

Unraveling Network Attacks

Mastering Network Threat Detection & Automated Incident Response: Practical insights with Zeek and Wazuh

A S M Shamim Reza



Agenda

Learn how to analyze
and detect network
attacks using Zeek.

Map findings to the
MITRE ATT&CK
Framework.

Hands-On Exercise:
Attack Detection,
Analyze, Reporting
and Response.

[~]\$ Whoami



A S M Shamim Reza

- Founder & Chief of Research, *TheTeamPhoenix*
- SME & CT at *APNIC, Australia*
- ex-CTO, *Pipeline Inc. Japan*
- 12+ years, worked for *Link3 Technologies Limited*
- PC Member, *btNOG, npNOG, APNIC.*
- Ambassador, *Wazuh, USA*

“

The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards – and even then I have my doubts.

– Gene Spafford



Why Network Security Matters

Starbucks has gone back to pen and paper after vendor ransomware attack

News By Ellen Jennings-Trace published November 27, 2024

Coffee giant forced to go analogue to track employee pay



When you purchase through links on our site, we may earn an affiliate commission. [Here's how it works.](#)



- Starbucks stores are using pen and paper to track employee hours after attack
- Third-party Vendor Blue Yonder hit with ransomware attack
- Retail stores in the UK and US affected

Why Network Security Matters

BY THE NUMBERS; Too Many Threats

- **Mandiant** indicates it currently tracks **3,500 threat groups** in 2023, an increase of 900 from the previous year. The firm also started tracking 588 new malware families in 2022.
- In 2023, **Microsoft** indicated that it tracks **300 unique threat actors**, including **160 nation state actors** and **50 ransomware groups**
- In 2021, **Google's** Threat Analysis Group announced that it tracks **more than 270 government-sponsored actor groups** associated with **more than 50 countries**

Overview of Attack Vectors

An **attack vector** is a method or path that a **cyber attacker** can use to gain **unauthorized access** to a computer system, network, or application.

ICMP-based Command & Control (C2)
Reflection Attack
DNS Tunneling
DGA
ARP Spoofing
DDoS
ICMP Ping Flood
Ping of death

Name	Description
Anchor	Anchor has used ICMP in C2 communications.
APT3	An APT3 downloader establishes SOCKS5 connections for its initial C2.
Aria-body	Aria-body has used TCP in C2 communications.
BITTER	BITTER has used TCP for C2 communications.
Brute Ratel C4	Brute Ratel C4 has the ability to use TCP for external C2.
C0021	During C0021, the threat actors used TCP for some C2 communications.
Carbon	Carbon uses TCP and UDP for C2.
Cobalt Strike	Cobalt Strike can be configured to use TCP, ICMP, and UDP for C2 communications.
Crimson	Crimson uses a custom TCP protocol for C2.
Cryptoistic	Cryptoistic can use TCP in communications with C2.
Cuckoo Stealer	Cuckoo Stealer can use sockets for communications to its C2 server.
Cutting Edge	During Cutting Edge, threat actors used the Unix socket and a reverse TCP shell for C2 communications.
Derusbi	Derusbi binds to a raw socket on a random source port between 31800 and 31900 for C2.
FakeM	Some variants of FakeM use SSL to communicate with C2 servers.
FunnyDream	FunnyDream can communicate with C2 over TCP and UDP.
Gelsemium	Gelsemium has the ability to use TCP and UDP in C2 communications.
gh0st RAT	gh0st RAT has used an encrypted protocol within TCP segments to communicate with the C2.
HAFNIUM	HAFNIUM has used TCP for C2.
KEYPLUG	KEYPLUG can use TCP and KCP (KERN Communications Protocol) over UDP for C2 communication.
LookBack	LookBack uses a custom binary protocol over sockets for C2 communications.
LunarMail	LunarMail can ping a specific C2 URL with the ID of a victim machine in the subdomain.
Mafalda	Mafalda can use raw TCP for C2.

Network Security and Threat Hunting Concepts

Network Security	Threat Hunting
The practice of securing a computer network from unauthorized access, misuse, or attacks.	A proactive search for threats that evade traditional detection mechanisms.
Involves monitoring, detecting, and responding to potential threats targeting the network layer.	Focuses on identifying unusual patterns or behavior that might indicate malicious activity.

Network Security and Threat Hunting Concepts

Proactive Defense	Reactive Defense
Involves continuous monitoring, anomaly detection, and hunting for early signs of compromise	Responding to threats or attacks after detection.
Threat hunting, implementing advanced detection systems like Zeek, and mapping TTPs with frameworks like MITRE ATT&CK.	Incident response, patch management, and forensic analysis.

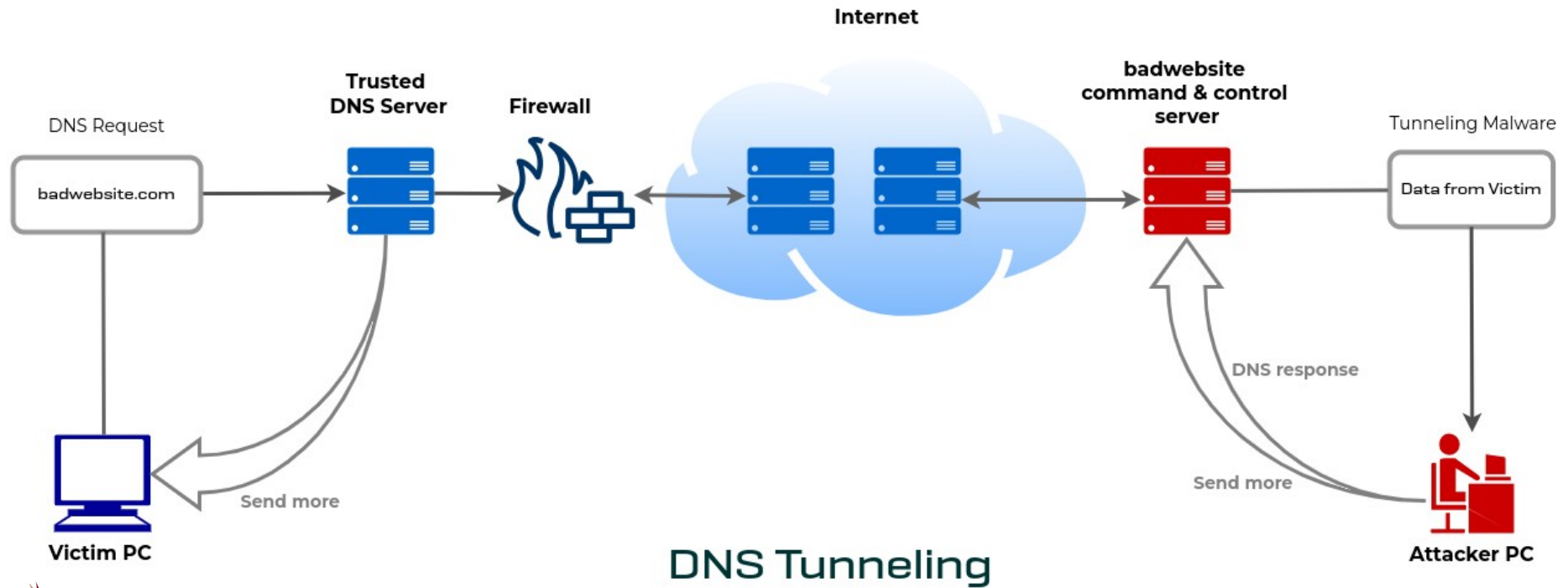
Attack Vectors | Example: *DNS Tunneling Attack*

"Scenario"

An attacker ***infiltrates*** an organization's network and uses DNS tunneling to ***exfiltrate*** sensitive data while maintaining a Command & Control (***C2***) channel. The attacker leverages ***DNS***, often overlooked in monitoring, to ***bypass*** traditional ***detection*** mechanisms.



Attack Vectors | Example: *DNS Tunneling Attack*



Historical Stats; DGA

Malware Family	DGA Type	Domain Sample	Generation Rate
Conficker	Time-based	ablksqvxx.biz	250/day
Bebloh	Dictionary-based	bankofstruggle.com	Variable
Necurs	Seed + Counter	duguxuwed.ru	~2048/day
Tinba	Date-based	wvjvhnytyco.com	1000/day
Bamital	XOR+Base64	xqhjvyoipw.co.uk	~100/day

Historical Stats; DGA

Birth of a Beast	Kraken (2008)
The Conficker Storm	50,000 unique domains daily (late 2008)
The New Normal	Integrated into GameOver Zeus, Necurs, Emotet
The Challenge	Overwhelming traditional blacklists
Evolving Tactics	Dynamic seeding, mimicking legitimate domains

Historical Stats; DNS Tunneling

Early Whispers	Conceptual discussions (1998)
Tools of the Trade	NSTX (2000), OzymanDNS (2004), iodine (2006), dnscat2 (2010)
Covert Operations	Primary for data exfiltration & C2
The Stealth Factor	Exploiting DNS as a trusted protocol
Advanced Evasion	DNS over HTTPS/TLS (DoH/DoT) tunneling

Fundamental of Zeek



Foundation of Zeek

What is Zeek?

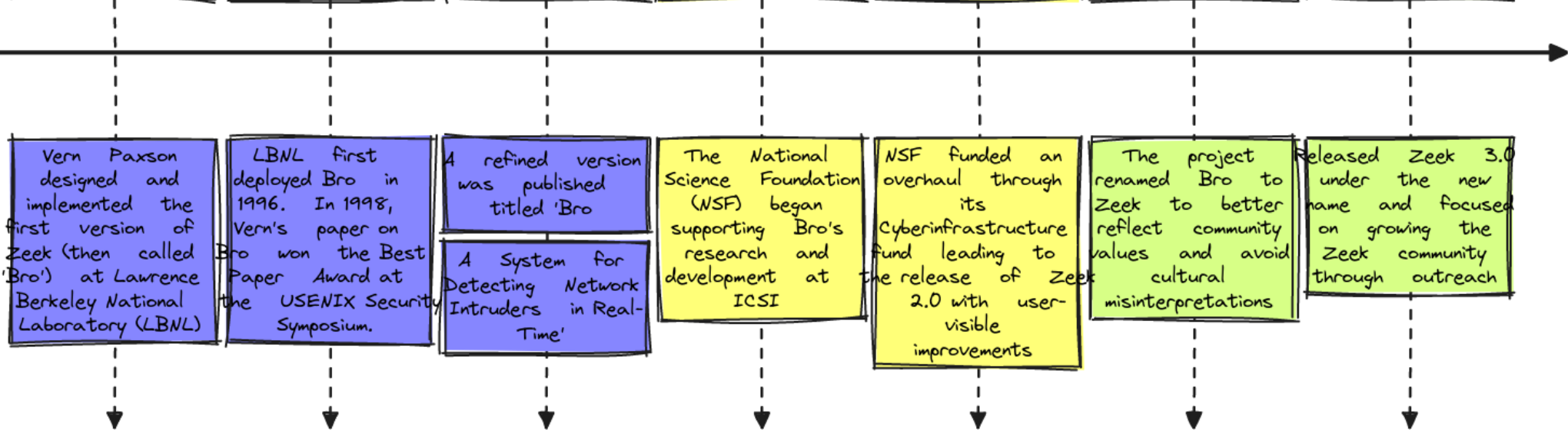
- Zeek is a **network security monitoring** tool that acts as a passive **network analyzer**. It provides deep insights into network traffic and creates **high-fidelity logs** for analysis.
- Often used for **incident response**, **network forensics**, and **threat hunting**.

Why use Zeek?

- Provides detailed analysis of **Layer 7 protocols** like HTTP, DNS, FTP, and SSL/TLS.
- Highly **scriptable** and customizable.
- Generates logs in a standardized format, enabling easy integration with SIEMs & with any Log analyzer.

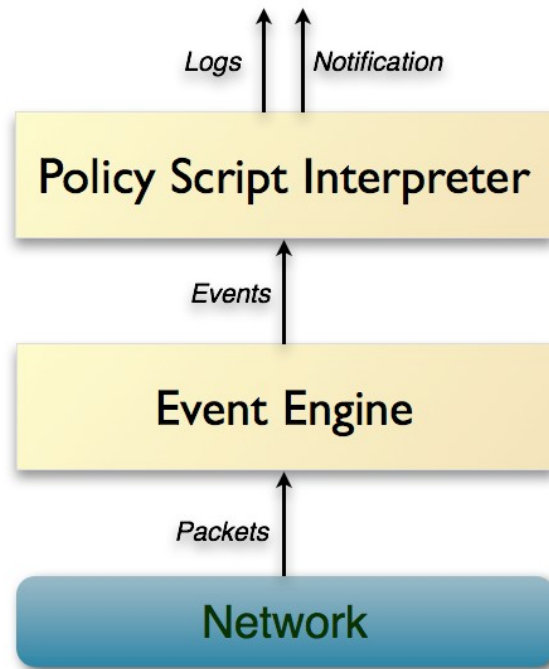
Foundation of Zeek | History

History of Zeek



Foundation of Zeek | Zeek Architecture Overview

- Zeek operates by capturing network traffic from a specified network interface and processing it using an event-driven model.
- It creates logs and can execute custom actions based on network events.

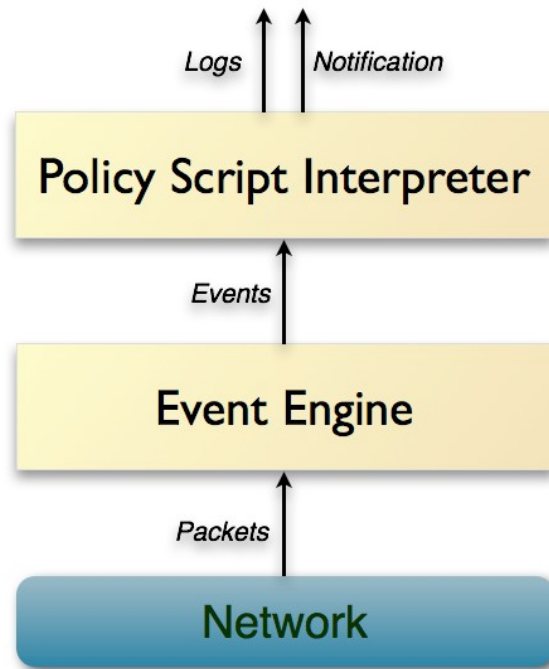


Foundation of Zeek | Zeek Architecture Overview

Event Engine: The Event Engine is responsible for processing raw network traffic and converting it into high-level events.

How it works:

- Receives raw packets from the network.
- Analyzes network protocols (like HTTP, DNS, SSL) and triggers events based on network activities.
- Events represent actions such as an HTTP request, a DNS query, or a new connection.

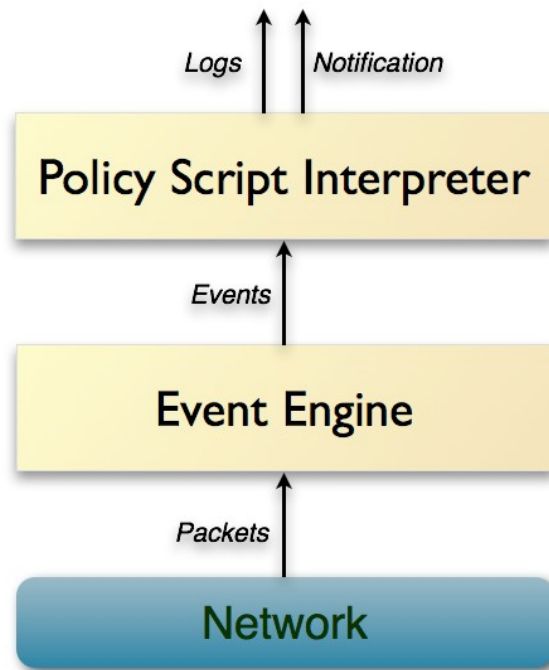


Foundation of Zeek | Zeek Architecture Overview

Policy Script Interpreter: The Policy Script Interpreter uses Zeek scripts to respond to events generated by the Event Engine.

How it works:

- Takes the events created by the Event Engine and executes user-defined scripts.
- Generates logs, alerts, or custom actions based on the scripts.
- Allows users to define custom detection rules, anomaly checks, and analysis workflows.



Foundation of Zeek | Instrumentation and Collection in Zeek

As the primary purpose is Live Traffic Monitoring, placing Zeek, is a strategic call to take.

Basic Setup for Beginners:

- Zeek can be run on a single computer (like a laptop) to monitor its own network traffic.
- Similar to running Tcpdump or Wireshark for educational purposes.

Professional Deployment on a Dedicated Sensor:

- For comprehensive monitoring, Zeek is deployed on a dedicated system or “sensor” located in a key network segment.
- These sensors are carefully chosen to provide maximum visibility into network traffic.

Foundation of Zeek | Instrumentation and Collection in Zeek

Identifying the Best Monitoring Location

Deploying at an Ideal Point:

- Key goal is to find a location where Zeek can see all network traffic with original source IP addresses.
- ***Network TAPs*** or ***SPAN ports*** are recommended for this purpose.

Challenges in Standard SOHO Architectures:

- Limited visibility due to ***NAT (Network Address Translation)*** or shared IP addresses.
- Best locations are often difficult to access.

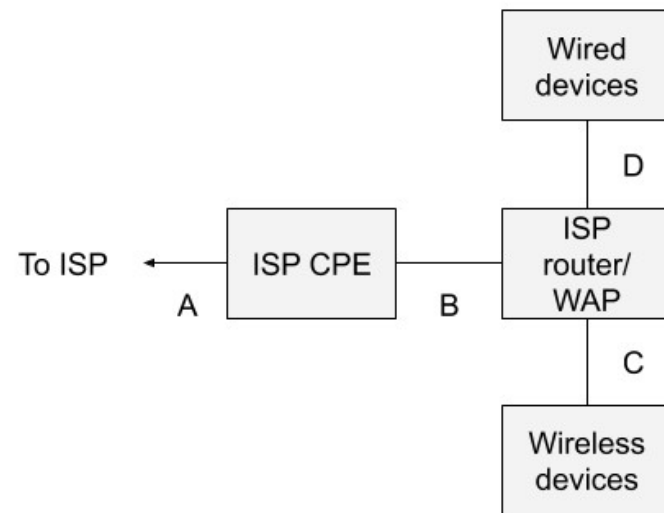
Foundation of Zeek | Instrumentation and Collection in Zeek

Basic SOHO Network Architecture – Limited Visibility

Most home and small office setups use an ISP-provided gateway that acts as both a router and a WiFi Access Point (WAP).

Challenges:

- Location A: Traffic here is inaccessible to the customer due to ISP restrictions.
- Location B: Monitoring possible, but all devices share the same public IP due to NAT.
- Location C: WiFi traffic monitoring is difficult and often not usable.
- Location D: Can monitor wired traffic, but lacks visibility into WiFi.



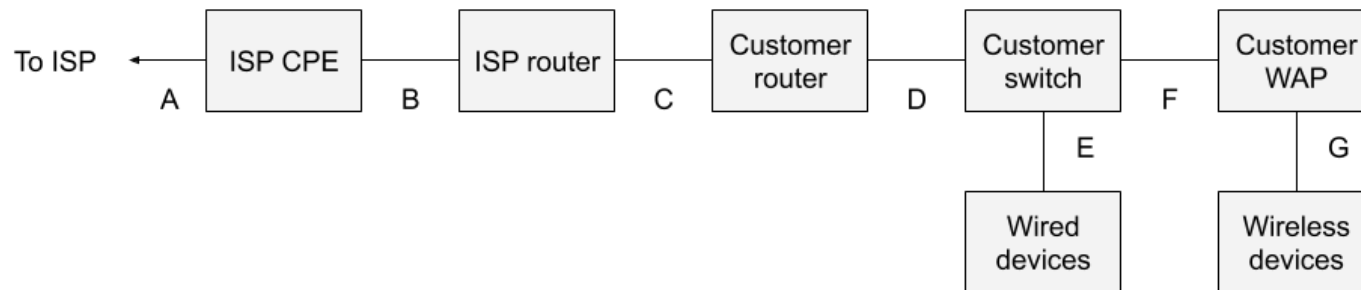
Foundation of Zeek | Instrumentation and Collection in Zeek

Visible Network Architecture – Optimized for Monitoring

This setup separates the WiFi Access Point (WAP) and Router functions and introduces a dedicated switch with a SPAN port.

Improvements:

- Location C: Allows monitoring of both wired and WiFi traffic without NAT interference.
- Location D: Provides comprehensive monitoring when TAPs or SPAN ports are used.
- Eliminated NAT Issues: Clear traffic visibility with original source IPs.



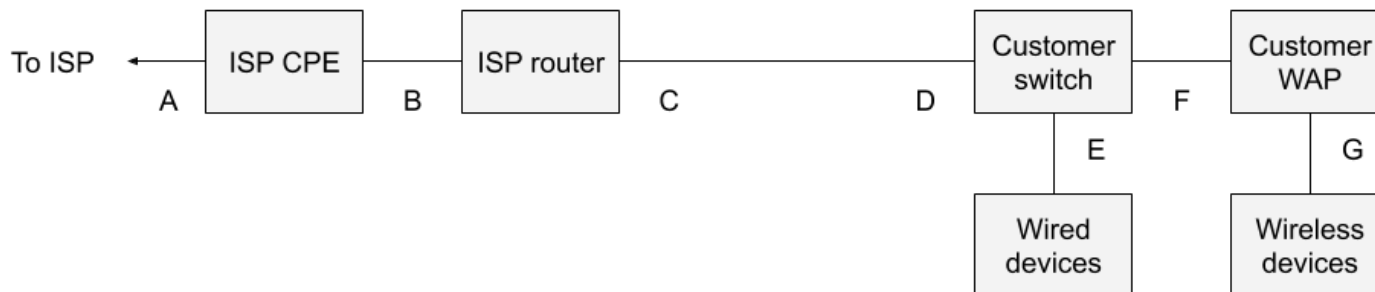
Foundation of Zeek | Instrumentation and Collection in Zeek

Simplified Visible Network Architecture – Streamlined for SOHO

A simplified version of the visible architecture for SOHO (Small Office/Home Office) environments.

Key Features:

- Location C and D: Identified as ideal monitoring points with TAPs or SPAN ports.
- Customer Switch with SPAN Port: Offers affordable monitoring options without complex configurations.
- Eliminated Customer Router: Relies on the ISP router, avoiding additional configurations.



Foundation of Zeek | Instrumentation and Collection in Zeek

Enterprise Network Monitoring

Large enterprise networks are complex, consisting of multiple VLANs, data centers, remote offices, and cloud environments.

Challenges:

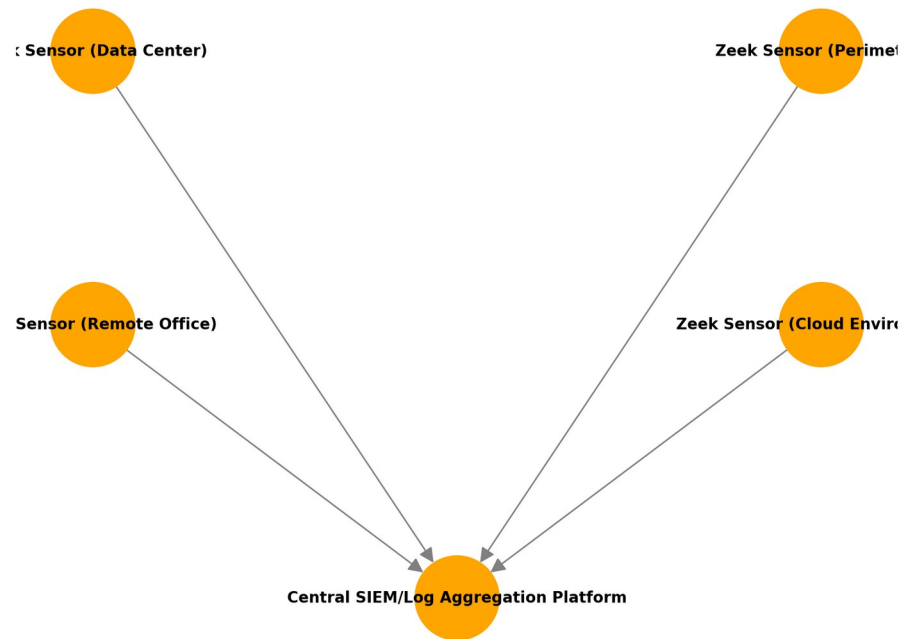
- High volumes of traffic, distributed infrastructure, and numerous network segments.
- The need for centralized visibility across multiple sites and data centers.
- Encryption complicates monitoring in secure environments.

Foundation of Zeek | Instrumentation and Collection in Zeek

Enterprise Network Monitoring

Large enterprise networks are complex, consisting of multiple VLANs, data centers, remote offices, and cloud environments.

Figure: Centralized Threat Analysis with Distributed Zeek Sensors



Foundation of Zeek | Instrumentation and Collection in Zeek

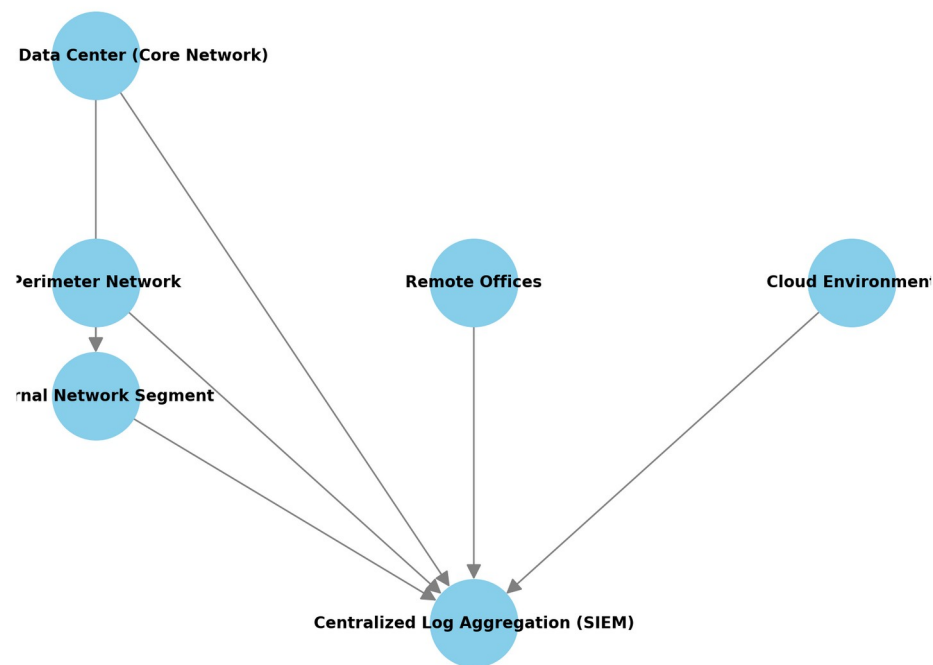
Enterprise Network Monitoring

Use a **distributed deployment of Zeek sensors** across critical network segments to achieve full visibility.

Components:

- **Data Center Sensors:** Deployed at core switches or routers to monitor east-west traffic within the data center.
- **Perimeter Sensors:** Positioned at the internet gateway to monitor inbound and outbound traffic.
- **Remote Office Sensors:** Deployed in major regional or remote offices to monitor branch traffic.
- **Cloud-Based Sensors:** Deployed using cloud-native solutions to monitor cloud workloads.

Figure: Distributed Zeek Deployment in an Enterprise Network



Foundation of Zeek | Instrumentation and Collection in Zeek

Enterprise Network Monitoring

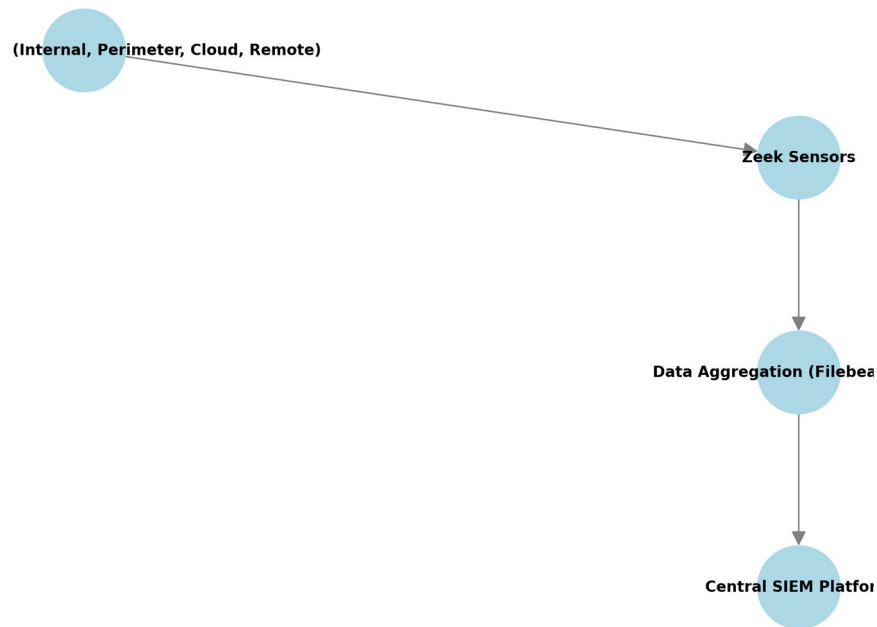
Data Aggregation:

- Aggregate logs from all distributed Zeek sensors into a central Security Information and Event Management (SIEM) system like Graylog or Elastic Stack.
- Use Filebeat or Kafka to stream logs from remote sensors to the central analysis platform.

Centralized Threat Analysis:

- Correlate logs from different segments to detect lateral movement, unauthorized access, and data exfiltration.
- Integrate with Threat Intelligence Feeds to enrich Zeek logs and create automated alerts.

Figure: Data Flow and Central Aggregation in Enterprise Network



Foundation of Zeek

Foundation of Zeek | **installation & configuration**

Installing Zeek in Ubuntu 22.04 server LTS edition

Installation Requirements

- Ubuntu 22.04 (preferred), 2 CPU Cores, 4GB RAM minimum, 20GB storage.

Update the package manager:

```
sudo apt update && sudo apt upgrade -y
```

Install Dependencies:

```
sudo apt-get install cmake make gcc g++ flex bison libpcap-dev libssl-dev python3  
python3-dev swig zlib1g-dev -y
```

Foundation of Zeek

Foundation of Zeek | **installation & configuration**

Installing Zeek in Ubuntu 22.04 server LTS edition with ***Zeek 6.0 LTS release***

```
$ echo 'deb
http://download.opensuse.org/repositories/security:/zeek/xUbuntu_22.04/ /' | sudo
tee /etc/apt/sources.list.d/security:zeek.list

$ curl -fsSL
https://download.opensuse.org/repositories/security:zeek/xUbuntu_22.04/Release.key
| gpg --dearmor | sudo tee /etc/apt/trusted.gpg.d/security_zeek.gpg > /dev/null

$ sudo apt update

$ sudo apt install zeek-6.0
```

Foundation of Zeek

Foundation of Zeek | **installation & configuration**

Once the Zeek is installed, add Zeek path to **.bashrc** file to complete the environment variables.

Reload the .bashrc file using the following command, and verify.

```
$ sudo su
# echo "export PATH=$PATH:/opt/zeek/bin" >> ~/.bashrc
# source ~/.bashrc
# zeek --version
```


Foundation of Zeek

Basic Configuration

- Zeek Installation Directory: ***/opt/zeek/etc***
- Configuration Files: Key files such as ***node.cfg***, ***zeekctl.cfg***, and ***network.cfg***

```
shamim@labmainsrv:/opt/zeek/etc$ ls -lah
total 24K
drwxrwsr-x  3 root zeek 4.0K Oct 26 02:15 .
drwxr-xr-x 10 root root 4.0K Oct 24 07:58 ..
-rw-rw-r--  1 root zeek  436 Jan 28  2015 networks.cfg
-rw-rw-r--  1 root zeek   653 Oct 24 10:15 node.cfg
-rw-rw-r--  1 root zeek  3.0K Oct 26 02:11 zeekctl.cfg
drwxr-xr-x  2 root zeek 4.0K Oct 24 07:58 zkg
shamim@labmainsrv:/opt/zeek/etc$
```

Foundation of Zeek | Overview of Zeek logs

Types of Zeek Logs and their Purposes

Log type	Definition
conn.log	Records all connection-level traffic with essential metadata like IPs, ports, bytes transferred, and connection state.
dns.log	Logs DNS requests and responses, including domain names queried and associated metadata.
http.log	Records HTTP requests and responses, including URLs, HTTP methods, response codes, and User-Agent strings.
ssl.log	Captures SSL/TLS handshake details such as certificates, issuers, and SSL version.
files.log	Logs metadata for files detected during network activity.

Foundation of Zeek | Overview of Zeek logs

Sample conn.log file content.

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	missed_bytes	history	orig_pkts	orig_ip_bytes	resp_pkts	resp_ip_bytes	tunnel_id
time	string	addr	port	addr	port	enum	string	interval	count	count	string	bool	bool	count	string	count	count	count	count	set[str]
1732806000.54918	Cyt6v94ZHY6G8eGQUh	192.168.68.106	57694	203.76.96.5	53	udp	dns	0.02021	0	422	SHR	T	F	0Cd	0	0	4	534	-	
1732806000.5685	CWT91B2QiHqmLFekZ	192.168.68.106	39319	203.76.96.5	53	udp	dns	0.07917	0	422	SHR	T	F	0Cd	0	0	4	534	-	
1732806000.65947	CiW2be4H8hrMynJwi	192.168.68.106	39853	203.76.96.5	53	udp	dns	0.094479	0	414	SHR	T	F	0Cd	0	0	4	526	-	
1732806000.77221	CuViEutxmRyAM4ajc	192.168.68.106	41374	203.76.96.5	53	udp	dns	0.006025	0	434	SHR	T	F	0Cd	0	0	4	546	-	
1732806000.80795	Ccc0gt3YL18z0Rq9Od	192.168.68.106	40851	203.76.96.5	53	udp	dns	0.660426	0	638	SHR	T	F	0Cd	0	0	4	750	-	
1732806001.49831	Czy2vj4GLdptutraVc	192.168.68.106	36892	203.76.96.5	53	udp	dns	0.073941	0	422	SHR	T	F	0Cd	0	0	4	534	-	
1732806001.61443	Ckk4JTyDq4KaxoXx	192.168.68.106	55478	203.76.96.5	53	udp	dns	0.57137	0	638	SHR	T	F	0Cd	0	0	4	750	-	
1732806002.20475	CXkleX2Lat0TQV5e98	192.168.68.106	55095	203.76.96.5	53	udp	dns	0.012141	0	638	SHR	T	F	0Cd	0	0	4	750	-	
1732806002.22593	COlzWt360ImeldYGla	192.168.68.106	50634	203.76.96.5	53	udp	dns	0.00236	0	688	SHR	T	F	0Cd	0	0	2	744	-	
1732805954.50547	CXqHeDlvQFtErRntvg	192.168.68.106	38345	185.125.190.58	123	udp	ntp	0.265767	0	96	SHR	T	F	0Cd	0	0	2	152	-	
1732806000.5694	CD8itT2sauOpWYnJg	192.168.68.106	3	203.76.96.5	3	icmp	-	1.002894	892	0	QTH	T	F	0-	7	1088	0	0	-	
1732806015.67433	Cnl0fG3A7BP0ev8V4f	192.168.68.107	58561	239.255.255.250	1900	udp	-	3.002417	688	0	SO	T	F	0D	4	800	0	0	-	
1732805969.0042	C2MfsWycwa8BUox7i	192.168.68.1	8	192.168.68.106	0	icmp	-	59.491223	32	32	QTH	T	T	0-	4	144	4	144	-	
1732806074.36666	Cf7yXuld8jeuAVsKz6	192.168.68.1	42004	192.168.68.255	20002	udp	-	-	-	-	SO	T	T	0	Dl	943	0	0	-	
1732806135.67524	C6xAQhsbPufISpe7	192.168.68.107	34189	239.255.255.250	1900	udp	-	3.003146	688	0	SO	T	F	0D	4	800	0	0	-	
1732806089.01876	Cwr2dx4R5kCGQw2qCF7	192.168.68.1	8	192.168.68.106	0	icmp	-	59.48415	32	32	QTH	T	T	0-	4	144	4	144	-	
1732806255.67536	CgmJv0tkzD9uQd0HBk	192.168.68.107	35443	239.255.255.250	1900	udp	-	3.004	688	0	SO	T	F	0D	4	800	0	0	-	
1732806209.01789	CYp72pFeGpr06inV	192.168.68.1	8	192.168.68.106	0	icmp	-	59.391124	32	32	QTH	T	T	0-	4	144	4	144	-	

Foundation of Zeek | Overview of Zeek logs

Understanding Log Content with Examples

Fields Explanation:

- ***ts:*** Timestamp of the event.
- ***id.orig_h*** and ***id.resp_h:*** IPs of the origin and destination hosts.
- ***proto:*** Protocol (TCP, UDP, etc.)
- ***service:*** Service detected on the port (like HTTP, SSL, DNS, etc.)
- ***history:*** Flags representing the state of the connection.

The conn.log primarily captures so-called “layer 3” and “layer 4” elements of network activity.

Foundation of Zeek | Zeek log navigation

If you want to read Zeek log in JSON format, follow the following configuration.

```
root@labmainsrv:/opt/zeek/  
logs/current# jq . dns.log
```

```
"ts":1729773278.719612,  
"uid":"C9OyM122V1CF7BrVZi",  
"id.orig_h":"192.168.68.106",  
"id.orig_p":54462,  
"id.resp_h":"203.76.96.5",  
"id.resp_p":53,  
"proto":"udp",  
"trans_id":60555,  
"query":"theteamphoenix.org",  
"rcode":0,  
"rcode_name":"NOERROR",  
"AA":false,  
"TC":false,  
"RD":false,  
"RA":true,  
"Z":0,  
"answers":["63.250.43.15","63.250.43.16"],  
"TTLS":[60.0,60.0],  
"rejected":false}
```

Foundation of Zeek | Zeek log navigation

Checking timestamp from logs.

```
# date -d @"1729773278.719612"
```

Thu Oct 24 12:34:38 PM UTC 2024

```
"ts":1729773278.719612,  
"uid":"C9OyM122V1CF7BrVZi",  
"id.orig_h":"192.168.68.106",  
"id.orig_p":54462,  
"id.resp_h":"203.76.96.5",  
"id.resp_p":53,  
"proto":"udp",  
"trans_id":60555,  
"query":"theteamphoenix.org",  
"rcode":0,  
"rcode_name":"NOERROR",  
"AA":false,  
"TC":false,  
"RD":false,  
"RA":true,  
"Z":0,  
"answers":["63.250.43.15","63.250.43.16"],  
"TTLS":[60.0,60.0],  
"rejected":false}
```

Foundation of Zeek | Zeek log navigation

Navigating Zeek log with Basic CLI tools.

View the beginning of the log files.

```
root@labmainsrv:/opt/zeek/logs/current# head -n 2 ssh.log
```

```
{"ts":1729775010.25454,"uid":"CdAZ7X1SZx3m3vwmA2","id.orig_h":"192.168.68.107","id.orig_p":42918,"id.resp_h":"192.168.68.106","id.resp_p":22,"auth_attempts":0,"client":"SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.5"}
```

```
{"ts":1729775887.920954,"uid":"CvyqSs2sq9fevEhngi","id.orig_h":"192.168.68.107","id.orig_p":38750,"id.resp_h":"192.168.68.106","id.resp_p":22,"auth_attempts":0,"client":"SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.5"}
```

Foundation of Zeek | Zeek log navigation

Navigating Zeek log with Basic CLI tools.

View the beginning of the log files.

```
root@labmainsrv:/opt/zeek/logs/current# grep "192.168.68.107" ssh.log
```

```
{"ts":1729775010.25454,"uid":"CdAZ7X1SZx3m3vwmA2","id.orig_h":"192.168.68.107","id.orig_p":42918,"id.resp_h":"192.168.68.106","id.resp_p":22,"auth_attempts":0,"client":"SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.5"}
```

```
{"ts":1729775887.920954,"uid":"CvyqSs2sq9fevEhngi","id.orig_h":"192.168.68.107","id.orig_p":38750,"id.resp_h":"192.168.68.106","id.resp_p":22,"auth_attempts":0,"client":"SSH-2.0-OpenSSH_9.6p1 Ubuntu-3ubuntu13.5"}
```


Foundation of Zeek | Zeek log navigation

Navigating Zeek log with Basic CLI tools.

View the beginning of the log files.

```
root@labmainsrv:/opt/zeek/logs/current# grep -c "192.168.68.107" ssh.log
```

4

Foundation of Zeek | Zeek log navigation

Filtering Data with **zeek-cut** and **awk**

The zeek-cut tool helps filter out specific columns from Zeek logs quickly.

```
root@labmainsrv:~# cat http.log | zeek-cut ts id.orig_h id.resp_h uri referrer  
1729777456.179294      192.168.68.106      202.12.29.1      -      -
```

This command extracts timestamps, origin IP, destination IP, URI, and referrer fields from http.log

Foundation of Zeek | Zeek log navigation

Using **awk** for Custom Filtering | Display only specific Columns, this extracts the first, third, and fifth columns from **conn.log**; which is timestamp, source IP and response IP.

```
root@labmainsrv:# awk '{print $1, $3, $5}' conn.log
```

```
1729777439.136097 192.168.68.106 203.76.96.145
1729777443.074588 192.168.68.106 63.250.43.16
1729777439.139549 192.168.68.106 203.76.96.145
1729777439.126408 192.168.68.106 123.200.0.254
1729777443.379825 192.168.68.106 63.250.43.16
1729777451.881253 192.168.68.106 202.12.29.1
1729777442.203939 192.168.68.106 123.200.0.254
1729777442.204311 192.168.68.106 123.200.0.254
1729777455.771767 192.168.68.106 202.12.29.1
1729777452.083491 192.168.68.106 202.12.29.1
1729777455.976419 192.168.68.106 202.12.29.1
1729777451.717110 192.168.68.106 123.200.0.254
1729777451.717311 192.168.68.106 123.200.0.254
1729777430.681661 fe80::4ccd:53ff:fe2a:33b ff02::fb
```

Foundation of Zeek | Zeek log navigation

Filter connections based on duration; this command filters and displays connections lasting longer than 60 seconds.

```
root@labmainsrv:~# awk '$9 > 60 {print $0}' conn.log
```

#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	missed_bytes	history	orig_pkts				
orig_ip_bytes	resp_pkts	resp_ip_bytes	tunnel_parents									
#types	time	string	addr	port	addr	port	enum	string	interval	count	count	string
bool	bool	count	string	count	count	count	count	set[string]				
1729777573.115258			COgivf47idEwTbjI5i	192.168.68.104	5353	224.0.0.251	5353	udp	dns	150.731054		
14280	0	S0	T	F	0D	96	16968	0	0	-		
1729777573.115638			CrupzO1KrfX6UNEWGa	fe80::4ccd:53ff:fe2a:33b	5353	ff02::fb	5353	udp	dns			
150.730675	14199	0	S0	TF	0	D	95	18759	0	0	-	
1729777739.615904			Cw9ZDJ2sjmQdg7LDIh	fe80::4ccd:53ff:fe2a:33b	5353	ff02::fb	5353	udp	dns			
107.007402	10531	0	S0	TF	0	D	78	14275	0	0	-	
1729777739.615336			CPZd0u1eBIzA6j08e6	192.168.68.104	5353	224.0.0.251	5353	udp	dns	107.007432		
10531	0	S0	T	F	0D	78	12715	0	0	-		

Foundation of Zeek | Zeek log navigation

In this part, we'll analyze ***dns.log*** to explore DNS queries and responses.; ***Identify Top Queried Domains:***

```
root@labmainsrv:~# cat dns.log | zeek-cut query | sort | uniq -c | sort -nr | head -n 10
```

```
269 android.local
```

```
40 -
```

```
9 _googlecast._tcp.local
```

```
8 _googlezone._tcp.local
```

```
6 7ad23c7c-995c-46f3-245c-ac12ed9e27eb.local
```

```
4 reqbin.com
```

```
3 google-nest-mini-7ad23c7c995c46f3245cac12ed9e27eb._googlecast._tcp.local
```

```
2 _ipps._tcp.local
```

```
2 7ad23c7c-995c-46f3-245c-ac12ed9e27eb._googlezone._tcp.local
```

```
2 254.0.200.123.in-addr.arpa
```

Foundation of Zeek | Zeek log navigation

In this part, we'll analyze ***dns.log*** to explore DNS queries and responses.; ***Extract and Count Response Codes:***

```
root@labmainsrv:~# cat dns.log | zeek-cut rcode | sort | uniq -c | sort -nr
```

```
410 0
```

```
40 3
```

```
11 -
```

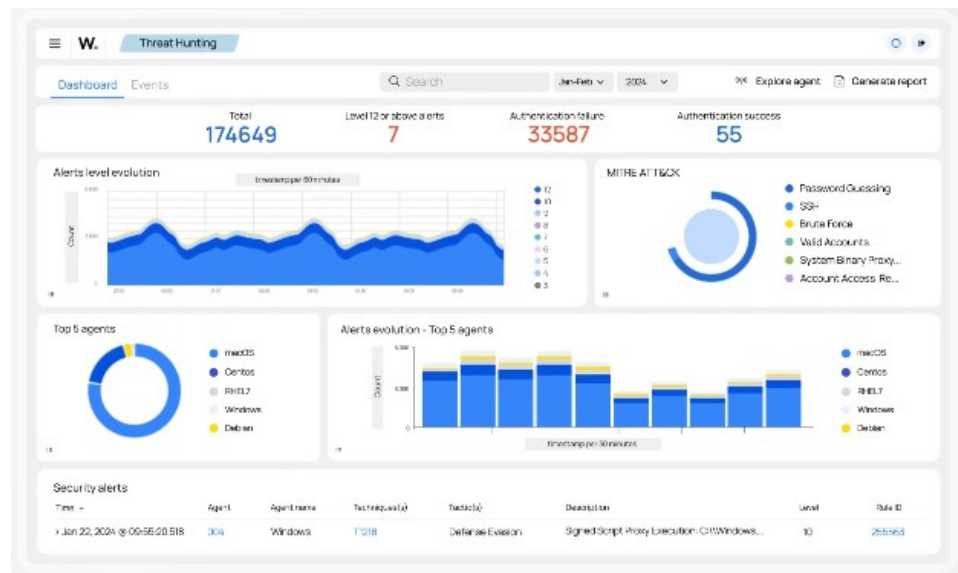
Foundation of Zeek | Zeek log navigation

In this part, we'll analyze dns.log to explore DNS queries and responses.; ***Extract all unique domains from dns.log***

```
root@labmainsrv:~# awk '{ print $10 }' dns.log | sort | uniq
```

```
apnic.net  
_ewelink._tcp.local  
gmail.com  
link3.net  
prothomalo.com  
reqbin.com  
rtt  
theteamphoenix.org  
yahoo.com
```

Fundamental of Wazuh



Foundation of Wazuh

The Open Source Security Platform provides unified XDR and SIEM protection for endpoints and cloud workloads.



15+ Million
Protected endpoints



100+ Thousand
Enterprise users



30+ Million
Downloads per year

Endpoint Security

- Configuration Assessment
- Malware Detection
- File Integrity Monitoring

Threat Intelligence

- Threat Hunting
- Log Data Analysis
- Vulnerability Detection

Security Operations

- Incident Response
- Regulatory Compliance
- IT Hygiene

Cloud Security

- Container Security
- Posture Management
- Workload Protection

Wazuh Installation

Installing Wazuh in Ubuntu 22.04 server LTS edition

Installation Requirements

- Ubuntu 22.04 (preferred), 2 CPU Cores, 8GB RAM minimum, 20GB storage.

Update the package manager:

```
sudo apt update && sudo apt upgrade -y
```

Install Dependencies:

```
sudo apt install vim curl apt-transport-https unzip wget libcap2-bin software-properties-common lsb-release gnupg2
```

Wazuh Installation

Installing Wazuh in Ubuntu 22.04 server LTS edition with BASH script.

Installation Requirements

- Ubuntu 22.04 (preferred), 2 CPU Cores, 8GB RAM minimum, 20GB storage.

Download the Script:

```
curl -sO https://packages.wazuh.com/4.7/wazuh-install.sh
```

Once the script is downloaded, run it accordingly:

```
sudo bash ./wazuh-install.sh -a
```

Wazuh Installation

```
shamim@srv-zeek:~/wazuh$ sudo bash ./wazuh-install.sh -a
22/08/2025 23:50:15 INFO: Starting Wazuh installation assistant. Wazuh version: 4.7.5
22/08/2025 23:50:15 INFO: Verbose logging redirected to /var/log/wazuh-install.log
22/08/2025 23:50:39 INFO: Wazuh web interface port will be 443.
22/08/2025 23:50:50 INFO: Wazuh repository added.
22/08/2025 23:50:50 INFO: --- Configuration files ---
22/08/2025 23:50:50 INFO: Generating configuration files.
22/08/2025 23:50:51 INFO: Created wazuh-install-files.tar. It contains the Wazuh cluster key, certificates, and passwords necessary for installation.
22/08/2025 23:50:51 INFO: --- Wazuh indexer ---
22/08/2025 23:50:51 INFO: Starting Wazuh indexer installation.
23/08/2025 00:08:08 INFO: Wazuh indexer installation finished.
23/08/2025 00:08:08 INFO: Wazuh indexer post-install configuration finished.
23/08/2025 00:08:08 INFO: Starting service wazuh-indexer.
23/08/2025 00:08:25 INFO: wazuh-indexer service started.
23/08/2025 00:08:25 INFO: Initializing Wazuh indexer cluster security settings.
23/08/2025 00:08:37 INFO: Wazuh indexer cluster initialized.
23/08/2025 00:08:37 INFO: --- Wazuh server ---
23/08/2025 00:08:37 INFO: Starting the Wazuh manager installation.
23/08/2025 00:15:28 INFO: Wazuh manager installation finished.
23/08/2025 00:15:28 INFO: Starting service wazuh-manager.
23/08/2025 00:15:52 INFO: wazuh-manager service started.
23/08/2025 00:15:52 INFO: Starting Filebeat installation.
23/08/2025 00:16:59 INFO: Filebeat installation finished.
23/08/2025 00:17:02 INFO: Filebeat post-install configuration finished.
23/08/2025 00:17:02 INFO: Starting service filebeat.
23/08/2025 00:17:08 INFO: filebeat service started.
23/08/2025 00:17:08 INFO: --- Wazuh dashboard ---
23/08/2025 00:17:08 INFO: Starting Wazuh dashboard installation.
23/08/2025 00:32:12 INFO: Wazuh dashboard installation finished.
23/08/2025 00:32:12 INFO: Wazuh dashboard post-install configuration finished.
23/08/2025 00:32:12 INFO: Starting service wazuh-dashboard.
23/08/2025 00:32:15 INFO: wazuh-dashboard service started.
23/08/2025 00:32:43 INFO: Initializing Wazuh dashboard web application.
23/08/2025 00:32:44 INFO: Wazuh dashboard web application initialized.
23/08/2025 00:32:44 INFO: --- Summary ---
23/08/2025 00:32:44 INFO: You can access the web interface https://<wazuh-dashboard-ip>:443
    User: admin
    Password: P3M0*v+wcFVuv6gr*H3vTlaPYqqWUXnk
23/08/2025 00:32:44 INFO: Installation finished.
shamim@srv-zeek:~/wazuh$
```

Wazuh Installation

Not secure

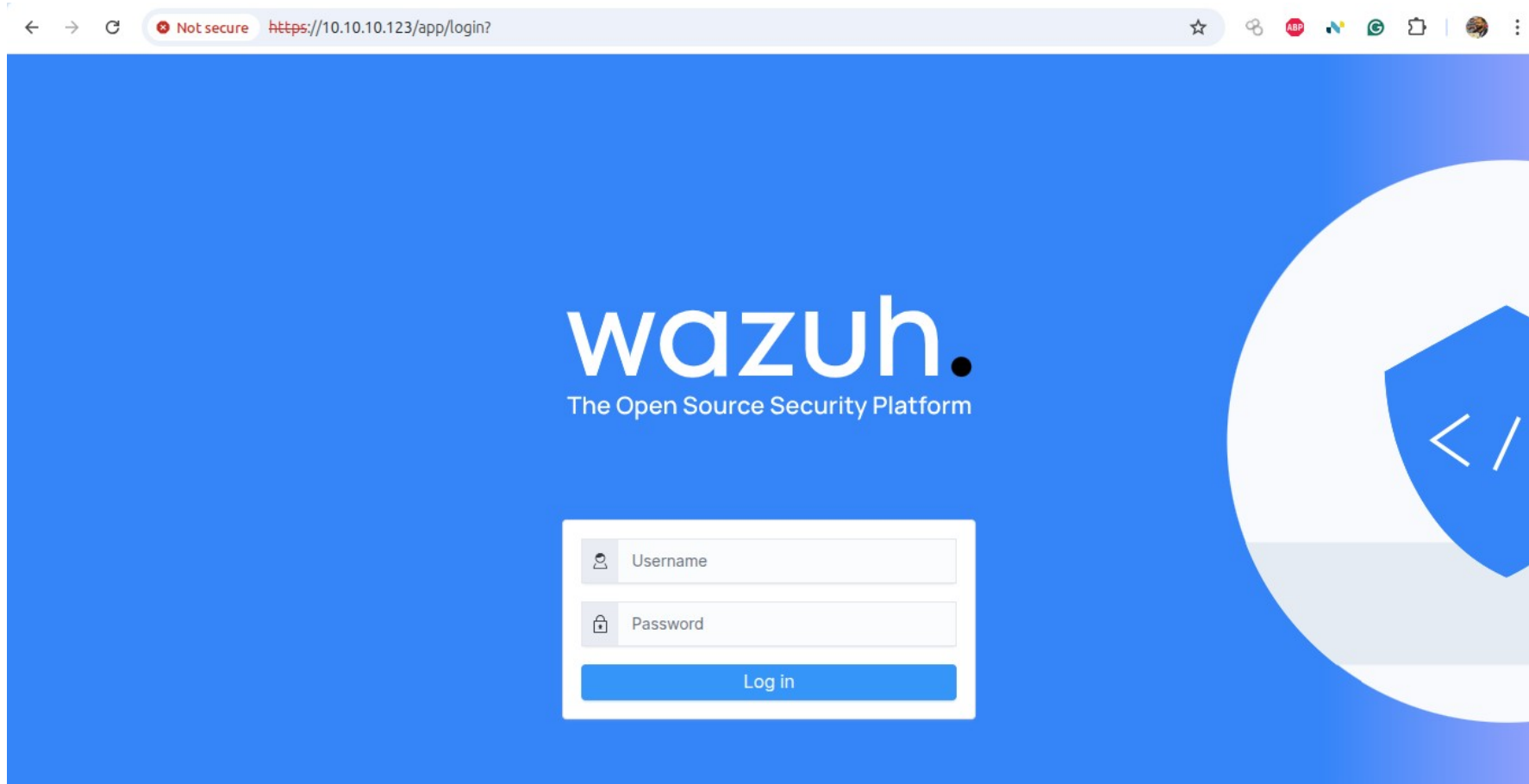
https://10.10.10.123/app/login?



wazuh.

Loading ...

Wazuh Installation



Wazuh Installation

← → ↻ Not secure https://10.10.10.123/app/wazuh#/overview/?_g=(filters:!),refreshInterval:(pause:1t,value:0),time:(from:now-24h,to:now))&... 🔍 ☆

☰ 🏠 wazuh. ▾ Modules a ⓘ

Total agents
0

Active agents
0


Disconnected agents
0


Pending agents
0

Never connected agents
0


⚠ No agents were added to this manager. [Add agent](#)


SECURITY INFORMATION MANAGEMENT


**Security events**
Browse through your security alerts, identifying issues and threats in your environment.

**Integrity monitoring**
Alerts related to file changes, including permissions, content, ownership and attributes.


AUDITING AND POLICY MONITORING


**Policy monitoring**
Verify that your systems are configured according to your security policies baseline.

**System auditing**
Audit users behavior, monitoring command execution and alerting on access to critical files.


**Security configuration assessment**
Scan your assets as part of a configuration assessment audit.


THREAT DETECTION AND RESPONSE


**Vulnerabilities**
Discover what applications in your environment are affected by well-known vulnerabilities.


**MITRE ATT&CK**
Security events from the knowledge base of adversary tactics and techniques based on real-world observations


REGULATORY COMPLIANCE

**PCI DSS**
Global security standard for entities that process, store or transmit payment cardholder data.

**NIST 800-53**
National Institute of Standards and Technology Special Publication 800-53 (NIST 800-53) sets guidelines for federal information systems.

**TSC**
Trust Services Criteria for Security, Availability, Processing Integrity, Confidentiality, and Privacy

**GDPR**
General Data Protection Regulation (GDPR) sets guidelines for processing of personal data.

**HIPAA**

The Detection Blueprint

Hypothesis to detect attacks.

"

If an internal host or set of internal hosts is infected with malware using a Domain Generation Algorithm (DGA) for Command and Control (C2) communication, then the DNS logs will show a high frequency of unique and often random-looking domain names queried by these hosts, accompanied by a high rate of NXDOMAIN responses, and anomalous patterns in domain entropy and character length."

The Detection Blueprint

Key Characteristics to Look for in DGA-Related DNS Traffic

To refine this hypothesis and provide a more structured approach to your investigation, consider these specific behavioral traits and metrics:

- **High Volume of Unique Domain Queries:** DGA-based malware generates numerous domain names within a short time window. An infected host will attempt to resolve many unique domains to connect with the active C2 server.
 - **Metric:** High count of unique domain names queried from a single host in a given time period.
- **Frequent NXDOMAIN Responses:** Since most of the generated domains are not registered, a high number of DNS queries will result in NXDOMAIN (non-existent domain) responses from the DNS server.
 - **Metric:** High proportion of NXDOMAIN responses relative to other response codes from a single host.
- **Domain Name Entropy and Length:** DGA-generated domain names often have a higher randomness and length compared to legitimate domain names. Analyzing entropy and domain length can reveal domains that are likely algorithmically generated.
 - **Metric:** Domains with high entropy values or longer-than-average lengths.
- **Time-based Patterns of Queries:** Malware employing DGA techniques typically attempts to resolve domains at regular or near-regular intervals, indicating automated behavior.
 - **Metric:** Patterns of DNS queries that occur at predictable time intervals.

The Detection Blueprint

Let us start the hunt.

1. Frequency Analysis of Queried Domains

```
# cat dns.log | zeek-cut query | sort | uniq -c | sort -nr | head -n 10
```

This command extracts the queried domains, counts their occurrences, and sorts them in descending order. Unusually high numbers of unique queries could indicate DGA-based behavior.

2. Identify High NXDOMAIN Rates

```
# cat dns.log | zeek-cut query rcode_name | grep "NXDOMAIN" | sort | uniq -c | sort -nr | head -n 10
```

This command extracts domains that resulted in an NXDOMAIN response. A high number of these responses from a single source can indicate that the host is attempting to resolve numerous non-existent domains, a common DGA tactic.

The Detection Blueprint

Let us start the hunt.

3. Filter Domains by Length

```
# cat dns.log | zeek-cut query | awk 'length($1) > 12' | sort | uniq -c | sort -nr | head -n 10
```

This filters out domains longer than 12 characters, which are often generated by DGA algorithms. You can adjust the threshold length based on typical domain patterns in your environment.

4. Entropy Analysis for Domain Names

```
# cat dns.log | zeek-cut query | awk '{print $1}' | while read domain; do
    entropy=$(echo -n "$domain" | awk -v OFS="" '{for (i=1; i<=length; i++) { freq[substr($0, i, 1)]++; } for (i in freq) { p=freq[i]/length; sum+=-p*log(p)/log(2); } print sum }' );
    echo "$entropy $domain";
done | sort -nr | head -n 10
```

This script calculates the entropy for each domain and sorts them in descending order. Domains with high entropy are more likely to be DGA generated.

The Detection Blueprint

Let us start the hunt.

5. Identify anomalous host

```
# cat dns.log | zeek-cut id.orig_h query | sort | uniq -c | awk '$1 > 50 {print $0}'
```

This command identifies hosts that have queried more than 50 unique domains, which is an indicator of potentially automated querying behavior, typical in DGA attacks

“Threat hunting & Detection isn't just about finding malicious actors; it's about understanding and mastering your network better than they do.



Reference

- <https://www.techradar.com/pro/security/starbucks-has-gone-back-to-pen-and-paper-after-vendor-ransomware-attack>
- <https://www.blackberry.com/us/en/solutions/endpoint-security/mitre-attack/mitre-attack-vs-cyber-r-kill-chain>
- <https://attack.mitre.org/>
- <https://docs.zEEK.org/en/master/scripting/basics.html>
- <https://docs.zEEK.org/en/master/scripting/basics.html#data-types-and-data-structures>
- <https://www.cybercrowd.co.uk/news/mandiant-m-trends-2023-threat-intelligence-what-do-you-need-to-know/>
- <https://www.oreilly.com/library/view/the-practice-of/9781457185175/>
- https://www.splunk.com/en_us/form/10-ways-to-take-the-mitre-att-and-ck-framework-from-plan-to-action.html
- <https://www.crowdstrike.com/en-us/resources/reports/global-threat-report-executive-summary-2023/>
- <https://www.crowdstrike.com/en-us/resources/reports/global-threat-report-executive-summary-2024/>
- <https://www.paloaltonetworks.com/resources/research/unit-42-incident-response-report>
- <https://blog.checkpoint.com/research/a-closer-look-at-q3-2024-75-surge-in-cyber-attacks-worldwide/>
- <https://www.ibm.com/reports/data-breach>
- <https://darktrace.com/blog/breaking-down-nation-state-attacks-on-supply-chains#:~:text=Threat%20actors%20tied%20to%20Russian,as%20several%20major%20tech%20companies>
- <https://www.nytimes.com/2021/05/08/us/politics/cyberattack-colonial-pipeline.html>
- <https://documentation.wazuh.com/current/getting-started/components/index.html>

Thank you



TheTeamPhoenix
Machine Secures Machines