcp	Malware Hash Registry Detection rate: 35% Last seen: 2021-07-15 17:52:58	https://www.virustotal.com/gui/search/a38b59afb1f03b2c2cfc14ae5a953d8e5fd6b56d	10.5.26.4	192.236.155.230	80

Figure 57: notice.log in BRIM

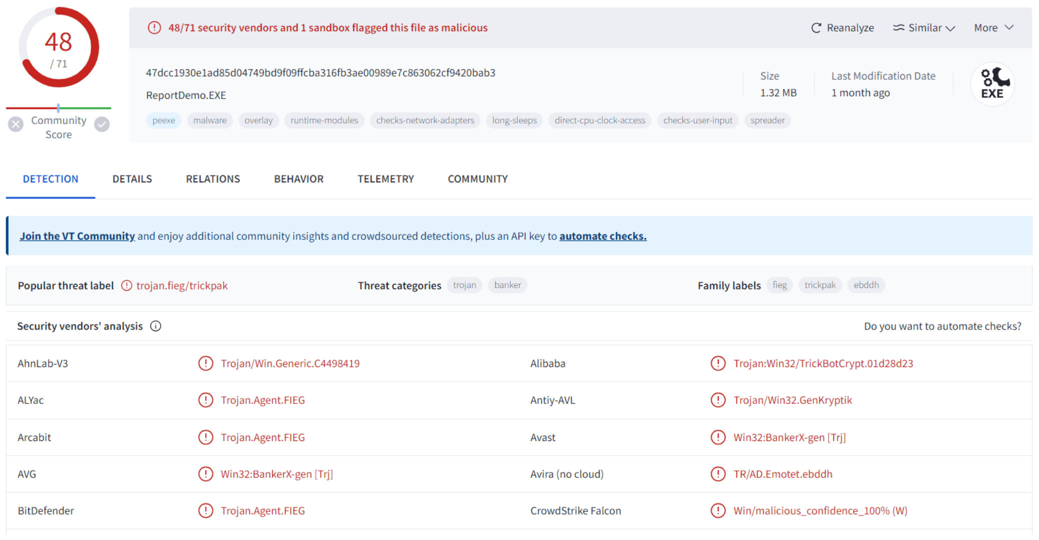
- Within BRIM, each entry's detailed information is accessible, including path, timestamp, unique identifiers, file information, MIME type, descriptions, protocols, notices, messages, source and destination details, and more. This detailed view aids in a comprehensive analysis of each detected event.



_path	notice
ts	2021-05-26T21:43:28.697
uid	CND2B92baoCEreUi
id ▸ orig_h	10.5.26.4
id ▸ orig_p	56042
id ▸ resp_h	192.236.155.230
id ▸ resp_p	80
fuid	FL17xT1Qqv67iExoGi
file_mime_type	application/x-dosexec
file_desc	http://192.236.155.230/images/redbutton.png
proto	tcp
note	TeamCymruMalwareHashRegistry::Match
msg	Malware Hash Registry Detection rate: 35% Last seen: 2021-07-15 17:52:58
sub	https://www.virustotal.com/gui/search/a38b59afb1f03b2c2cfc14ae5a953d8e5fd6b56d
src	10.5.26.4
dst	192.236.155.230
p	80
n	
peer_descr	
actions	|[...1]|
email_dest	|[...0]|
suppress_for	1h
remote_location ▸ country_code	
remote_location ▸ region	
remote_location ▸ city	
remote_location ▸ latitude	
remote_location ▸ longitude	

Figure 58: Entries of notice.log in BRIM

- Following the links in the notice.log to VirusTotal provides a detailed security report for each file hash, including the community score, detection rates by various security vendors, and additional malware characteristics. This external validation offers insights into the potential threats and the behavior of the detected files.



48/71 security vendors and 1 sandbox flagged this file as malicious

47dcc1930e1ad85d04749bd9f09ffcba316fb3ae00989e7c863062cf9420bab3

ReportDemo.EXE

peexe · malware · overlay · runtime-modules · checks-network-adapters · long-sleeps · direct-cpu-clock-access · checks-user-input · spreader

Size: 1.32 MB — Last Modification Date: 1 month ago

Join the VT Community and enjoy additional community insights and crowdsourced detections, plus an API key to automate checks.

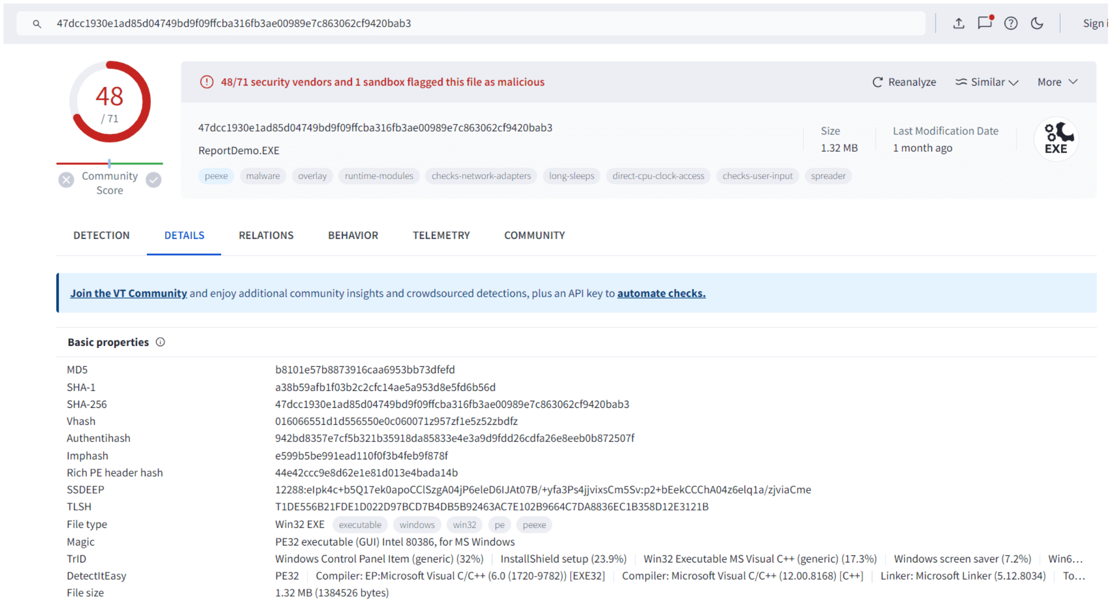
Popular threat label: trojan.fieg/trickpak — Threat categories: trojan, banker — Family labels: fieg, trickpak, ebddh

Security vendors' analysis — Do you want to automate checks?

AhnLab-V3	Trojan/Win.Generic.C4498419	Alibaba	Trojan:Win32/TrickBotCrypt.01d28d23
ALYac	Trojan.Agent.FIEG	Antiy-AVL	Trojan/Win32.GenKryptik
Arcabit	Trojan.Agent.FIEG	Avast	Win32:BankerX-gen [Trj]
AVG	Win32:BankerX-gen [Trj]	Avira (no cloud)	TR/AD.Emotet.ebddh
BitDefender	Trojan.Agent.FIEG	CrowdStrike Falcon	Win/malicious_confidence_100% (W)

Figure 59: VirusTotal Page Showing Malicious Percentage

- The VirusTotal reports also reveal basic file properties such as MD5, SHA-1, and SHA-256 hashes, providing an authoritative fingerprint of the files for cross-reference and further investigation into their origins, distribution, and impact.



47dcc1930e1ad85d04749bd9f09ffcba316fb3ae00989e7c863062cf9420bab3

48/71 security vendors and 1 sandbox flagged this file as malicious

47dcc1930e1ad85d04749bd9f09ffcba316fb3ae00989e7c863062cf9420bab3

ReportDemo.EXE

Basic properties

MD5	b8101e57b8873916caa6953bb73dfefd
SHA-1	a38b59afb1f03b2c2cfc14ae5a953d8e5fd6b56d
SHA-256	47dcc1930e1ad85d04749bd9f09ffcba316fb3ae00989e7c863062cf9420bab3
Vhash	016066551d1d556550e0c060071z957zf1e5z52zbdfz
Authentihash	942bd8357e7cf5b321b35918da85833e4e3a9d9fdd26cdfa26e8eeb0b872507f
Imphash	e599b5be991ead110f0f3b4feb9f878f
Rich PE header hash	44e42ccc9e8d62e1e81d013e4bada14b
SSDEEP	12288:elpk4c+b5Q17ek0apoCClSzgA04jP6eleD6IJAt07B/+yfa3Ps4jjvixsCm5Sv:p2+bEekCCChA04z6elq1a/zjviaCme
TLSH	T1DE556B21FDE1D022D97BCD7B4DB5B92463AC7E102B9664C7DA8836EC1B358D12E3121B
File type	Win32 EXE · executable · windows · win32 · pe · peexe
Magic	PE32 executable (GUI) Intel 80386, for MS Windows
TrID	Windows Control Panel Item (generic) (32%) | InstallShield setup (23.9%) | Win32 Executable MS Visual C++ (generic) (17.3%) | Windows screen saver (7.2%) | Win6…
DetectItEasy	PE32 | Compiler: EP:Microsoft Visual C/C++ (6.0 (1720-9782)) [EXE32] | Compiler: Microsoft Visual C/C++ (12.00.8168) [C++] | Linker: Microsoft Linker (5.12.8034) | To…
File size	1.32 MB (1384526 bytes)

Figure 60: VirusTotal Page Showing File Info

12 Codebase Structure

Zeek (formerly Bro) is an open-source framework for network analysis, security monitoring, and forensics. Developed with contributions from many developers and led by key contributor Vern Paxson, a UC Berkeley professor, it has become a vital tool in the computer security community.

12.1 Language :

Zeek is primarily written in C++ and uses scripting languages for its analysis scripts. The transition to C++ has allowed Zeek to leverage the performance and efficiency of compiled languages, making it highly suitable for high-speed network environments.

12.2 Repository

The source code for Zeek is hosted on GitHub, making it accessible for community contributions, issue tracking, and feature requests.

- Access URL: `https://github.com/zeek/zeek`
- Repository Owner: Zeek

12.3 Key Components

1. **Packet Capture and Analysis Engine**: : Optimized with C++ for high-throughput network traffic processing, featuring selective data filtering and modular protocol analysis capabilities.
2. **Event Engine**: Built on a C++ publish-subscribe model, it efficiently generates and dispatches network events, separating event creation from processing for flexible script execution
3. **Scripting Language Interpreter**: Integral to Zeek, it executes analysis scripts, offering direct access to network events and data, and supports dynamic script management.
4. **Logging Framework**: Versatile logging system supporting multiple formats like JSON, CSV, and script-driven customization for tailored data recording
5. **Plugin Architecture**: Allows dynamic plugin integration without Zeek restarts, offering APIs for seamless extension of core functionalities
6. **Analysis Scripts and Signature Database**: : Runtime-loaded scripts in Zeek's language define detection logic and extend analysis, with a signature database identifying network threats.

12.4 Directories and Major Files :

12.4.1 /src - Core C++ Code

- **analyzer/**: Protocol analyzers dissect network protocols to generate events.

  - *ProtocolAnalyzer.cc / .h*: Base classes for protocol analyzers.
  - Protocol-specific subdirectories (e.g., *http/*, *dns/*) contain implementations for each analyzer.

- **file_analysis/**: Framework for file extraction and analysis.
  - *File.cc / .h*: Core classes for file analysis.
  - *Analyzer.cc / .h*: Base and implementation classes for file analyzers.
- **iosource/**: Input sources management, including packet capture.
  - *PktSrc.cc / .h*: Abstract base for packet sources.
  - *PcapSrc.cc / .h*: pcap-based packet source implementation.
- **logging/**: Zeek's structured logging framework.
  - *WriterFrontend.cc / .h*: Interfaces for log writers.
  - *writers/*: Implementations for various log output formats.
- *main.cc*: Main entry point, initializing the environment and starting the processing loop.
- *net_util.cc / .h*: Network utility functions for address manipulation.
- **policy/**: Script-level policy handlers for security policies and detections.
- *Reporter.cc / .h*: Reporting of warnings, errors, and runtime messages.
- **script_opt/**: Script execution optimizations for performance.
- **threading/**: Manages threading and concurrency for parallelized operations.

12.4.2 /scripts - Zeek Script Library

- **base/**: Default behaviors, protocol parsing, and logging scripts.
  - *protocols/*: Scripts for network protocol handling.
  - *frameworks/*: Core frameworks like logging, notices, and file analysis.
- **policy/**: Optional scripts for feature extensions or behavior modifications.

12.4.3 /aux - Auxiliary Tools and Scripts

Tools and scripts supporting Zeek's core functionality, aiding in tasks like plugin development, testing, or integration.

12.4.4 /testing - Testing Framework and Suites

- *btest/*: BTest testing framework scripts.
- *Traces/*: Network trace files for testing protocol analyzers and script behaviors.
- *Scripts/*: Defines test cases and expected outcomes.

12.4.5 /doc - Documentation

Zeek's documentation, including user guides, developer documentation, and API references, primarily in reStructuredText format.

12.4.6 /cmake - Build System Configurations

CMake configurations and scripts for compiler options, dependencies, and build targets.

12.4.7 /zeekctl - Command-Line Control Interface

(For older versions) Contains 'zeekctl' scripts and configurations. In newer versions, 'zeekctl' is a separate project.

12.4.8 Plugin Support

Framework for extending Zeek's capabilities via plugins, typically developed in separate repositories or directories.

13 References

Below are the key resources for Zeek, a powerful network analysis framework:

- Zeek Official Website: https://zeek.org

- Zeek Documentation: Detailed information about Zeek can be found on their documentation site at https://docs.zeek.org/en/master/

- Zeek GitHub Repository: For source code and development resources, visit Zeek's GitHub page at https://github.com/zeek/zeek