TABLE OF CONTENTS

APT Detection & Threat Intelligence Platform Documentation

Documentation for the adAPT Threat Detection Framework.

System Hardware & Resource Allocation

Host Hardware

Machine: ThinkPad

Resources: 8 Cores / 16GB RAM

VM Resource Allocation

Tailscale

pfSense

Ubuntu Server

1 Core / 512MB

2 Cores / 2GiB

4 Cores / 4GiB

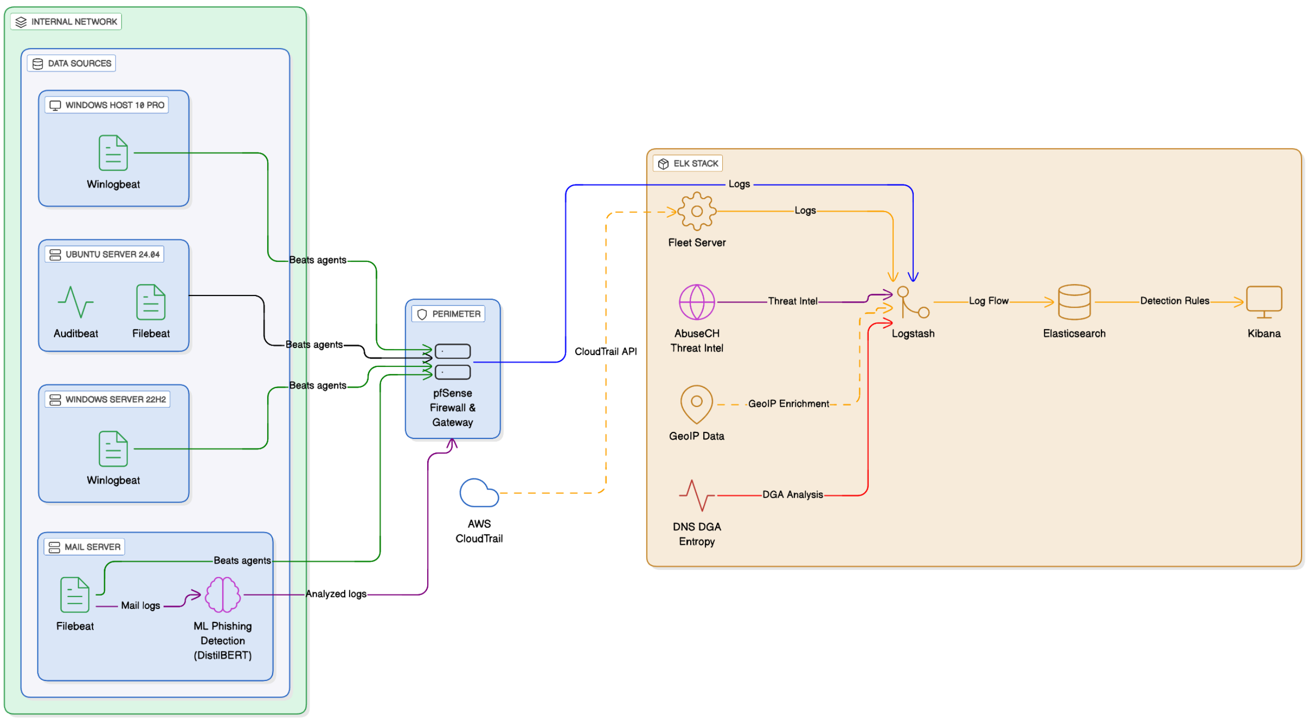
ELK Server

4 Cores / 6GiB

Windows Server

4 Cores / 4GiB

System Architecture



The diagram above illustrates the data flow from various sources in the internal network, through the perimeter firewall, and into the ELK Stack for analysis and alerting.

Part 1: Log Collection Infrastructure

Overview

This platform collects logs from multiple sources across Windows, Ubuntu, and AWS environments using Elastic Beats agents. All logs flow through a perimeter gateway before reaching the ELK stack.

1.1 Windows Host VM Setup

System Requirements

OS: Windows 10/11 or Windows Server 2016+
RAM: Minimum 4GB
Disk: 50GB available space
Network: Access to perimeter gateway on port 5044

Winlogbeat Installation

```
# Download Winlogbeat
Invoke-WebRequest -Uri https://artifacts.elastic.co/downloads/beats/winlogbeat/winlog
Expand-Archive winlogbeat.zip -DestinationPath "C:\Program Files\"
cd "C:\Program Files\winlogbeat-8.17.0-windows-x86_64"

# Install as service
.\install-service-winlogbeat.ps1

# Start service
Start-Service winlogbeat
```

Configuration (winlogbeat.yml)

```
winlogbeat.event_logs:
  - name: Application
  - name: Security
  - name: System
  - name: Microsoft-Windows-Sysmon/Operational
  - name: Windows PowerShell
  - name: Microsoft-Windows-PowerShell/Operational

output.logstash:
  hosts: [":5044"]
```

Logs Collected from Windows Host

- Security Events: Authentication, account management, privilege escalation
- Sysmon Events: Process creation, network connections, file modifications
- PowerShell Logs: Script block logging, command execution
- Application/System: Service failures, application crashes

1.2 Windows Server VM Setup

System Requirements

Same as Windows Host (Section 1.1)

Installation Steps

Follow the same Winlogbeat installation process. Add these additional event logs to winlogbeat.yml:

```
winlogbeat.event_logs:
  # ... (previous logs)
  - name: Microsoft-Windows-DNS-Server/Analytical
  - name: Microsoft-Windows-TerminalServices-LocalSessionManager/Operational
  - name: Microsoft-Windows-TaskScheduler/Operational
```

Logs Collected from Windows Server

- DNS Server Logs: DNS queries, zone transfers
- RDP Sessions: Remote desktop connections
- Scheduled Tasks: Task creation, execution
- All Windows Host logs

1.3 Dedicated Mail Server VM Setup

Role: Dedicated Mail Transfer Agent (Postfix)

System Requirements: Ubuntu 20.04/22.04 LTS (Dedicated VM, distinct from General Ubuntu Server)

Postfix Setup

```
sudo apt install -y postfix
# Edit /etc/postfix/main.cf:
home_mailbox = Maildir/
mailbox_command =
# Restart
sudo systemctl restart postfix
```

Filebeat Configuration (Strict Separation)

Note: The configuration below separates the Mail Application data from the underlying Ubuntu System logs.

```
# /etc/filebeat/filebeat.yml

filebeat.inputs:

# ==========================================
# INPUT 1: Mail Application Logs (The Data)
# ==========================================
- type: log
  enabled: true
  paths:
    - /var/log/mail.log
    - /home/*/Maildir/new/*
    - /home/*/Maildir/cur/*
  fields:
    log_type: email_raw
    service: postfix_mail_server

# ==========================================
# INPUT 2: Ubuntu System Logs (The OS)
# ==========================================
- type: log
  enabled: true
  paths:
    - /var/log/syslog
    - /var/log/auth.log
  fields:
    log_type: os_system_logs
    host_role: mail_server_os

output.logstash:
  hosts: [":5044"]
```

Auditbeat Installation (File Integrity)

```
sudo apt install -y auditbeat
```

Configure `/etc/auditbeat/auditbeat.yml` to watch critical mail config files:

```
auditbeat.modules:
- module: file_integrity
```

```
  paths:
  - /etc/postfix
  - /bin
  - /usr/bin
  - /home/*/Maildir

output.logstash:
  hosts: [":5044"]
```

1.4 AWS CloudTrail Log Collection

Prerequisites

- AWS Account with CloudTrail enabled
- S3 bucket for CloudTrail logs
- IAM user with S3 read permissions

AWS Configuration

Enable CloudTrail (AWS Console):

1. Navigate to CloudTrail service
2. Create trail → Configure S3 bucket
3. Enable management events
4. Save trail configuration

Create IAM User:

```
{
  "Version": "2012-10-17",
  "Statement": [{
    "Effect": "Allow",
    "Action": ["s3:GetObject", "s3:ListBucket"],
    "Resource": [
      "arn:aws:s3:::your-cloudtrail-bucket/*",
      "arn:aws:s3:::your-cloudtrail-bucket"
    ]
  }]
}
```

Filebeat AWS Module

```
sudo filebeat modules enable aws
```

Edit /etc/filebeat/modules.d/aws.yml:

```
- module: aws
  cloudtrail:
    enabled: true
    var.bucket_arn: "arn:aws:s3:::your-cloudtrail-bucket"
    var.access_key_id: "YOUR_ACCESS_KEY"
    var.secret_access_key: "YOUR_SECRET_KEY"
```

```
sudo systemctl restart filebeat
```

Logs Collected

- IAM activity, EC2 events, S3 access
- Lambda executions, VPC changes
- All AWS API calls with metadata

1.5 pfSense Firewall & Gateway Setup

System Requirements

pfSense: 2.7.0 or later (free)
RAM: 2GB minimum
Network: 2 interfaces (WAN, LAN)

Configuration

Access pfSense: https://<PFSENSE_IP>

Enable Logging:

- Status > System Logs > Settings
- Enable packet logging
- Enable remote logging to <LOGSTASH_IP>:5140

Logs Collected

- Firewall allow/block events
- Gateway status changes
- DHCP assignments
- System events

1.6 Ubuntu General Server Log Collection

Role: Central Utility Server (ML Model Hosting & Caldera)

System: Ubuntu 22.04 LTS (4 Cores / 4 GiB)

Filebeat Configuration (General Server)

This configuration captures OS-level events and potential Python/ML script logs from the general utility server.

```
# /etc/filebeat/filebeat.yml on General Ubuntu VM

filebeat.inputs:
- type: log
  enabled: true
  paths:
    - /var/log/syslog
    - /var/log/auth.log
    - /var/log/kern.log
    - /var/log/dpkg.log
  fields:
    log_type: os_general_server
    host_role: ubuntu_ml_caldera

# Optional: Capture Python/ML Script Logs
- type: log
  enabled: true
  paths:
    - /var/log/phishing_detector/*.log
  fields:
    log_type: ml_application_logs

output.logstash:
  hosts: [":5044"]
```

Auditbeat Configuration

Monitors execution of suspicious commands (e.g., if Caldera agents go rogue).

```
auditbeat.modules:
- module: system
  datasets:
    - process  # Tracks every process started
    - socket   # Tracks network connections
    - user     # Tracks user logins

  # Detect when ML scripts are modified
- module: file_integrity
  paths:
    - /home/*/phishing-detector/
    - /usr/bin/python3

output.logstash:
  hosts: [":5044"]
```

1.7 Network Architecture

Data Sources → Beats → pfSense Gateway → Logstash → Elasticsearch → Kibana

Verification:

```
# Linux
sudo systemctl status filebeat auditbeat
sudo filebeat test output

# Windows
Get-Service winlogbeat
Test-NetConnection -ComputerName  -Port 5044
```

Part 2: APT Simulation with Caldera

Overview

Caldera simulates APT campaigns on separate dedicated VMs to test detection capabilities across the environment.

2.1 Caldera Server VM Setup

System Requirements

OS: Ubuntu 20.04/22.04 (dedicated VM)
RAM: 4GB
Disk: 20GB
Python: 3.8+
Network: Accessible on port 8888

Installation

```
# Clone repository
git clone https://github.com/mitre/caldera.git --recursive
cd caldera

# Install dependencies
pip3 install -r requirements.txt

# Configure
cp conf/default.yml conf/local.yml
nano conf/local.yml
# Edit:

host: 0.0.0.0
port: 8888
api_key: CHANGE_THIS_KEY
# Start Caldera
python3 server.py --insecure
```

Access: http://<CALDERA_IP>:8888 (admin / admin)

2.2 Caldera Agent VMs

Deploy agents on separate VMs (not on production Windows/Ubuntu servers).

Agent VM Requirements

- Windows Agent VM: Windows 10, 4GB RAM
- Linux Agent VM: Ubuntu 20.04, 2GB RAM
- Both VMs should have network access to Caldera server

Windows Agent Deployment

```
$server = "http://:8888"
$url = "$server/file/download"
Invoke-WebRequest -Uri $url -OutFile "sandcat.exe"
.\sandcat.exe -server $server -group red_team
```

Linux Agent Deployment

```
server="http://:8888"
curl -s -X POST -H "file:sandcat.go" -H "platform:linux" $server/file/download > sanc
chmod +x sandcat.sh
./sandcat.sh -server $server -group red_team
```

2.3 APT Simulation Operations

2.3.1 APT28 (Fancy Bear)

TTPs: Credential dumping, lateral movement, persistence

Execute:

- Caldera UI → Operations → Create
- Adversary: Hunter
- Target group: red_team
- Mode: Autonomous
- Start operation

Key TTPs Simulated:

- T1003: LSASS credential dumping
- T1021: RDP lateral movement
- T1053: Scheduled task persistence
- T1087: Account discovery

Expected Detections:

- Sysmon Event 10: LSASS access

- Event 4624: RDP logon type 10
- Event 4698: Scheduled task creation

2.3.2 APT36 (Transparent Tribe)

TTPs: Phishing, PowerShell execution, keylogging

Simulation:

- Send test phishing email to mail server
- Execute PowerShell obfuscated command
- Deploy keylogger simulation
- Establish C2 connection

Key TTPs:

- T1566.001: Spearphishing
- T1059.001: PowerShell
- T1056: Keylogging
- T1071: Application Layer Protocol (C2)

Expected Detections:

- ML phishing detection alert
- Event 4104: PowerShell script block
- Sysmon Event 3: Network connection
- Registry modification (Sysmon Event 13)

2.3.3 Full MITRE ATT&CK Coverage

Execute Complete Kill Chain:

- Reconnaissance → Discovery commands
- Initial Access → Phishing
- Execution → PowerShell/WMI
- Persistence → Registry/Services

- Privilege Escalation → UAC bypass
- Defense Evasion → Log clearing
- Credential Access → Mimikatz
- Discovery → Network enumeration
- Lateral Movement → RDP/WMI
- Collection → Data staging
- C2 → Beacon traffic
- Exfiltration → Data transfer

Caldera Operations: Run multiple adversary profiles in sequence or create custom operation.

2.4 Detection Mapping

APT	TTP	Detection Rule	Log Source
APT28	T1003	LSASS Memory Access	Sysmon Event 10
APT28	T1021.001	RDP Lateral Movement	Event 4624
APT36	T1566.001	Phishing Email	ML Model Alert
APT36	T1059.001	PowerShell	Event 4104
Both	T1053	Scheduled Task	Event 4698
Both	T1070	Log Clearing	Event 1102

Part 3: ELK Stack Setup & Detection Rules

3.1 ELK Stack Installation (Latest Version 8.17.0)

System Requirements
Elasticsearch: 16GB RAM, 500GB SSD
Logstash: 8GB RAM, 100GB disk
Kibana: 4GB RAM, 50GB disk
OS: Ubuntu 22.04 LTS

Elasticsearch Installation

```
wget -qO - https://artifacts.elastic.co/GPG-KEY-elasticsearch | sudo apt-key add -
echo "deb https://artifacts.elastic.co/packages/8.x/apt stable main" | sudo tee /etc/
sudo apt update
sudo apt install -y elasticsearch

# Configure heap
sudo nano /etc/elasticsearch/jvm.options.d/heap.options
# Add:

-Xms8g
-Xmx8g
sudo systemctl enable elasticsearch
sudo systemctl start elasticsearch

# Generate passwords
sudo /usr/share/elasticsearch/bin/elasticsearch-setup-passwords auto
```

Logstash Installation

```
sudo apt install -y logstash
```

Kibana Installation

```
sudo apt install -y kibana

# Configure
sudo nano /etc/kibana/kibana.yml
# Edit:

server.port: 5601
server.host: "0.0.0.0"
elasticsearch.hosts: ["http://localhost:9200"]
elasticsearch.username: "kibana_system"
elasticsearch.password: "YOUR_PASSWORD"
sudo systemctl enable kibana
sudo systemctl start kibana
```

Access: http://<KIBANA_IP>:5601

3.2 Logstash Pipelines

Input Configuration

/etc/logstash/conf.d/01-input.conf:

```
input {
  beats {
    port => 5044
    type => "beats"
  }
  udp {
    port => 5140
    type => "pfsense"
  }
}
```

Filter Configuration

/etc/logstash/conf.d/02-filter.conf:

```
filter {
  # GeoIP Enrichment (Free MaxMind GeoLite2)
  if [source][ip] {
    geoip {
      source => "[source][ip]"
      target => "[source][geo]"
      database => "/usr/share/GeoIP/GeoLite2-City.mmdb"
    }
  }

  # Custom DNS DGA Entropy Detection
  if [dns][question][name] {
    ruby {
      code => '
        domain = event.get("[dns][question][name]")
        if domain
          entropy = domain.chars.group_by(&:itself).values.map { |v| v.length / domai
          event.set("[dns][entropy]", entropy)
          event.set("[dns][suspicious]", entropy > 3.5)
        end
      '
    }
  }
}
```

Output Configuration

/etc/logstash/conf.d/03-output.conf:

```
output {
  elasticsearch {
    hosts => ["localhost:9200"]
    index => "%{[@metadata][beat]}-%{+YYYY.MM.dd}"
    user => "elastic"
    password => "YOUR_PASSWORD"
  }
}
```

```
sudo systemctl restart logstash
```

3.3 Detection Rules (TTP-Based)

Creating Detection Rules
Navigate: Kibana → Security → Rules → Detection rules (SIEM)

Rule 1: LSASS Memory Dump (T1003.001)

Type: Custom Query (KQL)
Query:

```
event.code:10 and winlog.event_data.TargetImage:*lsass.exe
```

Settings:
Name: LSASS Memory Access Detected
Severity: Critical
Risk Score: 90
MITRE: T1003.001

Rule 2: Suspicious PowerShell (T1059.001)

Type: Threshold
Query:

```
event.code:4104 and (powershell.file.script_block_text:*DownloadString* or powershell
```

Threshold: 3 events in 5 minutes per host
Settings:
Name: Suspicious PowerShell Execution
Severity: High
MITRE: T1059.001

Rule 3: RDP Lateral Movement (T1021.001)

Query:

```
event.code:4624 and winlog.event_data.LogonType:10 and not user.name:(Administrator o
```

Settings:
Name: RDP Lateral Movement
Severity: Medium
MITRE: T1021.001

3.4 Sequence Rules for APT Detection

APT28 Kill Chain

Type: EQL Sequence
Query:

```
sequence by host.name with maxspan=1h
  [process where process.name : "net.exe"]
  [process where process.name : "procdump.exe"]
  [network where process.name : "powershell.exe" and destination.port : 443]
```

Explanation: Detects reconnaissance → credential dumping → C2 communication
Settings:
Name: APT28 Kill Chain Detected
Severity: Critical
MITRE: Multiple TTPs

APT36 Phishing Chain

Query:

```
sequence by host.name with maxspan=2h
  [file where file.extension : "docx" or file.extension : "pdf"]
  [process where process.name : "powershell.exe"]
  [registry where registry.path : "*\\Run\\*"]
```

Explanation: Phishing attachment → PowerShell execution → Persistence

3.5 Using Elastic Security APIs

List Detection Rules

```
curl -X GET "http://localhost:5601/api/detection_engine/rules/_find" \
  -u elastic:YOUR_PASSWORD \
  -H "kbn-xsrf: true" \
  -H "Content-Type: application/json"
```

Create Detection Rule via API

```
curl -X POST "http://localhost:5601/api/detection_engine/rules" \
  -u elastic:YOUR_PASSWORD \
  -H "kbn-xsrf: true" \
  -H "Content-Type: application/json" \
  -d '{
    "name": "Test Rule",
    "description": "API created rule",
    "risk_score": 75,
    "severity": "high",
    "type": "query",
    "query": "event.code:4624",
    "language": "kuery",
    "interval": "5m",
    "enabled": true
  }'
```

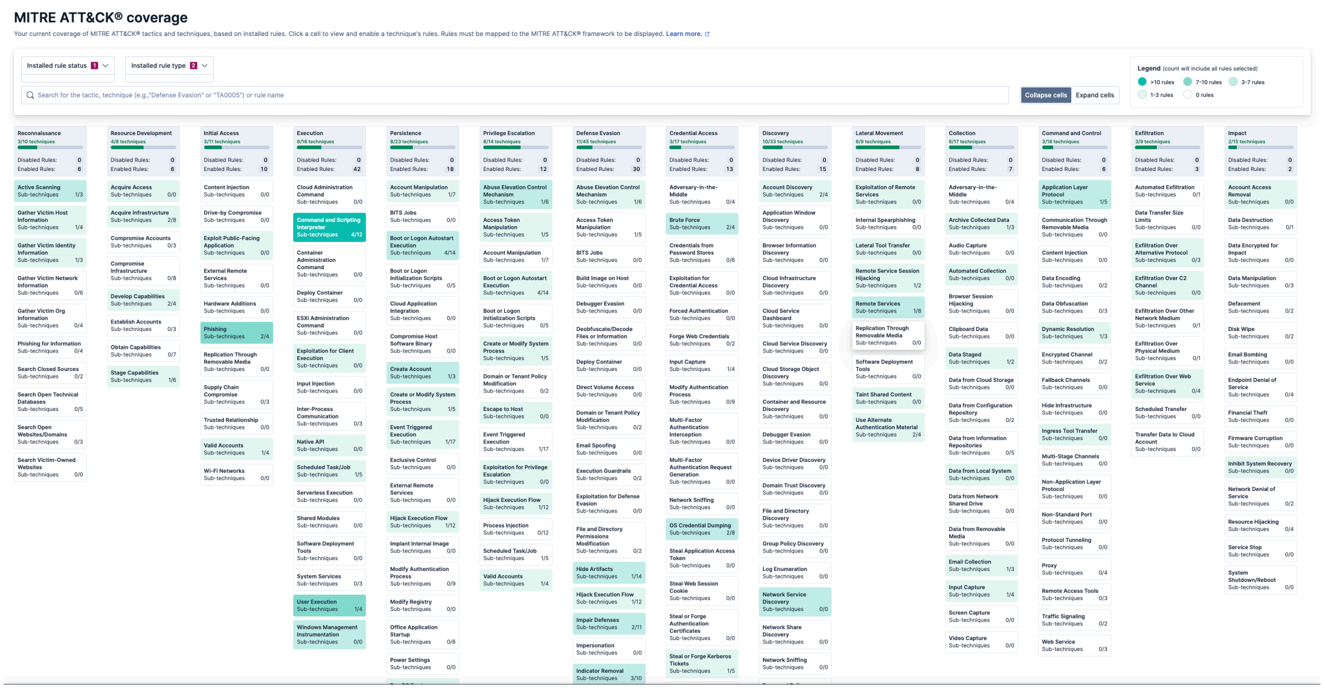
Get Alerts

```
curl -X GET "http://localhost:5601/api/detection_engine/signals" \
  -u elastic:YOUR_PASSWORD \
    -H "kbn-xsrf: true"
```

API Documentation: https://www.elastic.co/guide/en/security/current/rule-api-overview.html

3.6 MITRE ATT&CK Coverage

The following matrix displays the current coverage of MITRE ATT&CK tactics and techniques based on the installed detection rules within the Elastic Security platform.



MITRE ATT&CK® coverage

Your current coverage of MITRE ATT&CK® tactics and techniques, based on installed rules. Click a cell to view and enable a technique's rules. Rules must be mapped to the MITRE ATT&CK® framework to be displayed. Learn more.

Installed rule status 1 | Installed rule type 2

Search for the tactic, technique (e.g.,"Defense Evasion" or "TA0005") or rule name

Collapse cells | Expand cells

Legend (count will include all rules selected): ● >10 rules ● 7-10 rules ● 3-7 rules ○ 1-3 rules ○ 0 rules

Reconnaissance — 3/10 techniques — Disabled Rules: 0 — Enabled Rules: 6
- Active Scanning — Sub-techniques 1/3
- Gather Victim Host Information — Sub-techniques 1/4
- Gather Victim Identity Information — Sub-techniques 1/3
- Gather Victim Network Information — Sub-techniques 0/6
- Gather Victim Org Information — Sub-techniques 0/4
- Phishing for Information — Sub-techniques 0/4
- Search Closed Sources — Sub-techniques 0/2
- Search Open Technical Databases — Sub-techniques 0/5
- Search Open Websites/Domains — Sub-techniques 0/3
- Search Victim-Owned Websites — Sub-techniques 0/0

Resource Development — 4/8 techniques — Disabled Rules: 0 — Enabled Rules: 6
- Acquire Access — Sub-techniques 0/0
- Acquire Infrastructure — Sub-techniques 2/8
- Compromise Accounts — Sub-techniques 0/3
- Compromise Infrastructure — Sub-techniques 0/8
- Develop Capabilities — Sub-techniques 2/4
- Establish Accounts — Sub-techniques 0/3
- Obtain Capabilities — Sub-techniques 0/7
- Stage Capabilities — Sub-techniques 1/6

Initial Access — 3/11 techniques — Disabled Rules: 0 — Enabled Rules: 10
- Content Injection — Sub-techniques 0/0
- Drive-by Compromise — Sub-techniques 0/0
- Exploit Public-Facing Application — Sub-techniques 0/0
- External Remote Services — Sub-techniques 0/0
- Hardware Additions — Sub-techniques 0/0
- Phishing — Sub-techniques 2/4
- Replication Through Removable Media — Sub-techniques 0/0
- Supply Chain Compromise — Sub-techniques 0/3
- Trusted Relationship — Sub-techniques 0/0
- Valid Accounts — Sub-techniques 1/4
- Wi-Fi Networks — Sub-techniques 0/0

Execution — 6/16 techniques — Disabled Rules: 0 — Enabled Rules: 42
- Cloud Administration Command — Sub-techniques 0/0
- Command and Scripting Interpreter — Sub-techniques 4/12
- Container Administration Command — Sub-techniques 0/0
- Deploy Container — Sub-techniques 0/0
- ESXi Administration Command — Sub-techniques 0/0
- Exploitation for Client Execution — Sub-techniques 0/0
- Input Injection — Sub-techniques 0/0
- Inter-Process Communication — Sub-techniques 0/3
- Native API — Sub-techniques 0/0
- Scheduled Task/Job — Sub-techniques 1/5
- Serverless Execution — Sub-techniques 0/0
- Shared Modules — Sub-techniques 0/0
- Software Deployment Tools — Sub-techniques 0/0
- System Services — Sub-techniques 0/3
- User Execution — Sub-techniques 1/4
- Windows Management Instrumentation — Sub-techniques 0/0

Persistence — 8/23 techniques — Disabled Rules: 0 — Enabled Rules: 18
- Account Manipulation — Sub-techniques 1/7
- BITS Jobs — Sub-techniques 0/0
- Boot or Logon Autostart Execution — Sub-techniques 4/14
- Boot or Logon Initialization Scripts — Sub-techniques 0/5
- Cloud Application Integration — Sub-techniques 0/0
- Compromise Host Software Binary — Sub-techniques 0/0
- Create Account — Sub-techniques 1/3
- Create or Modify System Process — Sub-techniques 1/5
- Event Triggered Execution — Sub-techniques 1/17
- Exclusive Control — Sub-techniques 0/0
- External Remote Services — Sub-techniques 0/0
- Hijack Execution Flow — Sub-techniques 1/12
- Implant Internal Image — Sub-techniques 0/0
- Modify Authentication Process — Sub-techniques 0/9
- Modify Registry — Sub-techniques 0/0
- Office Application Startup — Sub-techniques 0/6
- Power Settings — Sub-techniques 0/0

Privilege Escalation — 8/14 techniques — Disabled Rules: 0 — Enabled Rules: 12
- Abuse Elevation Control Mechanism — Sub-techniques 1/6
- Access Token Manipulation — Sub-techniques 1/5
- Account Manipulation — Sub-techniques 1/7
- Boot or Logon Autostart Execution — Sub-techniques 4/14
- Boot or Logon Initialization Scripts — Sub-techniques 0/5
- Create or Modify System Process — Sub-techniques 1/5
- Domain or Tenant Policy Modification — Sub-techniques 0/2
- Escape to Host — Sub-techniques 0/0
- Event Triggered Execution — Sub-techniques 1/17
- Exploitation for Privilege Escalation — Sub-techniques 0/0
- Hijack Execution Flow — Sub-techniques 1/12
- Process Injection — Sub-techniques 0/12
- Scheduled Task/Job — Sub-techniques 1/5
- Valid Accounts — Sub-techniques 1/4

Defense Evasion — 11/45 techniques — Disabled Rules: 0 — Enabled Rules: 30
- Abuse Elevation Control Mechanism — Sub-techniques 1/6
- Access Token Manipulation — Sub-techniques 1/5
- BITS Jobs — Sub-techniques 0/0
- Build Image on Host — Sub-techniques 0/0
- Debugger Evasion — Sub-techniques 0/0
- Deobfuscate/Decode Files or Information — Sub-techniques 0/0
- Deploy Container — Sub-techniques 0/0
- Direct Volume Access — Sub-techniques 0/0
- Domain or Tenant Policy Modification — Sub-techniques 0/2
- Email Spoofing — Sub-techniques 0/0
- Execution Guardrails — Sub-techniques 0/2
- Exploitation for Defense Evasion — Sub-techniques 0/0
- File and Directory Permissions Modification — Sub-techniques 0/2
- Hide Artifacts — Sub-techniques 1/14
- Hijack Execution Flow — Sub-techniques 1/12
- Impair Defenses — Sub-techniques 2/11
- Impersonation — Sub-techniques 0/0
- Indicator Removal — Sub-techniques 3/10

Credential Access — 3/17 techniques — Disabled Rules: 0 — Enabled Rules: 13
- Adversary-in-the-Middle — Sub-techniques 0/4
- Brute Force — Sub-techniques 2/4
- Credentials from Password Stores — Sub-techniques 0/6
- Exploitation for Credential Access — Sub-techniques 0/0
- Forced Authentication — Sub-techniques 0/0
- Forge Web Credentials — Sub-techniques 0/2
- Input Capture — Sub-techniques 1/4
- Modify Authentication Process — Sub-techniques 0/9
- Multi-Factor Authentication Interception — Sub-techniques 0/0
- Multi-Factor Authentication Request Generation — Sub-techniques 0/0
- Network Sniffing — Sub-techniques 0/0
- OS Credential Dumping — Sub-techniques 2/8
- Steal Application Access Token — Sub-techniques 0/0
- Steal Web Session Cookie — Sub-techniques 0/0
- Steal or Forge Authentication Certificates — Sub-techniques 0/0
- Steal or Forge Kerberos Tickets — Sub-techniques 1/5

Discovery — 10/33 techniques — Disabled Rules: 0 — Enabled Rules: 15
- Account Discovery — Sub-techniques 2/4
- Application Window Discovery — Sub-techniques 0/0
- Browser Information Discovery — Sub-techniques 0/0
- Cloud Infrastructure Discovery — Sub-techniques 0/0
- Cloud Service Dashboard — Sub-techniques 0/0
- Cloud Service Discovery — Sub-techniques 0/0
- Cloud Storage Object Discovery — Sub-techniques 0/0
- Container and Resource Discovery — Sub-techniques 0/0
- Debugger Evasion — Sub-techniques 0/0
- Device Driver Discovery — Sub-techniques 0/0
- Domain Trust Discovery — Sub-techniques 0/0
- File and Directory Discovery — Sub-techniques 0/0
- Group Policy Discovery — Sub-techniques 0/0
- Log Enumeration — Sub-techniques 0/0
- Network Service Discovery — Sub-techniques 0/0
- Network Share Discovery — Sub-techniques 0/0
- Network Sniffing — Sub-techniques 0/0

Lateral Movement — 6/9 techniques — Disabled Rules: 0 — Enabled Rules: 8
- Exploitation of Remote Services — Sub-techniques 0/0
- Internal Spearphishing — Sub-techniques 0/0
- Lateral Tool Transfer — Sub-techniques 0/0
- Remote Service Session Hijacking — Sub-techniques 1/2
- Remote Services — Sub-techniques 1/8
- Replication Through Removable Media — Sub-techniques 0/0
- Software Deployment Tools — Sub-techniques 0/0
- Taint Shared Content — Sub-techniques 0/0
- Use Alternate Authentication Material — Sub-techniques 2/4

Collection — 6/17 techniques — Disabled Rules: 0 — Enabled Rules: 7
- Adversary-in-the-Middle — Sub-techniques 0/4
- Archive Collected Data — Sub-techniques 1/3
- Audio Capture — Sub-techniques 0/0
- Automated Collection — Sub-techniques 0/0
- Browser Session Hijacking — Sub-techniques 0/0
- Clipboard Data — Sub-techniques 0/0
- Data Staged — Sub-techniques 1/2
- Data from Cloud Storage — Sub-techniques 0/0
- Data from Configuration Repository — Sub-techniques 0/2
- Data from Information Repositories — Sub-techniques 0/5
- Data from Local System — Sub-techniques 0/0
- Data from Network Shared Drive — Sub-techniques 0/0
- Data from Removable Media — Sub-techniques 0/0
- Email Collection — Sub-techniques 1/3
- Input Capture — Sub-techniques 1/4
- Screen Capture — Sub-techniques 0/0
- Video Capture — Sub-techniques 0/0

Command and Control — 3/18 techniques — Disabled Rules: 0 — Enabled Rules: 6
- Application Layer Protocol — Sub-techniques 1/5
- Communication Through Removable Media — Sub-techniques 0/0
- Content Injection — Sub-techniques 0/0
- Data Encoding — Sub-techniques 0/2
- Data Obfuscation — Sub-techniques 0/3
- Dynamic Resolution — Sub-techniques 1/3
- Encrypted Channel — Sub-techniques 0/2
- Fallback Channels — Sub-techniques 0/0
- Hide Infrastructure — Sub-techniques 0/0
- Ingress Tool Transfer — Sub-techniques 0/0
- Multi-Stage Channels — Sub-techniques 0/0
- Non-Application Layer Protocol — Sub-techniques 0/0
- Non-Standard Port — Sub-techniques 0/0
- Protocol Tunneling — Sub-techniques 0/0
- Proxy — Sub-techniques 0/4
- Remote Access Tools — Sub-techniques 0/3
- Traffic Signaling — Sub-techniques 0/2
- Web Service — Sub-techniques 0/3

Exfiltration — 3/9 techniques — Disabled Rules: 0 — Enabled Rules: 3
- Automated Exfiltration — Sub-techniques 0/1
- Data Transfer Size Limits — Sub-techniques 0/0
- Exfiltration Over Alternative Protocol — Sub-techniques 0/3
- Exfiltration Over C2 Channel — Sub-techniques 0/0
- Exfiltration Over Other Network Medium — Sub-techniques 0/1
- Exfiltration Over Physical Medium — Sub-techniques 0/1
- Exfiltration Over Web Service — Sub-techniques 0/4
- Scheduled Transfer — Sub-techniques 0/0
- Transfer Data to Cloud Account — Sub-techniques 0/0

Impact — 2/15 techniques — Disabled Rules: 0 — Enabled Rules: 2
- Account Access Removal — Sub-techniques 0/0
- Data Destruction — Sub-techniques 0/1
- Data Encrypted for Impact — Sub-techniques 0/0
- Data Manipulation — Sub-techniques 0/3
- Defacement — Sub-techniques 0/2
- Disk Wipe — Sub-techniques 0/2
- Email Bombing — Sub-techniques 0/0
- Endpoint Denial of Service — Sub-techniques 0/4
- Financial Theft — Sub-techniques 0/0
- Firmware Corruption — Sub-techniques 0/0
- Inhibit System Recovery — Sub-techniques 0/0
- Network Denial of Service — Sub-techniques 0/2
- Resource Hijacking — Sub-techniques 0/4
- Service Stop — Sub-techniques 0/0
- System Shutdown/Reboot — Sub-techniques 0/0

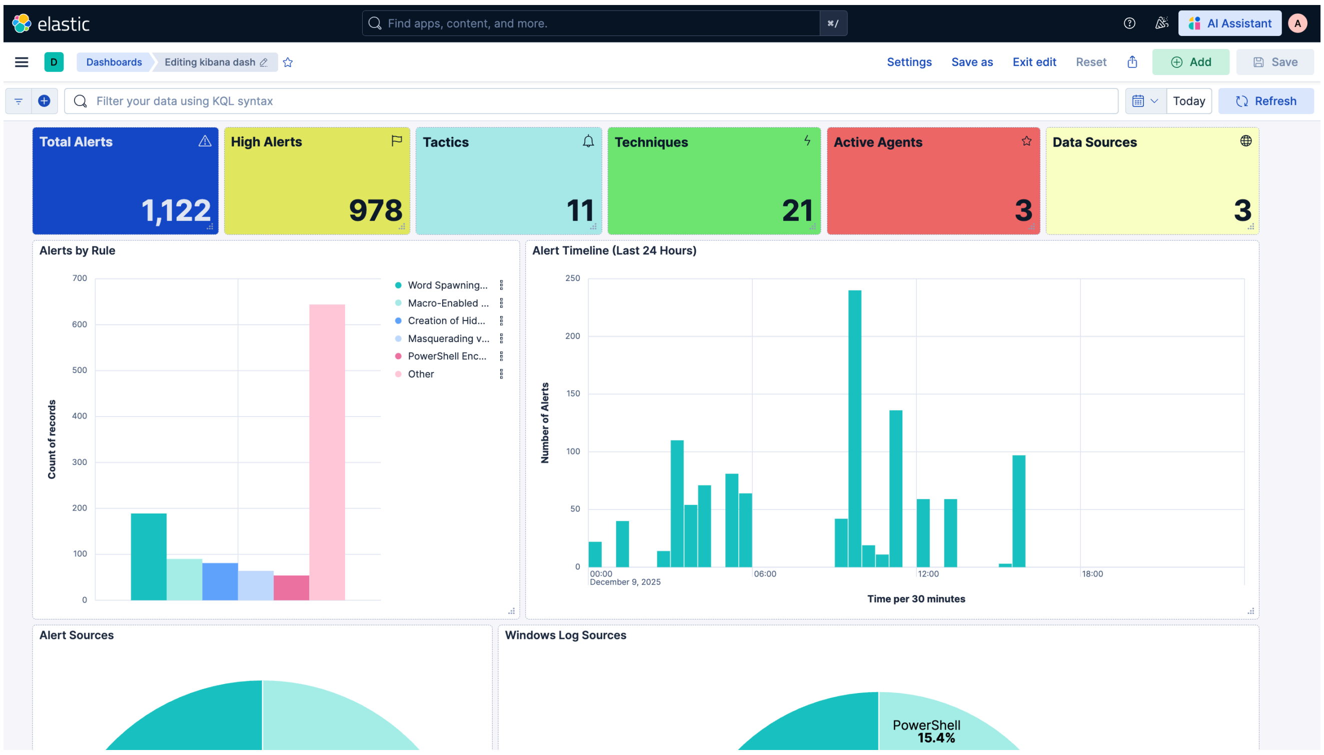
The heatmap visualizes the number of enabled rules for each technique, providing an overview of the platform's defensive capabilities against known adversarial behaviors.

3.7 Kibana Dashboards

The following dashboards provide real-time visibility into the security posture, alert trends, and specific APT activity sequences.

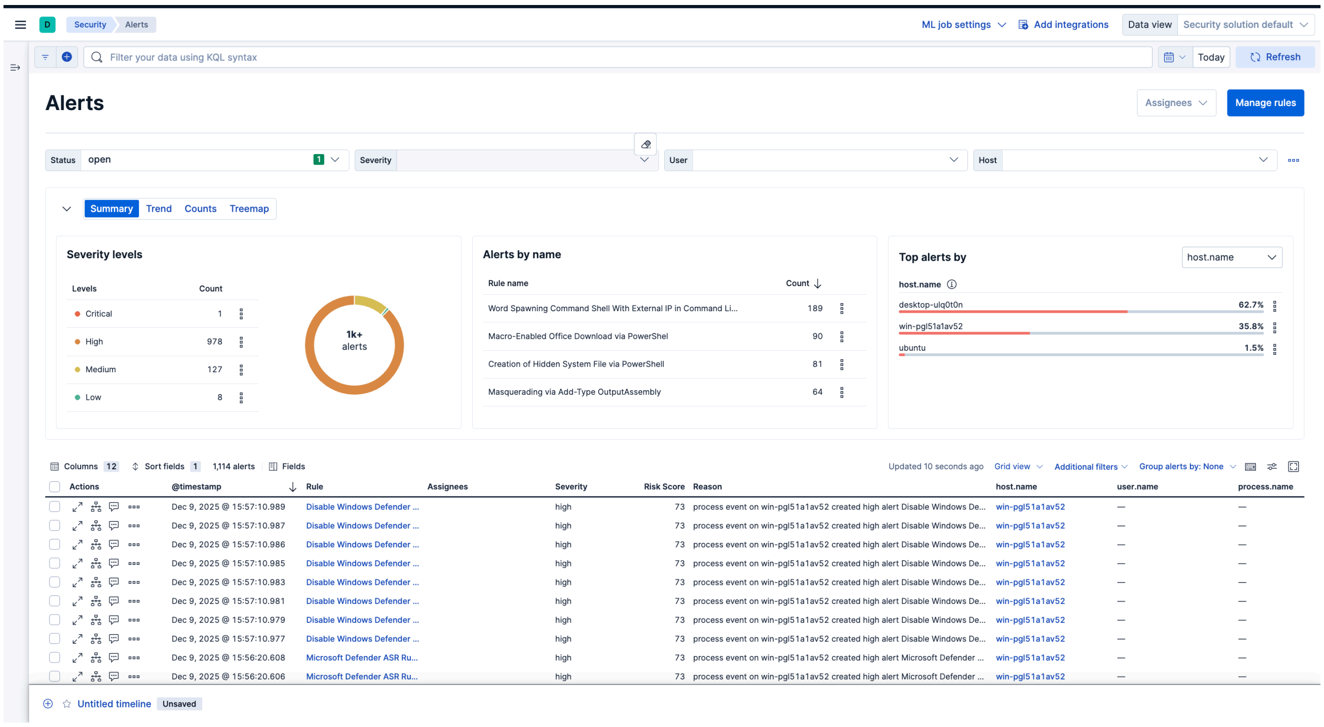
Overview Dashboard

High-level metrics showing total alerts (1,122), high-severity incidents (978), and active agents.



Alerts & Host Activity

Detailed breakdown of alert sources, showing top alerts by rule name and host distribution (e.g., desktop-ulq0t0n).


Security / Alerts

ML job settings | Add integrations | Data view: Security solution default

Filter your data using KQL syntax | Today | Refresh

Alerts

Assignees | Manage rules

Status: open (1) | Severity | User | Host

Summary | Trend | Counts | Treemap

Severity levels

Levels	Count
● Critical	1
● High	978
● Medium	127
● Low	8

1k+ alerts

Alerts by name

Rule name	Count
Word Spawning Command Shell With External IP in Command Li...	189
Macro-Enabled Office Download via PowerShel	90
Creation of Hidden System File via PowerShell	81
Masquerading via Add-Type OutputAssembly	64

Top alerts by host.name

host.name	
desktop-ulq0t0n	62.7%
win-pgl51a1av52	35.8%
ubuntu	1.5%

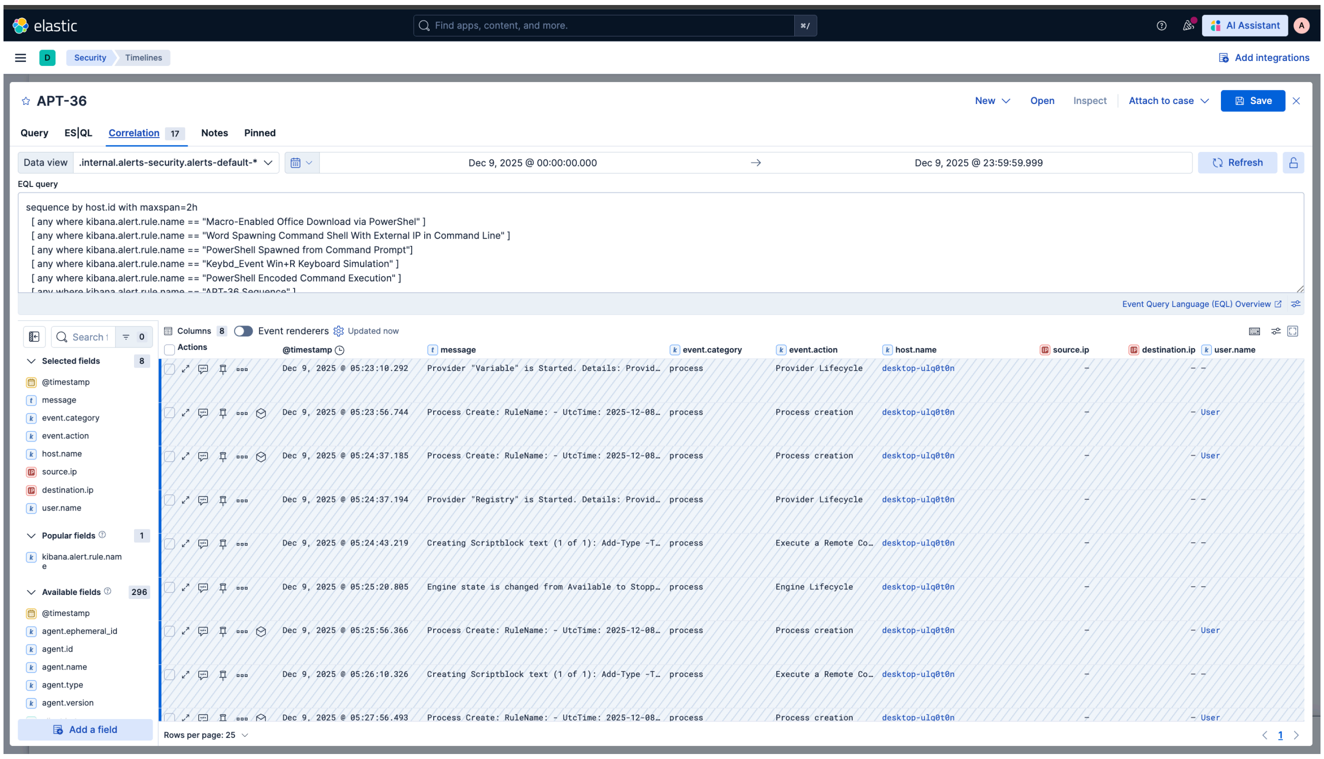
Columns 12 | Sort fields 1 | 1,114 alerts | Fields | Updated 10 seconds ago | Grid view | Additional filters | Group alerts by: None

@timestamp	Rule	Assignees	Severity	Risk Score	Reason	host.name	user.name	process.name
Dec 9, 2025 @ 15:57:10.989	Disable Windows Defender ...		high	73	process event on win-pgl51a1av52 created high alert Disable Windows De...	win-pgl51a1av52	—	—
Dec 9, 2025 @ 15:57:10.987	Disable Windows Defender ...		high	73	process event on win-pgl51a1av52 created high alert Disable Windows De...	win-pgl51a1av52	—	—
Dec 9, 2025 @ 15:57:10.986	Disable Windows Defender ...		high	73	process event on win-pgl51a1av52 created high alert Disable Windows De...	win-pgl51a1av52	—	—
Dec 9, 2025 @ 15:57:10.985	Disable Windows Defender ...		high	73	process event on win-pgl51a1av52 created high alert Disable Windows De...	win-pgl51a1av52	—	—
Dec 9, 2025 @ 15:57:10.983	Disable Windows Defender ...		high	73	process event on win-pgl51a1av52 created high alert Disable Windows De...	win-pgl51a1av52	—	—
Dec 9, 2025 @ 15:57:10.981	Disable Windows Defender ...		high	73	process event on win-pgl51a1av52 created high alert Disable Windows De...	win-pgl51a1av52	—	—
Dec 9, 2025 @ 15:57:10.979	Disable Windows Defender ...		high	73	process event on win-pgl51a1av52 created high alert Disable Windows De...	win-pgl51a1av52	—	—
Dec 9, 2025 @ 15:57:10.977	Disable Windows Defender ...		high	73	process event on win-pgl51a1av52 created high alert Disable Windows De...	win-pgl51a1av52	—	—
Dec 9, 2025 @ 15:56:20.608	Microsoft Defender ASR Ru...		high	73	process event on win-pgl51a1av52 created high alert Microsoft Defender ...	win-pgl51a1av52	—	—
Dec 9, 2025 @ 15:56:20.606	Microsoft Defender ASR Ru...		high	73	process event on win-pgl51a1av52 created high alert Microsoft Defender ...	win-pgl51a1av52	—	—

Untitled timeline | Unsaved

APT-36 Correlation Timeline

Sequence correlation rule "APT-36 Sequence" triggering on specific TTPs like Macro-Enabled Downloads and Keybd_Event simulation.



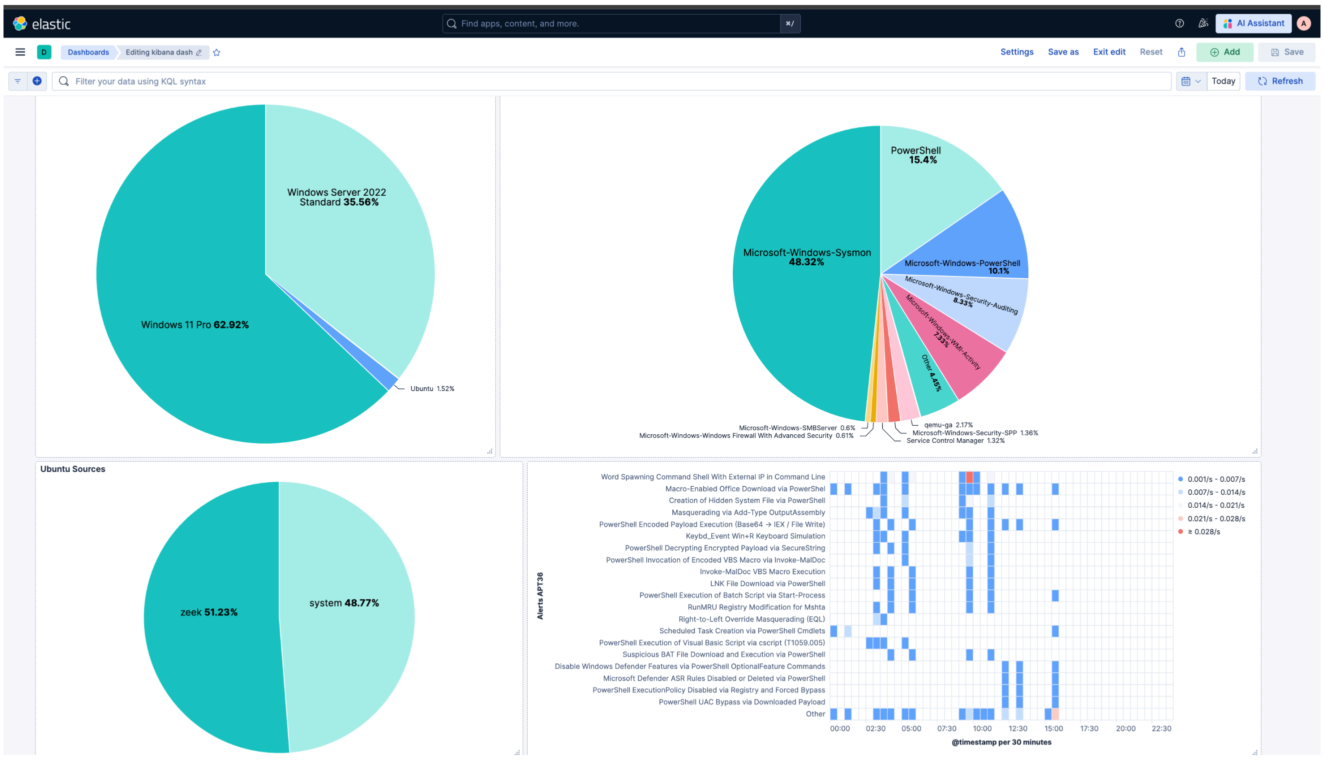
EQL query

```
sequence by host.id with maxspan=2h
  [ any where kibana.alert.rule.name == "Macro-Enabled Office Download via PowerShel" ]
  [ any where kibana.alert.rule.name == "Word Spawning Command Shell With External IP in Command Line" ]
  [ any where kibana.alert.rule.name == "PowerShell Spawned from Command Prompt"]
  [ any where kibana.alert.rule.name == "Keybd_Event Win+R Keyboard Simulation" ]
  [ any where kibana.alert.rule.name == "PowerShell Encoded Command Execution" ]
  [ any where kibana.alert.rule.name == "APT-36 Sequence" ]
```

@timestamp	message	event.category	event.action	host.name	source.ip	destination.ip	user.name
Dec 9, 2025 @ 05:23:10.292	Provider "Variable" is Started. Details: Provid…	process	Provider Lifecycle	desktop-ulq0t0n	–	–	–
Dec 9, 2025 @ 05:23:56.744	Process Create: RuleName: - UtcTime: 2025-12-08…	process	Process creation	desktop-ulq0t0n	–	–	User
Dec 9, 2025 @ 05:24:37.185	Process Create: RuleName: - UtcTime: 2025-12-08…	process	Process creation	desktop-ulq0t0n	–	–	User
Dec 9, 2025 @ 05:24:37.194	Provider "Registry" is Started. Details: Provid…	process	Provider Lifecycle	desktop-ulq0t0n	–	–	–
Dec 9, 2025 @ 05:24:43.219	Creating Scriptblock text (1 of 1): Add-Type -T…	process	Execute a Remote Co…	desktop-ulq0t0n	–	–	–
Dec 9, 2025 @ 05:25:20.805	Engine state is changed from Available to Stopp…	process	Engine Lifecycle	desktop-ulq0t0n	–	–	–
Dec 9, 2025 @ 05:25:56.366	Process Create: RuleName: - UtcTime: 2025-12-08…	process	Process creation	desktop-ulq0t0n	–	–	User
Dec 9, 2025 @ 05:26:10.326	Creating Scriptblock text (1 of 1): Add-Type -T…	process	Execute a Remote Co…	desktop-ulq0t0n	–	–	–
Dec 9, 2025 @ 05:27:56.493	Process Create: RuleName: - UtcTime: 2025-12-08…	process	Process creation	desktop-ulq0t0n	–	–	User

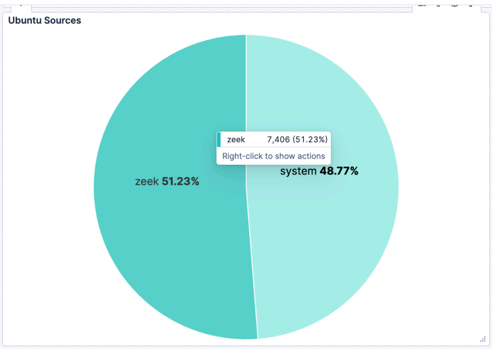
Log Source Distribution

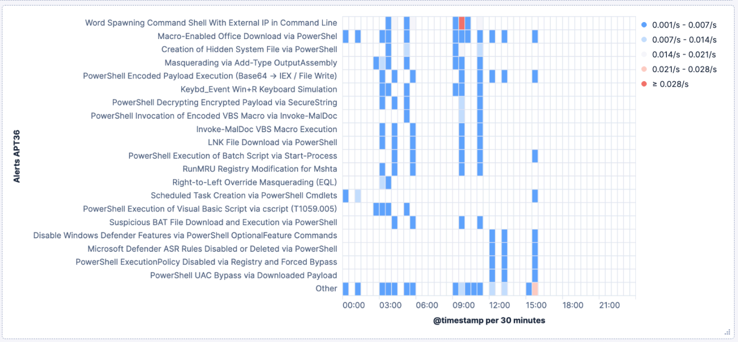
Visual breakdown of incoming log volumes by operating system (Windows 11, Server 2022) and specific log channels (Sysmon, PowerShell).

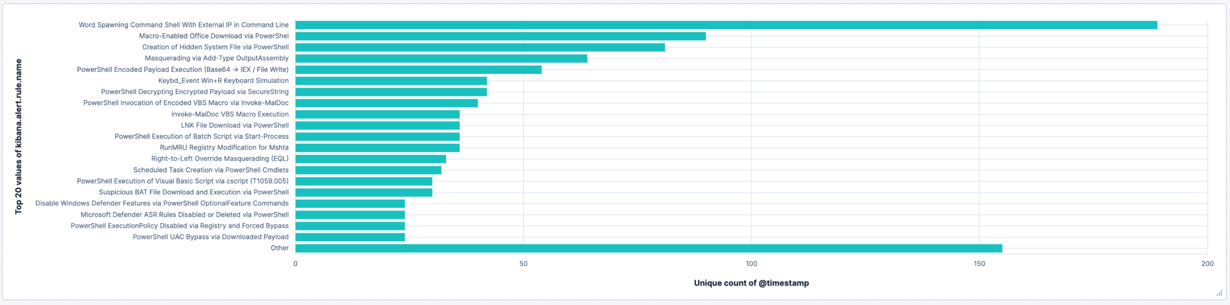


Ubuntu & Zeek Network Logs

Visualization of network traffic logs from Zeek (51.23%) and system logs (48.77%) on the Ubuntu sensor.

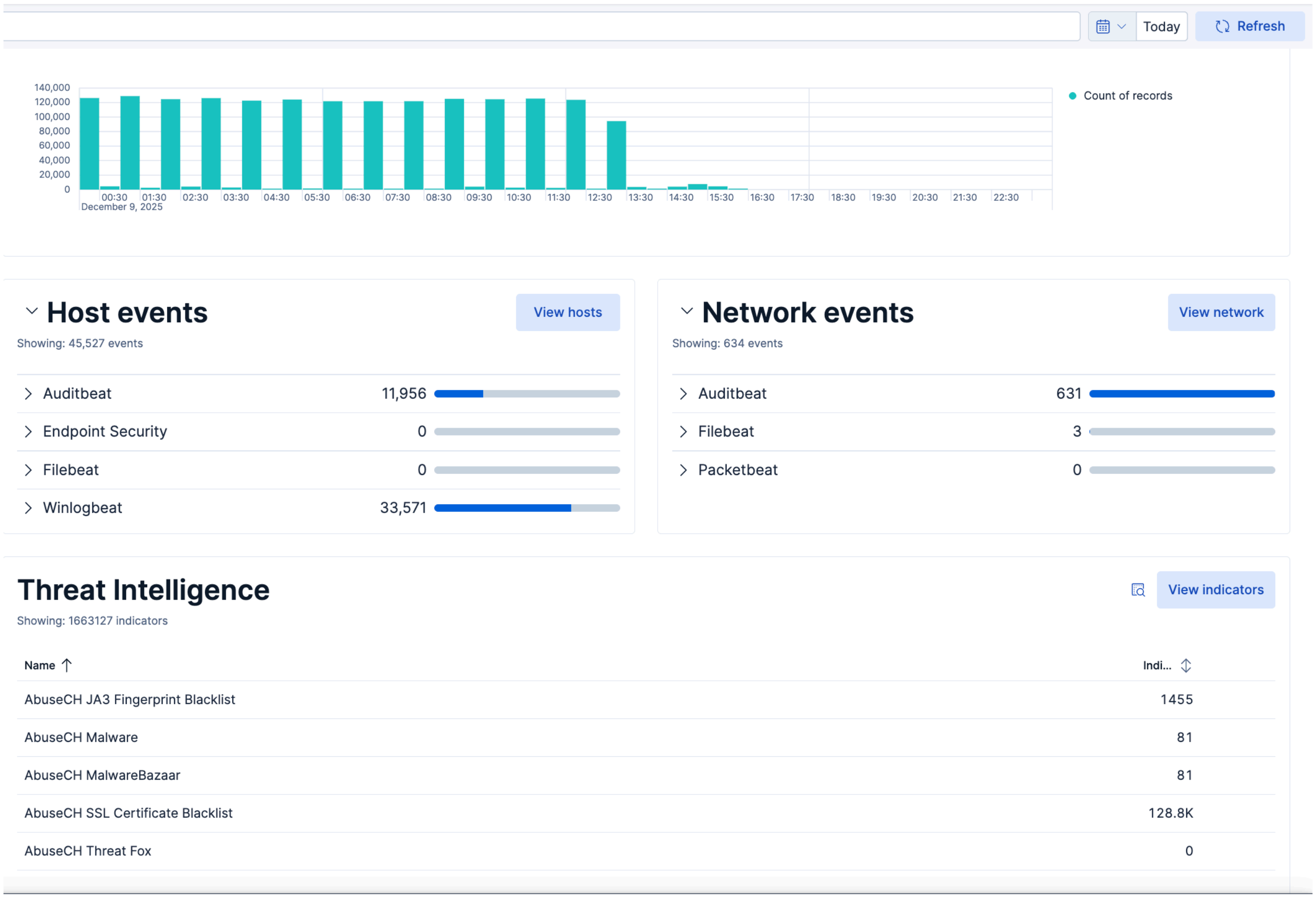






Data Ingestion Metrics

Ingestion rates showing record counts for Winlogbeat (33k+ events) and Auditbeat, confirming active data flow from all endpoints.



Host events View hosts

Showing: 45,527 events

Auditbeat	11,956
Endpoint Security	0
Filebeat	0
Winlogbeat	33,571

Network events View network

Showing: 634 events

Auditbeat	631
Filebeat	3
Packetbeat	0

Threat Intelligence View indicators

Showing: 1663127 indicators

Name ↑	Indi...
AbuseCH JA3 Fingerprint Blacklist	1455
AbuseCH Malware	81
AbuseCH MalwareBazaar	81
AbuseCH SSL Certificate Blacklist	128.8K
AbuseCH Threat Fox	0

Part 4: ML-Based Phishing Detection

Overview

Dedicated ML server running DistilBERT-based phishing detection model, monitoring mail server for incoming emails.

4.1 ML Server VM Setup

System Requirements
OS: Ubuntu 22.04 LTS (dedicated VM)
RAM: 8GB minimum, 16GB recommended
GPU: Optional (NVIDIA GPU with CUDA for faster inference)
Disk: 50GB
Python: 3.9+
Network: Access to mail server Maildir

Base Installation

```
sudo apt update && sudo apt upgrade -y
sudo apt install -y python3 python3-pip python3-venv git
```

Python Environment Setup

```
# Create virtual environment
python3 -m venv ~/phishing-detector
source ~/phishing-detector/bin/activate

# Install PyTorch (CPU version - free)
pip install torch torchvision torchaudio --index-url https://download.pytorch.org/whl

# Install Transformers and dependencies
pip install transformers==4.36.0
pip install scikit-learn pandas numpy
pip install email-parser mailparser
```

4.2 Email Reception & Processing

Postfix Configuration (on Mail Server)

Mail server should deliver emails to Maildir format (configured in Part 1.3).

Maildog Installation (on ML Server)

```
pip install watchdog
```

Create monitoring script email_monitor.py:

```
from watchdog.observers import Observer
from watchdog.events import FileSystemEventHandler
import os
import time

class EmailHandler(FileSystemEventHandler):
    def on_created(self, event):
        if not event.is_directory and '/new/' in event.src_path:
            print(f"New email detected: {event.src_path}")
            # Call ML processing
            process_email(event.src_path)

def process_email(filepath):
    # Extract email content
    with open(filepath, 'r', errors='ignore') as f:
        email_content = f.read()

    # Send to ML model
    result = classify_email(email_content)

    # Log result to file (will be picked up by Filebeat)
    log_result(filepath, result)

if __name__ == "__main__":
    observer = Observer()
    handler = EmailHandler()

    # Monitor mail server's Maildir (mount via NFS/SSHFS)
    path = "/mnt/mailserver/Maildir/new"
    observer.schedule(handler, path, recursive=False)
    observer.start()

    try:
        while True:
            time.sleep(1)
    except KeyboardInterrupt:
        observer.stop()
    observer.join()
```

4.3 DistilBERT Model Setup

Model Training (Free Approach)

Option 1: Use pre-trained phishing detection model from Hugging Face:

```
pip install huggingface_hub
from transformers import AutoTokenizer, AutoModelForSequenceClassification
import torch

# Load pre-trained model (example)
model_name = "ealvaradob/bert-finetuned-phishing"
tokenizer = AutoTokenizer.from_pretrained(model_name)
model = AutoModelForSequenceClassification.from_pretrained(model_name)
```

Option 2: Fine-tune DistilBERT on custom dataset (free public datasets available).

Model Inference Script

Create phishing_classifier.py:

```
from transformers import DistilBertTokenizer, DistilBertForSequenceClassification
import torch
import email
from email import policy

class PhishingDetector:
    def __init__(self):
        self.tokenizer = DistilBertTokenizer.from_pretrained('distilbert-base-uncasec
        self.model = DistilBertForSequenceClassification.from_pretrained('./model')
        self.model.eval()

    def extract_features(self, email_path):
        with open(email_path, 'rb') as f:
            msg = email.message_from_binary_file(f, policy=policy.default)

        subject = msg['subject'] or ""
        body = ""

        if msg.is_multipart():
            for part in msg.walk():
                if part.get_content_type() == "text/plain":
                    body = part.get_payload(decode=True).decode('utf-8', errors='ignc
                    break
        else:
            body = msg.get_payload(decode=True).decode('utf-8', errors='ignore')

        # Combine subject and body
        text = f"{subject} {body[:500]}"  # First 500 chars
        return text

    def classify(self, text):
        inputs = self.tokenizer(text, return_tensors="pt", truncation=True,
                                max_length=512, padding=True)

        with torch.no_grad():
            outputs = self.model(**inputs)
            logits = outputs.logits
            probs = torch.softmax(logits, dim=1)
            prediction = torch.argmax(probs, dim=1).item()
            confidence = probs[0][prediction].item()

        # 0: legitimate, 1: phishing
        label = "phishing" if prediction == 1 else "legitimate"
        return label, confidence

detector = PhishingDetector()

def classify_email(email_path):
    text = detector.extract_features(email_path)
    label, confidence = detector.classify(text)
    return {"label": label, "confidence": confidence, "path": email_path}
```

4.4 Classification & Logging

Output to Elasticsearch

Create log_results.py:

```
import json
import datetime

def log_result(email_path, result):
    log_entry = {
        "timestamp": datetime.datetime.utcnow().isoformat(),
        "email_path": email_path,
        "classification": result["label"],
        "confidence": result["confidence"],
        "alert": result["label"] == "phishing"
    }

    # Write to log file (Filebeat will pick this up)
    with open("/var/log/phishing_detection.log", "a") as f:
        f.write(json.dumps(log_entry) + "\n")
```

Filebeat Configuration (on ML Server)

```
filebeat.inputs:
- type: log
  paths:
    - /var/log/phishing_detection.log
  json.keys_under_root: true
  json.add_error_key: true
  fields:
    log_type: ml_phishing_detection

output.logstash:
  hosts: [":5044"]
```

4.5 System Architecture

Mail Server → Maildir → [Monitor] → ML Server (DistilBERT) → Classification
↓
Log Results → Filebeat → Logstash → Elasticsearch → Kibana

Starting the System

```
# On ML Server
source ~/phishing-detector/bin/activate
python3 email_monitor.py &
```

Monitoring

View logs:

```
tail -f /var/log/phishing_detection.log
```

Kibana Dashboard: Create visualization for phishing alerts:

- Filter: log_type: ml_phishing_detection AND alert: true
- Visualization: Time series of phishing detections

Part 5: Threat Intelligence Integration

Overview

Integration with abuse.ch (free threat intelligence) to enrich logs with IOC matching for malicious IPs, domains, hashes, and SSL certificates.

5.1 Abuse.ch Overview

Free Services:

- URLhaus: Malicious URLs
- MalwareBazaar: Malware samples
- ThreatFox: IOCs (IPs, domains, hashes)
- SSLBL: Malicious SSL certificates

API Access: Free, no authentication required for most endpoints

5.2 Elastic Abuse.ch Integration Setup

Installation

```
# On Elasticsearch/Kibana server
curl -X PUT "localhost:9200/_index_template/threat-indicators" -H 'Content-Type: appl
{
  "index_patterns": ["threat-*"],
  "template": {
    "settings": {
      "number_of_shards": 1
    },
    "mappings": {
      "properties": {
        "indicator": {"type": "keyword"},
        "type": {"type": "keyword"},
        "threat_type": {"type": "keyword"},
        "confidence": {"type": "integer"},
        "tags": {"type": "keyword"}
      }
    }
  }
}'
```

Configure Filebeat Threat Intel Module

```
sudo filebeat modules enable threatintel
```

Edit /etc/filebeat/modules.d/threatintel.yml:

```
- module: threatintel
  abuseurl:
    enabled: true
    var.input: httpjson
    var.url: https://urlhaus-api.abuse.ch/v1/urls/recent/
    var.interval: 1h

  abusech:
    enabled: true
    var.input: httpjson
    var.url: https://sslbl.abuse.ch/blacklist/sslblacklist.csv
    var.interval: 6h

  malwarebazaar:
    enabled: true
    var.input: httpjson
    var.url: https://mb-api.abuse.ch/api/v1/
    var.interval: 2h

  threatfox:
    enabled: true
    var.input: httpjson
    var.url: https://threatfox-api.abuse.ch/api/v1/
    var.interval: 1h
```

```
sudo systemctl restart filebeat
```

5.3 Data Sources from Abuse.ch

1. **JA3 Fingerprints (SSLBL)**
   Endpoint: https://sslbl.abuse.ch/blacklist/ja3_fingerprints.csv
   Data: JA3 SSL/TLS fingerprints of malicious clients
   Use Case: Detect malware C2 communications based on SSL fingerprints

2. **Malware URLs (URLhaus)**
   Endpoint: https://urlhaus-api.abuse.ch/v1/urls/recent/
   Data: Recently reported malicious URLs hosting malware
   Use Case: Block/alert on access to known malware distribution sites

3. **Malware Hashes (MalwareBazaar)**
   Endpoint: https://mb-api.abuse.ch/api/v1/
   Data: File hashes (MD5, SHA256) of malware samples
   Use Case: Match file hashes in endpoint logs against known malware

4. **SSL Blacklist (SSLBL)**
   Endpoint: https://sslbl.abuse.ch/blacklist/sslblacklist.csv
   Data: SSL certificate fingerprints used by botnets/malware
   Use Case: Identify SSL certificates associated with C2 infrastructure

5. **ThreatFox IOCs**
   Endpoint: https://threatfox-api.abuse.ch/api/v1/
   Data: IPs, domains, URLs tied to malware families
   Use Case: Alert on connections to known malicious infrastructure

6. **URL Indicators (URLhaus)**
   Endpoint: https://urlhaus-api.abuse.ch/v1/urls/
   Data: Malicious URL indicators with associated tags
   Use Case: Web proxy log enrichment and alerting

5.4 IOC Matching & Enrichment

Logstash Enrichment Pipeline

Create /etc/logstash/conf.d/10-threat-intel.conf:

```
filter {
  # IP IOC matching
  if [source][ip] or [destination][ip] {
    translate {
      field => "[source][ip]"
      destination => "[threat][indicator_match]"
      dictionary_path => "/etc/logstash/threat-intel/malicious_ips.yml"
      fallback => "no_match"
    }
  }

  # Domain IOC matching
  if [dns][question][name] {
    translate {
      field => "[dns][question][name]"
      destination => "[threat][domain_match]"
      dictionary_path => "/etc/logstash/threat-intel/malicious_domains.yml"
      fallback => "no_match"
    }
  }

  # Hash IOC matching
  if [file][hash][sha256] {
    translate {
      field => "[file][hash][sha256]"
      destination => "[threat][malware_match]"
      dictionary_path => "/etc/logstash/threat-intel/malware_hashes.yml"
      fallback => "no_match"
    }
  }
}
```

Updating IOC Lists

Script to fetch abuse.ch IOCs update_iocs.sh:

```
#!/bin/bash

# Fetch ThreatFox IOCs
curl -X POST https://threatfox-api.abuse.ch/api/v1/ \
  -d '{"query":"get_iocs","days":7}' | \
  jq -r '.data[] | select(.ioc_type=="ip:port") | .ioc' > /tmp/malicious_ips.txt

# Convert to YAML for Logstash
echo "# Malicious IPs from ThreatFox" > /etc/logstash/threat-intel/malicious_ips.yml
while read ip; do
  echo "\"$ip\": \"threatfox\"" >> /etc/logstash/threat-intel/malicious_ips.yml
done < /tmp/malicious_ips.txt
```

```
# Restart Logstash
systemctl restart logstash
```

Schedule with cron:

```
# Update IOCs daily at 3 AM
0 3 * * * /path/to/update_iocs.sh
```

5.5 Detection Rules for Threat Intelligence

Rule: Malicious IP Connection

Query:

```
threat.indicator_match: threatfox OR threat.indicator_match: urlhaus
```

Settings:
Name: Connection to Known Malicious IP
Severity: High
Risk Score: 80

Rule: Malware Hash Detected

Query:

```
threat.malware_match: * and NOT threat.malware_match: no_match
```

Settings:
Name: Known Malware Hash Detected
Severity: Critical
Risk Score: 95

Rule: Malicious Domain Resolution

Query:

```
dns.question.name: * and threat.domain_match: *
```

Settings:
Name: DNS Query to Malicious Domain
Severity: High
Risk Score: 75

5.6 Kibana Dashboards for Threat Intel

Creating Threat Intelligence Dashboard
Navigate: Kibana → Dashboard → Create dashboard

Visualizations to Add:

- IOC Match Timeline
  - Type: Line chart
  - Y-axis: Count of IOC matches
  - X-axis: Timestamp
  - Split series: threat.indicator_match
- Top Malicious IPs
  - Type: Data table
  - Metrics: Count
  - Bucket: Terms on source.ip
  - Filter: threat.indicator_match exists
- Malware Family Distribution
  - Type: Pie chart
  - Slice by: threat.malware_family
- Threat Source Breakdown
  - Type: Tag cloud
  - Terms: threat.feed_name

5.7 API Integration Examples

Query ThreatFox API

```
curl -X POST https://threatfox-api.abuse.ch/api/v1/ \
  -H "Content-Type: application/json" \
  -d '{
    "query": "search_ioc",
    "search_term": "192.168.1.100"
  }'
```

Query URLhaus API

```
curl -X POST https://urlhaus-api.abuse.ch/v1/url/ \
  -d "url=http://malicious-site.com"
```

Query MalwareBazaar API

```
curl -X POST https://mb-api.abuse.ch/api/v1/ \
  -H "Content-Type: application/json" \
  -d '{
    "query": "get_info",
    "hash": "275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f"
  }'
```

5.8 Supported Use Cases

1. **Threat Detection**
   Alert on IOC matches in network logs
   Identify compromised hosts communicating with C2
   Detect malware by file hash matching

2. **Threat Hunting**
   Proactive searches for IOCs in historical data
   Pivot on threat intelligence to find related activity
   Correlate multiple indicators across timeframes

3. **Alert Enrichment**
   Add threat context to security alerts
   Link to threat intelligence sources for investigation
   Provide malware family and campaign information

4. **Dashboard Monitoring**
   Real-time IOC match tracking

- Threat trend analysis over time
- Geographic distribution of threats

5.9 Integration Testing

Test IOC Detection

```
# Simulate connection to malicious IP (from test VM)
curl http://KNOWN_MALICIOUS_IP

# Check Kibana for alert
# Navigate to: Security → Alerts
# Filter: threat.indicator_match exists
```

Verify Enrichment

Query Elasticsearch:

```
curl -X GET "http://localhost:9200/filebeat-*/_search?pretty" \
  -u elastic:PASSWORD \
  -H "Content-Type: application/json" \
  -d '{
    "query": {
      "exists": { "field": "threat.indicator_match" }
    }
  }'
```

Part 6: Recent Security Alerts

Recent high-severity incidents detected by the platform, exported from the SIEM.

Incident Rule Name	Host	OS Family	Timestamp (UTC)	MITRE Tactic	MITRE Technique
Macro-Enabled Office Download via PowerShell	desktop-ulq0t0n	windows	2025-12-08T23:53:10.292Z	Initial Access (TA0001)	Phishing (T1566)
Word Spawning Command Shell With External IP	desktop-ulq0t0n	windows	2025-12-08T23:53:56.744Z	Initial Access (TA0001)	Phishing (T1566)

Incident Rule Name	Host	OS Family	Timestamp (UTC)	MITRE Tactic	MITRE Technique
PowerShell Spawned from Command Prompt	desktop-ulq0t0n	windows	2025-12-08T23:54:37.185Z	Execution (TA0002)	Command/Scripting (T1059)
Keybd_Event Win+R Keyboard Simulation	desktop-ulq0t0n	windows	2025-12-08T23:54:37.194Z	Initial Access (TA0001)	Phishing (T1566)
PowerShell Encoded Command Execution	desktop-ulq0t0n	windows	2025-12-08T23:54:43.219Z	Execution (TA0002)	Command/Scripting (T1059)
APT-36 Sequence	desktop-ulq0t0n	windows	2025-12-08T23:55:20.805Z	Multiple	Multiple (Sequence)
Creation of Hidden System File via PowerShell	desktop-ulq0t0n	windows	2025-12-08T23:55:56.366Z	Defense Evasion (TA0005)	Hide Artifacts (T1564)
Masquerading via Add-Type OutputAssembly	desktop-ulq0t0n	windows	2025-12-08T23:56:10.326Z	Defense Evasion (TA0005)	Masquerading (T1036)
PowerShell Invocation of Encoded VBS Macro	desktop-ulq0t0n	windows	2025-12-08T23:57:56.493Z	Execution (TA0002)	Command/Scripting (T1059)
Creation of Hidden System File via PowerShell	desktop-ulq0t0n	windows	2025-12-08T23:58:56.661Z	Defense Evasion (TA0005)	Hide Artifacts (T1564)
PowerShell Decrypting Payload via SecureString	desktop-ulq0t0n	windows	2025-12-08T23:59:58.692Z	Defense Evasion (TA0005)	Obfuscated Files (T1027)
PowerShell Encoded Payload Execution	desktop-ulq0t0n	windows	2025-12-09T00:00:00.572Z	Defense Evasion (TA0005)	Obfuscated Files (T1027)
RunMRU Registry Modification for Mshta	desktop-ulq0t0n	windows	2025-12-09T00:00:57.598Z	Execution (TA0002)	User Execution (T1204)
Excel4MacroSheets XLM Creation	desktop-ulq0t0n	windows	2025-12-09T00:01:57.792Z	Execution (TA0002)	User Execution (T1204)
LNK File Download via PowerShell	desktop-ulq0t0n	windows	2025-12-09T00:01:58.658Z	Execution (TA0002)	User Execution (T1204)
Word Spawning Command Shell With External IP	desktop-ulq0t0n	windows	2025-12-09T00:02:57.412Z	Initial Access (TA0001)	Phishing (T1566)
Invoke-MalDoc VBS Macro Execution	desktop-ulq0t0n	windows	2025-12-09T00:02:58.416Z	Execution (TA0002)	User Execution (T1204)

References

Official Documentation

Elastic Stack
Elasticsearch Documentation:
https://www.elastic.co/guide/en/elasticsearch/reference/current/index.html
Logstash Reference: https://www.elastic.co/guide/en/logstash/current/index.html
Kibana Guide: https://www.elastic.co/guide/en/kibana/current/index.html
Elastic Security: https://www.elastic.co/guide/en/security/current/index.html
Detection Rules API: https://www.elastic.co/guide/en/security/current/rule-api-overview.html
ECS Field Reference: https://www.elastic.co/guide/en/ecs/current/ecs-field-reference.html

Beats
Filebeat Reference: https://www.elastic.co/guide/en/beats/filebeat/current/index.html
Winlogbeat Documentation:
https://www.elastic.co/guide/en/beats/winlogbeat/current/index.html
Auditbeat Guide: https://www.elastic.co/guide/en/beats/auditbeat/current/index.html
Beats Platform: https://www.elastic.co/guide/en/beats/libbeat/current/index.html

Threat Intelligence
Abuse.ch Main Site: https://abuse.ch/
URLhaus Documentation: https://urlhaus.abuse.ch/api/
ThreatFox API: https://threatfox.abuse.ch/api/
MalwareBazaar API: https://bazaar.abuse.ch/api/
SSLBL Information: https://sslbl.abuse.ch/
Elastic Threat Intel Module: https://www.elastic.co/guide/en/beats/filebeat/current/filebeat-module-threatintel.html

MITRE ATT&CK
MITRE ATT&CK Framework: https://attack.mitre.org/
ATT&CK Navigator: https://mitre-attack.github.io/attack-navigator/
APT28 Profile: https://attack.mitre.org/groups/G0007/
APT36 Profile: https://attack.mitre.org/groups/G0134/

Adversary Simulation
Caldera Documentation: https://caldera.readthedocs.io/en/latest/
Caldera GitHub: https://github.com/mitre/caldera
Caldera Plugin Guide: https://caldera.readthedocs.io/en/latest/Plugin-library.html

Machine Learning
Hugging Face Transformers: https://huggingface.co/docs/transformers/index
DistilBERT Paper: https://arxiv.org/abs/1910.01108

PyTorch Documentation: https://pytorch.org/docs/stable/index.html
Phishing Detection Models: https://huggingface.co/models?search=phishing

AWS

AWS CloudTrail User Guide: https://docs.aws.amazon.com/cloudtrail/
CloudTrail Log File Examples:
https://docs.aws.amazon.com/awscloudtrail/latest/userguide/cloudtrail-log-file-examples.html
AWS IAM Best Practices: https://docs.aws.amazon.com/IAM/latest/UserGuide/best-practices.html

System Configuration

Postfix Documentation: http://www.postfix.org/documentation.html
pfSense Documentation: https://docs.netgate.com/pfsense/en/latest/
Ubuntu Server Guide: https://ubuntu.com/server/docs
Windows Event Log Reference: https://learn.microsoft.com/en-us/windows/security/threat-protection/auditing/

Security Tools

Sysmon Documentation: https://learn.microsoft.com/en-us/sysinternals/downloads/sysmon
MaxMind GeoLite2: https://dev.maxmind.com/geoip/geolite2-free-geolocation-data
Watchdog Python Library: https://python-watchdog.readthedocs.io/

Additional Resources

SIGMA Rules Repository: https://github.com/SigmaHQ/sigma
Detection Lab: https://github.com/clong/DetectionLab
Atomic Red Team: https://github.com/redcanaryco/atomic-red-team

Quick Reference Commands

Elasticsearch

```
# Check cluster health
curl -X GET "localhost:9200/_cluster/health?pretty"

# List all indices
curl -X GET "localhost:9200/_cat/indices?v"

# Search logs
curl -X GET "localhost:9200/filebeat-*/_search?pretty"
```

Kibana API

```
# Authentication
-u elastic:PASSWORD
```

```
# List detection rules
curl -X GET "localhost:5601/api/detection_engine/rules/_find" -H "kbn-xsrf: true"

# Get alerts
curl -X GET "localhost:5601/api/detection_engine/signals" -H "kbn-xsrf: true"
```

Service Management

```
# Linux services
sudo systemctl status elasticsearch
sudo systemctl restart logstash
sudo systemctl stop kibana

# Windows services
Get-Service winlogbeat
Restart-Service winlogbeat
```

End of Documentation
Version: 1.0
Last Updated: December 2024
Platform Version: ELK Stack 8.17.0