Detection

- IDS
    - Intrusion Detection System (signature based (eg. snort) or behaviour based).
    - Snort/Suricata/YARA rule writing
    - Host-based Intrusion Detection System (eg. OSSEC)
- SIEM
    - Security Information and Event Management.
- IOC
    - Indicator of compromise (often shared amongst orgs/groups).
    - Specific details (e.g. IP addresses, hashes, domains)
- Security Signals
    - Things that create signals
        - Honeypots, snort.
    - Things that triage signals
        - SIEM, eg splunk.
    - Things that will alert a human
        - Automatic triage of collated logs, machine learning.
        - Notifications and analyst fatigue.
        - Systems that make it easy to decide if alert is actual hacks or not.
- Signatures
    - Host-based signatures
        - Eg changes to the registry, files created or modified.
        - Strings in found in malware samples appearing in binaries installed on hosts (/Antivirus).
    - Network signatures
        - Eg checking DNS records for attempts to contact C2 (command and control) servers.
- Anomaly or Behavior Based Detection
    - IDS learns model of “normal” behaviour, then can detect things that deviate too far from normal - eg unusual urls being accessed, user specific- login times / usual work hours, normal files accessed.
    - Can also look for things that a hacker might specifically do (eg, HISTFILE commands, accessing /proc).
    - If someone is inside the network- If action could be suspicious, increase log verbosity for that user.
- Firewall Rules
    - Brute force (trying to log in with a lot of failures).

    - Detecting port scanning (could look for TCP SYN packets with no following SYN ACK/ half connections).
    - Antivirus software notifications.
    - Large amounts of upload traffic.

- Honeypots
    - Canary tokens.
    - Dummy internal service / web server, can check traffic, see what attacker tries.

- Things to Know About Attackers
    - Slow attacks are harder to detect.
    - Attacker can spoof packets that look like other types of attacks, deliberately create a lot of noise.
    - Attacker can spoof IP address sending packets, but can check TTL of packets and TTL of reverse lookup to find spoofed addresses.
    - Correlating IPs with physical location (is difficult and inaccurate often).

- Logs to Look at
    - DNS queries to suspicious domains.
    - HTTP headers could contain wonky information.
    - Metadata of files (eg. author of file) (more forensics?).
    - Traffic volume.
    - Traffic patterns.
    - Execution logs.

- Detection Related Tools
    - Splunk.
    - Arcsight.
    - Qradar.
    - Darktrace.
    - Tcpdump.
    - Wireshark.
    - Zeek.

- A curated list of awesome threat detection resources

Intrusion Detection System (IDS):

An IDS is designed to monitor a network or system for **abnormal activities and potential intrusions or attacks**. There are two main methods of intrusion detection: **Signature-Based** Detection and **Behavior-Based** Detection.

- Signature-Based IDS:
  - Signature-based detection identifies **known attack patterns**. A signature refers to a rule or pattern that defines the unique characteristics of an attack, which is stored in a database. The IDS checks whether events or traffic in the system **match any known signatures**.
  - Advantages: It can detect known attacks **quickly and accurately**.
  - Disadvantages: It **struggles to detect new or zero-day attacks**. If an attacker **can bypass** the signature, the system won't detect the threat.
  - Examples: **Snort and Suricata** are well-known signature-based IDS tools that analyze network traffic based on **predefined patterns**.
- Behavior-Based IDS:
  - Behavior-based detection **learns what normal system behavior** looks like and **detects deviations from these norms**. It monitors real-time activity on the network or system and flags any **anomalies as potential threats**.
  - Advantages: It can **detect new types of attacks** by recognizing abnormal behavior patterns. It's also useful for **detecting zero-day attacks**.
  - Disadvantages: It may generate **many false positives** since not all abnormal behavior is malicious.

Snort/Suricata/YARA Rule Writing:

For IDS tools like Snort and Suricata, creating detection rules is essential for defining specific patterns of behavior or signatures to identify threats.

- Snort/Suricata Rule Writing:
  - Snort and Suricata are signature-based network IDS tools. Rules written for these systems define patterns to be detected in network packets, such as **certain strings, ports, or protocols**.
  - Example rule:

```
alert tcp any any -> 192.168.1.100 80 (msg:"Possible HTTP Attack";
content:"/cmd.exe"; sid:1001;)
```

This rule triggers an alert if any packet directed to IP 192.168.1.100 on port 80 contains the string /cmd.exe, which is common in certain attack types. Writing effective rules **requires a solid understanding of attack scenarios** and how to define the signatures of these attacks.

- YARA Rule Writing:
  - YARA is a tool designed to identify malware by writing rules that look for **specific patterns in files, processes, or memory**. These rules can include strings, byte patterns, and other conditions.

- Example rule:

```
rule MyMalware
{
  strings:
    $a = "malicious_string"
    $b = { 6A 40 68 00 30 00 00 6A 14 8D 91 }

  condition:
    $a or $b
}
```

This rule detects a file or memory pattern containing the defined string or byte sequence, identifying potential malware.

Host-based Intrusion Detection System (HIDS):

A HIDS focuses on monitoring and analyzing the activities **on individual hosts** (like servers or PCs), unlike network-based IDS that monitors network traffic. HIDS typically **examines log files, checks file integrity, and monitors system calls** to detect intrusions.

- OSSEC:
  - **OSSEC is an open-source HIDS** that performs real-time log analysis, file integrity monitoring, rootkit detection, and alerting. It tracks changes in critical files and directories, ensuring their integrity.
  - Key Features:
    - Log Analysis: OSSEC analyzes logs from various operating systems to detect attacks.
    - File Integrity Monitoring: It tracks changes to important files, alerting if unauthorized modifications occur.
    - Rootkit Detection: It checks for modifications or tampering with core system files and processes.

HIDS provides detailed monitoring **at the host level** and can be paired with network-based IDS for a more comprehensive security strategy.

Security Information and Event Management (SIEM)

SIEM is a comprehensive solution that provides **real-time analysis of security alerts** generated by applications and network hardware. It integrates two core functions: **Security Information Management (SIM) and Security Event Management (SEM)**. SIEM systems are vital in **detecting, analyzing, and responding** to security threats by collecting and analyzing log data from a wide range of sources across an organization's infrastructure.

Here's a breakdown of key aspects of SIEM:

1. Core Functions of SIEM:

- **Data Collection**: SIEM collects logs and event data from a variety of sources, such as firewalls, servers, applications, network devices, and intrusion detection/prevention systems (IDS/IPS). These logs can provide critical information about system health and potential threats.
- **Correlation**: SIEM systems correlate log and event data from different sources to identify patterns or relationships that might indicate a security issue. This correlation helps to detect complex attacks that might not be obvious when viewing data from just one source.
- **Alerting**: When SIEM detects suspicious behavior or matches event data with predefined rules (such as indicators of compromise), it generates alerts to notify security teams of potential threats. These alerts can vary in severity, helping prioritize responses.
- **Dashboards and Reporting**: SIEM systems provide dashboards for monitoring real-time security events and generating reports to comply with regulations or provide insights into security posture. These reports are essential for audits and regulatory compliance (e.g., HIPAA, PCI-DSS).
- **Incident Management and Response**: SIEM helps with tracking security incidents from detection to resolution. Many SIEMs integrate with Security Orchestration, Automation, and Response (SOAR) platforms to automate parts of the incident response process, making response faster and more efficient.

2. SIEM Data Sources:

- **Log Data**: This includes system, application, and network device logs. For example, logs from firewalls, VPNs, IDS/IPS systems, antivirus software, and other endpoints provide critical security information.
- **Network Traffic Data**: SIEM systems monitor network traffic for abnormal patterns or signs of malicious activity, such as high data transfers from a single device or unusual port activity.
- **Security Alerts**: IDS/IPS systems, malware detection tools, and other security infrastructure generate alerts that SIEM systems analyze.
- **Threat Intelligence**: SIEM solutions can integrate with external threat intelligence feeds, allowing the system to detect known threats like IP addresses associated with malicious actors or attack signatures.

3. Popular SIEM Tools:

- Splunk: One of the most widely used SIEM tools, Splunk provides advanced data indexing and search capabilities. It is known for its flexibility and the ability to integrate with a wide range of data sources. Splunk ES (Enterprise Security) is specifically designed for security use cases.

- IBM QRadar: QRadar provides log and event data collection, correlation, and alerting features, as well as strong integration with threat intelligence sources. It's well-known for its scalability.
- ArcSight: Micro Focus ArcSight is another well-established SIEM platform that focuses on event correlation and security analytics.
- AlienVault (AT&T Cybersecurity): This SIEM combines log management, threat detection, and incident response in one platform and integrates with external threat intelligence feeds.

4. SIEM Use Cases:

- **Real-time Threat Detection**: SIEM helps detect ongoing attacks by analyzing live data. For example, it can detect brute force attacks, privilege escalations, or unauthorized data exfiltration.
- **Incident Investigation**: By consolidating logs and providing historical data analysis, SIEM makes it easier for security teams to investigate incidents, find the root cause, and prevent future occurrences.
- **Compliance Monitoring**: SIEM systems are crucial in meeting regulatory requirements like PCI DSS, GDPR, HIPAA, and others, as they can track and report security controls and actions over time.
- **Forensic Analysis**: When a breach or attack is discovered, SIEM provides the data needed for post-incident analysis, allowing security teams to trace back through logs to see what happened and how to mitigate similar threats in the future.

5. Challenges of SIEM:

- Complexity: SIEM systems can be complex to deploy and configure properly. They require a lot of fine-tuning to avoid issues such as alert fatigue (too many false positives).
- Resource Intensive: SIEM solutions can demand significant resources in terms of storage, processing power, and staff expertise to manage effectively.
- Data Overload: The large volumes of data collected by a SIEM can sometimes overwhelm teams, making it challenging to focus on critical threats without proper filtering and correlation rules.

6. SIEM and Machine Learning:

- Many modern SIEM systems are **incorporating machine learning (ML) and artificial intelligence (AI) to improve threat detection**. These algorithms can learn normal patterns in a network and detect anomalies without relying solely on predefined signatures or rules. This enhances the ability to **detect zero-day attacks and advanced persistent threats (APTs)**.

> SIEM systems are central to modern security operations, offering organizations a bird's eye view of their security posture and helping identify and respond to threats in real-time.

IOC (Indicator of Compromise)

IOC (Indicator of Compromise) refers to pieces of forensic data that indicate a security breach or malicious activity has occurred within a network or system. **IOCs are critical in cybersecurity as they help identify the presence of an attacker or a threat in an organization's infrastructure, allowing for timely detection, mitigation, and response to security incidents.**

Key Points About IOCs:

1. Indicators of Compromise (IOC) are often shared amongst organizations or security groups to **help others detect and defend against the same or similar threats**. By exchanging IOCs, organizations can improve collective threat intelligence and strengthen their security posture.
2. Specific Details: IOCs typically consist of specific technical details that can be used to identify signs of malicious activity, such as:

- **IP Addresses**: Malicious or suspicious IP addresses that may be associated with attackers or command-and-control (C2) servers.
- **File Hashes**: Cryptographic hashes (e.g., MD5, SHA256) of malware files or suspicious executables used by attackers.
- **Domain Names**: Domains or URLs linked to malicious infrastructure or phishing campaigns.
- **File Names**: Specific filenames that attackers use for malware, payloads, or tools.
- **Registry Changes**: Registry keys or values altered by malware or malicious software on Windows systems.
- **Email Addresses**: Email addresses used in phishing campaigns or for delivering malicious content.
- **Timestamps**: Specific timestamps when malicious activity is suspected to have occurred.
- **Processes**: Abnormal or malicious processes running on a system that indicate compromise.

Example of Common IOCs:

- Suspicious IP Address: 192.168.1.100 flagged in multiple attack logs.
- File Hash: 9aaf3e8e9f7c450d7fd9d87d8d4d3bfa (MD5 hash) linked to known ransomware.
- Malicious Domain: maliciousdomain.com associated with phishing or malware delivery.

Why IOCs Are Important:

- **Early Detection**: IOCs can help detect a security breach early, before significant damage is done. By monitoring for these indicators, organizations can identify potential threats and stop them before they escalate.
- **Incident Response**: IOCs are vital during an incident response process, as they allow teams to track down the cause of the breach, remove the threat, and prevent further damage.
- **Threat Intelligence**: Sharing IOCs across organizations contributes to threat intelligence, allowing others to use the same IOCs to detect and mitigate threats in their own environments.

How IOCs Are Used:

- Monitoring Systems: Security tools such as intrusion detection systems (IDS), firewalls, and SIEM (Security Information and Event Management) platforms are configured to detect and alert on known IOCs.

- **IOC Sharing**: Organizations often participate in threat intelligence sharing communities (e.g., ISACs, government agencies, or industry groups) to distribute IOCs quickly and widely.
- Automation: IOCs can be automated through security tools to block traffic, flag malicious files, or detect abnormal activity based on known indicators.

Summary:

An IOC (Indicator of Compromise) is a piece of evidence or data that signifies potential malicious activity, such as suspicious IP addresses, file hashes, or domains. **These indicators are commonly shared among organizations to improve detection and defense against cyber threats.** IOCs are critical for early detection, incident response, and threat intelligence, helping organizations detect, respond to, and prevent further compromises.

Security Signals

Three categories, focused on how different tools and systems generate, triage, and alert based on security signals:

1. Things That Create Signals

These are **systems and tools designed to generate raw data or signals that could indicate potential security incidents.** They monitor network traffic, system activity, or user behavior to detect anomalies, threats, or suspicious activity.

- **Honeypots**: Honeypots are decoy systems set up to attract and detect attackers. They mimic real systems and services to trick attackers into interacting with them. When attackers engage with a honeypot, it generates alerts and logs that provide insights into their tactics and techniques, creating valuable signals for further analysis.
- **Snort**: Snort is an open-source intrusion detection system (IDS) and intrusion prevention system (IPS). It inspects network traffic in real time and uses predefined rules to detect potential threats. Snort generates signals (alerts or logs) when it detects suspicious traffic, such as a known attack signature or abnormal behavior.

These tools create raw signals that form the basis for further analysis by triage systems.

2. Things That Triage Signals:

Triage tools collect, correlate, and prioritize signals from multiple sources to identify which ones need immediate attention. These systems help security teams manage the vast amount of data generated by security tools and focus on the most critical threats.

- **SIEM (Security Information and Event Management)**: A SIEM, like Splunk, collects and aggregates logs and security data from various sources across the network (e.g., honeypots, IDS/IPS systems, firewalls). **SIEMs perform real-time analysis and correlation of events to detect potential incidents.** They prioritize alerts based on severity, risk, and context, enabling security teams to investigate the most important ones.
- Example:
    - Splunk: Splunk collects and analyzes log data from various sources and uses its search and reporting features to identify patterns, anomalies, and potential threats. It helps triage large volumes of data into actionable alerts.

The triage process **helps filter out false positives and ensures that only relevant security events are escalated for further action.**

3. Things That Will Alert a Human:

These tools and systems take the signals that have been triaged and **notify human analysts about potential threats.** They often employ automation, machine learning, and other techniques to reduce the burden on security teams.

- Automatic Triage and Machine Learning: **Machine learning models can be applied to the collated logs and signals to automatically triage and detect patterns that might not be visible to rule-**

based systems. These systems help **reduce false positives and flag suspicious activity** that warrants human investigation. For example, machine learning might help detect anomalous login patterns, suspicious behavior, or previously unseen attack vectors.

- These systems are designed to prioritize critical alerts and **reduce "noise,"** making it easier for security analysts to focus on genuine threats.
- Notifications and Analyst Fatigue: Continuous alerts can lead to alert fatigue, where security analysts become overwhelmed by the volume of notifications, potentially leading to missed threats. Modern systems aim to **minimize this by filtering irrelevant alerts and only notifying analysts when a true threat is suspected.**
- Systems That Help Analysts Decide: Tools that combine machine learning, context from threat intelligence, and past incident data **help security teams quickly decide whether an alert represents an actual hack or a false positive.** These systems often provide additional context, such as related events, indicators of compromise (IOCs), or historical patterns, to help analysts make informed decisions.

Summary:

1. Things that create signals: Tools like honeypots and Snort generate raw security signals by monitoring traffic and user behavior.
2. Things that triage signals: Systems like SIEMs (e.g., Splunk) aggregate and prioritize these signals, correlating them with other events to focus on the most significant threats.
3. Things that alert a human: Tools that use automatic triage, machine learning, and smart notification systems alert human analysts only when there's a high probability of an actual threat, reducing alert fatigue and making it easier to decide if an alert is a real attack.

Signatures

Signatures in cybersecurity refer to **predefined patterns or characteristics that are used to identify malicious activity or threats in a system**. Signatures **can be applied to both host-based and network-based environments to detect indicators of compromise (IOCs) or known malicious behavior**.

1. Host-Based Signatures:

These signatures **focus on detecting suspicious activity or changes within a particular host** (e.g., a computer or server). Host-based signatures are often **used by antivirus software, endpoint detection tools, and host-based intrusion detection systems (HIDS)**.

- Examples of Host-Based Signatures:
  - **Registry Changes**: Certain types of malware modify system registries in Windows to maintain persistence (e.g., adding an entry to auto-start during boot). A host-based signature might look for specific changes to critical registry paths, such as those in HKEY_LOCAL_MACHINE or HKEY_CURRENT_USER, which are commonly targeted by malware.
    - Example: A registry key being added to ensure the malware starts every time the system boots.
  - **File Creation or Modification**: Malware often creates or modifies files on a host system. Host-based signatures can detect unusual file creation (e.g., in system directories or hidden folders) or modifications to system files (e.g., changes to explorer.exe or other critical files).
    - Example: A signature that detects the creation of specific malicious files in the Windows System32 directory.
  - **Malware Strings**: Signatures can detect known strings or code fragments commonly found in malware samples. These signatures match certain sequences of bytes or text that are characteristic of a specific malware strain.
    - Example: An antivirus solution might search for strings in binaries that are known to belong to a particular piece of malware, such as ransomware or a keylogger.

Host-Based Signatures are commonly used by antivirus programs, host-based intrusion detection systems (HIDS), and other endpoint security solutions. These signatures work by matching known patterns of malicious behavior to detect threats at the device level.

2. Network-Based Signatures:

Network signatures **detect malicious or abnormal activity in network traffic**. They are **typically used by network-based intrusion detection systems (NIDS), firewalls, or other network security tools to monitor for known attack patterns**.

- Examples of Network-Based Signatures:
  - **DNS Record Checking**: Some malware, particularly those involved in command and control (C2) operations, communicates with remote servers using domain names. A network-based signature might look for DNS queries that are attempting to resolve domain names linked to known malicious infrastructure, such as C2 servers.
    - Example: A signature that detects DNS requests to domains associated with a botnet or malware family (e.g., domains used by a specific ransomware campaign for communication).

    - **Suspicious Network Traffic**: Network-based signatures might monitor for traffic patterns associated with known attack techniques, such as scanning activity, or attempts to exploit vulnerabilities over the network.
        - Example: A signature that flags large amounts of outbound traffic from a server to an unusual IP address as indicative of data exfiltration.
    - **Protocol Misuse**: Network signatures can identify anomalies in how network protocols are used, such as deviations in packet structure or unexpected sequences, which could indicate an attack.
        - Example: Detection of malformed HTTP requests designed to exploit vulnerabilities in web servers.

Network-Based Signatures are utilized in systems like NIDS (Network Intrusion Detection Systems) or firewalls to monitor traffic between devices and across network boundaries, identifying threats based on known patterns of malicious network behavior.

Summary:

- Host-Based Signatures detect malicious activity directly on the host, such as registry changes, file creation/modification, or known malware strings in binaries. Tools like antivirus software and HIDS typically use these signatures.
- Network-Based Signatures detect suspicious activity at the network level, such as checking DNS records for communications with C2 servers or detecting unusual network traffic. These signatures are used by NIDS, firewalls, and other network security tools.

Both types of signatures are **essential for identifying and responding to threats, whether they manifest at the host or network level**.

Anomaly or Behavior-based Detection

Anomaly or behavior-based detection **focuses on identifying abnormal activities by comparing current actions against an established model of "normal" behavior**. This method allows security systems to detect suspicious activity, even if it doesn't match known attack signatures. By understanding the typical behavior of users, devices, or networks, **anomaly-based detection can spot potential threats based on deviations from expected patterns**.

Key Concepts:

1. **Learning Normal Behavior**:

- In anomaly-based intrusion detection systems (IDS), the system **first learns a baseline of normal behavior by monitoring patterns such as typical network traffic, login times, accessed files, URLs, and user activity**.
- Once the model of normal behavior is established, the system **can flag activities that deviate too far from this baseline**.
- Examples:
    - Unusual URLs: If a user or system starts accessing URLs that are highly unusual compared to their normal browsing behavior, this could indicate malicious intent (e.g., contacting command-and-control servers or downloading malware).
    - User-Specific Activity: If a user typically logs in during regular work hours and suddenly logs in during odd hours or accesses files they don't usually handle, these deviations can trigger alerts. This could be a sign of a compromised account or insider threat.

2. **Detecting Actions Associated with Hackers**:

- Anomaly-based systems also look for specific actions that are often linked to malicious behavior or hacker activity. **Even if these actions don't directly match known signatures, their unusual nature can still be detected**.
- Examples:
    - HISTFILE Commands: If a user executes commands that interact with HISTFILE, it could indicate an attempt to erase command history, a tactic often used by attackers to hide their tracks.
    - Accessing /proc Files: /proc is a special directory in Unix/Linux that contains information about system processes. Unusual access to /proc files by non-administrative users could indicate an attempt to gather sensitive information or exploit system vulnerabilities.

3. **Detecting Insider Threats**:

- If an attacker has gained access to the network (whether as an outsider or as an insider threat), anomaly-based systems are well-suited to detect abnormal behavior that could indicate malicious intent.
- For example, if a user begins to access files, systems, or applications outside of their normal scope, or behaves inconsistently with their usual patterns, the system will flag these as suspicious actions.

Response:

- **When a suspicious activity is detected, the system may increase log verbosity for that user or system to gather more detailed information about their actions**. This allows security analysts to conduct a more thorough investigation into whether the behavior indicates a breach.

- Example Scenarios:

    - Unusual Login Times: If an employee usually logs in between 9 AM and 5 PM but suddenly logs in at 3 AM, this deviation from normal behavior would trigger an alert.
    - Accessing Sensitive Files: A user who typically works in marketing suddenly starts accessing finance-related documents, which is out of the ordinary for their role. This behavior could indicate a compromised account or an insider threat.
    - Increased Log Verbosity: If a user starts accessing sensitive directories or executing commands linked to privilege escalation, the system might increase the log verbosity for that user, capturing more detailed logs for closer analysis.

Benefits of Anomaly / Behavior-Based Detection:

- **Detecting Unknown Threats**: Unlike signature-based detection, which relies on known patterns of malicious activity, anomaly-based detection **can detect new or previously unseen threats by identifying deviations from normal behavior**.

- **Insider Threat Detection**: This method is particularly effective in identifying insider threats or attackers who have gained access to the network, as their actions may stand out from regular users' activity.

- **Adaptive Security**: As systems and users evolve, the model of "normal" behavior can be continuously updated, allowing the IDS to adapt to changes over time.

- Challenges:

    - **False Positives**: If the definition of "normal" is too strict, anomaly detection systems can generate a lot of false positives, flagging benign activities as suspicious. This can lead to alert fatigue for security teams.
    - **Learning Period**: These systems require a period of time to learn normal behavior before they can effectively detect anomalies. During this learning phase, they may not detect all threats.
    - **Context Required**: Anomaly-based systems might detect suspicious activity, but human analysts often need to investigate further to determine whether the activity is genuinely malicious or just unusual but harmless.

Summary:

- Anomaly/Behavior-Based Detection relies on learning what "normal" behavior looks like in a system or network and detecting activities that deviate from that baseline.
- It can detect things like unusual URLs being accessed, strange login times, or file access patterns that don't match a user's usual behavior.
- It can also spot specific hacker tactics, such as commands to manipulate HISTFILE or accessing sensitive directories like /proc.

- When suspicious activity is detected, the system can increase logging for the user or process in question, providing more data for further analysis. This approach is especially useful for detecting insider threats or unknown attacks.

Firewall Rules

Firewall rules are critical components in network security, **used to control incoming and outgoing traffic based on predefined security policies**. These rules can be configured to detect and block suspicious activities like brute force attacks, port scanning, large data transfers, or other indicators of compromise (IOCs). Here's how firewall rules can help with specific security concerns:

1. Brute Force Attacks (Detecting Multiple Failed Login Attempts):

- Brute force attacks involve repeated login attempts using different username-password combinations in an attempt to gain access to a system.
- Firewall Rule Example:
    - You can **set up rules to detect multiple failed login attempts within a short period**. If a certain threshold (e.g., 5 failed attempts within 1 minute) is reached, the firewall can block the IP address for a certain time to prevent further attempts.
    - Rule: If more than X failed login attempts are detected from a single IP address within Y minutes, block or throttle traffic from that IP.
        - Example:

```
Block traffic from IP 10.0.0.5 after 5 failed SSH login attempts within 2
minutes.
```

2. Detecting Port Scanning (Identifying TCP SYN Floods or Half-Open Connections):

- Port scanning is a common reconnaissance technique used by attackers to discover open ports and services on a target system. One common method of port scanning is SYN scanning, where attackers send TCP SYN packets without completing the handshake (no SYN-ACK response).
- Firewall Rule Example:
    - A rule **can be created to detect TCP SYN packets without corresponding SYN-ACK responses (half-open connections)**. This often indicates port scanning activity. Once detected, the firewall can block the source IP for a certain period or limit the connection rate from that IP.
    - Rule: If more than X half-open connections from a single IP are detected within Y seconds, block or drop further connections from that IP.
        - Example:

```
Block traffic from IP 192.168.1.20 if 50 SYN packets without SYN-ACK
responses are detected within 30 seconds.
```

3. Antivirus Software Notifications:

- Firewalls can also interact with antivirus software and block traffic from infected machines or devices that trigger malware alerts.

- Firewall Rule Example: When the antivirus software detects malware or suspicious behavior on a system, a firewall rule can be triggered to automatically isolate that machine by blocking its outgoing traffic or restricting access to certain network resources.
    - Rule: If antivirus software detects malware on host X, block outgoing traffic from that host until further inspection.
        - Example:

```
Block all outbound traffic from 192.168.1.10 after a malware detection
alert from the antivirus system.
```

4. Large Amounts of Upload Traffic:

- Unusually large upload traffic from a single device could be a sign of data exfiltration, malware activity (e.g., botnets), or a compromised device attempting to send stolen information.
- Firewall Rule Example:
    - A rule can monitor upload traffic and trigger an alert or block traffic if a device exceeds a specified threshold of outgoing data within a short time frame.
    - Rule: If more than X GB of upload traffic is detected from a single IP within Y minutes, block or throttle that device's connection.
        - Example:

```
Limit upload traffic to 500 MB per minute. Block traffic from any IP that
exceeds this limit.
```

Summary of Rules:

1. **Brute Force**: Detect excessive failed login attempts and block the attacker's IP after a threshold is reached to prevent further attempts.
2. **Port Scanning**: Monitor for half-open TCP connections (SYN packets without SYN-ACK) to detect port scanning, and block the offending IP if suspicious scanning is detected.
3. **Antivirus Notifications**: Automatically block traffic from a device that has triggered an antivirus alert to prevent the spread of malware.
4. **Large Upload Traffic**: Monitor for abnormal upload activity that could indicate data exfiltration or malware, and block or limit traffic from the offending device.

> These rules can significantly enhance the security posture of a network by preventing common attack vectors and detecting unusual behavior that could indicate an ongoing attack.

Honeypots

A honeypot **is a decoy system or service designed to mimic a real system within a network to attract and detect attackers**. The primary purpose of a honeypot is to monitor and record the activities of potential intruders, allowing security teams to understand attack methods, gather intelligence, and improve defenses without exposing critical systems. Honeypots **do not contain real data or services but are configured to look authentic to attackers**.

Types of Honeypots:

1. **Low-Interaction Honeypots**: Simulate a limited set of services and are less complex, designed to **detect basic attack patterns and reconnaissance activity**.
2. **High-Interaction Honeypots**: Provide full, realistic services (like web servers, SSH, databases) to **observe a wide range of attacker behaviors**. These honeypots allow the attacker to interact with what appears to be a genuine system, giving security teams more detailed insights into their tactics and tools.

Canary Tokens:

- Canary tokens are **a specific type of honeypot technique used to detect when attackers access sensitive or decoy data**. A canary token is **a trigger that silently alerts the security team when it is accessed, modified, or used in any way by an unauthorized entity**.
- How Canary Tokens Work:
    - Canary tokens can take many forms, such as:
        - **Decoy credentials**: Fake login credentials stored in an accessible place. If an attacker attempts to use these credentials, the system immediately alerts security teams.
        - **Fake files**: A file (e.g., a Word document or PDF) that sends a notification when opened.
        - **DNS Canary Tokens**: A token linked to a specific DNS name that, if queried, triggers an alert.
    - Canary tokens **allow organizations to detect early-stage breaches**, such as when attackers are performing reconnaissance or attempting lateral movement within the network.

Example: A decoy file named passwords.txt is placed in a folder where an attacker might search for credentials. If the file is opened, the token inside the file triggers an alert to the security team, notifying them of suspicious activity.

Dummy Internal Service / Web Server:

- A dummy internal service or web server is a honeypot set up inside the organization's network to mimic real services (e.g., web servers, database servers) that are used by attackers during reconnaissance or lateral movement.
- Purpose:
    - These decoy services are configured to appear vulnerable or valuable to an attacker. They allow security teams to:
        - Monitor Traffic: Track how attackers interact with the dummy service, what tools they use, and what vulnerabilities they try to exploit.

        - Detect Attackers: When attackers attempt to access or compromise the decoy service, security teams can observe these actions in real time.
        - Study Attack Patterns: By monitoring how attackers interact with the dummy service, organizations can better understand the latest attack methods and develop stronger defenses.
    - Example: A fake internal web server running a dummy e-commerce site or a fake database that appears to hold sensitive information. These honeypots attract attackers who might be looking for internal resources to exploit or steal data.

How Honeypots and Canary Tokens Help:

- **Early Detection**: Honeypots and canary tokens can detect attackers early in the attack chain, often during the reconnaissance phase, before they reach critical systems.
- **Minimal False Positives**: Since honeypots are not meant to be accessed by legitimate users, any interaction with them is almost certainly malicious, resulting in fewer false positives compared to other detection methods.
- **Gathering Intelligence**: Honeypots allow organizations to gather valuable intelligence about an attacker's methods, tools, and behavior, which can be used to strengthen security defenses.
- **Low-Cost Deception**: Honeypots and canary tokens are often low-cost methods of deception and can be highly effective in confusing and deterring attackers.

Summary:

- Honeypots are decoy systems designed to attract attackers, providing a way to monitor and record their actions without exposing real assets. These systems can simulate internal services or web servers.
- **Canary tokens are lightweight honeypot triggers** that alert security teams when accessed, used in decoy files, fake credentials, or DNS queries to detect attackers during early stages of a breach.
- Both honeypots and canary tokens are essential tools for early detection, monitoring attacker behavior, and improving security defenses without risking real assets.

Things to Know About Attackers

Understanding attacker tactics, techniques, and procedures (TTPs) is crucial for defending against cyber threats. Attackers often use various methods to evade detection, confuse security systems, and hide their true identity. Here are some key things to know about attackers:

1. Slow Attacks Are Harder to Detect:

- **Slow and Low**: Attackers often use slow attacks (sometimes called low-and-slow attacks) to avoid triggering security alerts. Rather than launching a rapid, high-volume attack that might quickly be flagged by intrusion detection systems (IDS) or firewalls, the attacker sends a small number of malicious packets or makes subtle changes over a long period of time.
    - Example: A slow brute-force attack where the attacker attempts one or two login attempts every few minutes to avoid being noticed by systems designed to detect rapid login failures.
- **Evasion**: By spreading their activities over a longer timeframe, attackers evade threshold-based detection systems, which are more likely to miss low-volume traffic that doesn't cross alert thresholds.

2. Attackers Can Spoof Packets to Create Noise:

- **Noise Generation**: Attackers can spoof packets that mimic other types of attacks, deliberately creating a lot of noise in the network. This tactic is meant to distract or overwhelm the defenders, forcing them to deal with large amounts of fake or decoy traffic while the real attack is conducted quietly elsewhere.
    - Example: The attacker floods the network with thousands of spoofed SYN packets that simulate a Distributed Denial-of-Service (DDoS) attack. While security teams are busy mitigating the DDoS attack, the attacker is attempting a more targeted and stealthy attack, like privilege escalation or data exfiltration.
- **False Positives**: This tactic can cause false positives in security systems, making it more challenging for defenders to separate real threats from fake ones. It can exhaust the resources of security analysts or automated systems.

3. Attackers Can Spoof IP Addresses:

- **IP Spoofing**: Attackers can spoof (forge) the source IP address of packets to hide their real identity and make it look like the traffic is coming from a different device or location. This can be used to avoid detection, complicate attribution, or trigger attacks that appear to come from trusted sources.
    - Example: In a DDoS attack, the attacker may spoof the source IP addresses to make it appear as if the traffic is coming from hundreds or thousands of different systems, even though it is being orchestrated from one location.
- **Detecting IP Spoofing**:
    - TTL (Time to Live): One way to detect spoofed IP addresses is **by analyzing the TTL value in packet headers**. The TTL indicates the number of hops a packet can take before it is discarded. Comparing the TTL of the incoming packet with the TTL value obtained from a reverse lookup (the response packet) can help identify discrepancies.
        - Example: If the TTL value of a packet seems inconsistent with what is expected based on its source, it might indicate that the packet's source address has been spoofed.

    - Limitations of IP Spoofing: While IP spoofing can make attacks harder to trace, **it doesn't allow the attacker to receive responses from the spoofed packets, as the responses would be sent to the spoofed address**. This makes spoofing more useful in specific scenarios, like DDoS attacks, but less practical for attacks that require interaction.

4. Correlating IPs with Physical Location Is Difficult:

- Geolocation Challenges: Attempting to **correlate an IP address with a physical location is difficult and often inaccurate**. Attackers may use techniques such as proxy servers, VPNs, or Tor to hide their real IP address and location, making it nearly impossible to pinpoint where an attack is truly coming from.
    - IP Address Databases: Geolocation services that map IP addresses to physical locations (such as cities or countries) often use IP address databases. However, these databases are not always up-to-date, and in many cases, **IP addresses may be incorrectly associated with a location**.
    - Dynamic IPs and NAT: Some networks use NAT (Network Address Translation) or dynamic IP addressing, meaning that multiple devices share a single public IP address, or the IP address changes periodically, making it hard to tie an attack to a specific device or location.
- Proxy and VPN Evasion: Attackers frequently **route their traffic through proxies, VPNs, or anonymizing services like Tor to further obscure their location**. This can make it appear as though the attack is originating from a completely different country or region than where the attacker is truly located.

Summary:

1. **Slow Attacks Are Harder to Detect**: Attackers can use slow, subtle attacks to stay under the radar of detection systems.
2. Attackers Can **Spoof Packets**: By spoofing packets, attackers can create noise, simulate false attacks, and distract defenders from the real threat.
3. Attackers Can **Spoof IP** Addresses: Attackers can hide their true location by using spoofed IP addresses, but TTL analysis and reverse lookups can help detect discrepancies.
4. **Correlating IPs to Physical Locations Is Difficult**: Mapping IP addresses to physical locations is often inaccurate, and attackers can use VPNs, proxies, and other methods to obscure their true origin.

> Understanding these tactics helps defenders design better detection methods, mitigate threats effectively, and improve the accuracy of their response efforts.

Logs to Look at for Suspicious Activity Detection

When investigating potential security incidents, **analyzing specific types of logs is crucial for identifying malicious activities**. Logs can provide valuable insights into an attacker's behavior, help detect anomalies, and reveal the full scope of a breach. Below are important logs to monitor and analyze:

1. DNS Queries to Suspicious Domains:

- DNS query logs track requests made by devices on the network to resolve domain names into IP addresses. Monitoring these logs **can help identify when devices attempt to connect to suspicious or malicious domains**, which is often an indicator of command and control (C2) communication or malware trying to exfiltrate data.
- What to look for:
    - Domains associated with known threats: Malware often communicates with hard-coded or dynamically generated domain names. Queries to known malicious domains should trigger alerts.
    - Unusual domain patterns: Look for strange, randomized domain names (e.g., hsadf12sd.com), which may indicate DGA (Domain Generation Algorithm) usage by malware.
- Example: DNS requests for newly registered domains or domains associated with phishing campaigns could indicate an ongoing attack.

2. HTTP Headers Containing Suspicious Information:

- HTTP headers provide metadata about web requests and responses. Attackers may inject malicious or malformed data into HTTP headers to exploit vulnerabilities or carry out attacks.
- What to look for:
    - **Unusual user-agent strings**: Attackers might spoof user-agent strings or use abnormal ones to disguise malware or bots.
    - **Referrer headers**: Unexpected or suspicious referrers could indicate phishing or redirection to malicious sites.
    - **Custom or missing headers**: Some malware may use non-standard headers or omit headers that are expected in normal traffic.
- Example: HTTP requests with overly long or malformed headers (buffer overflow attempts) or requests missing common headers like User-Agent can signal malicious activity.

3. Metadata of Files (Forensic Focus):

- File metadata logs store information such as the file's author, creation time, and modification details. **In a forensic investigation, this data can help track how files were created, modified, or distributed**.
- What to look for:
    - **Unusual file author**: Metadata showing an unexpected or suspicious file author might indicate tampered or malicious files.
    - **Creation and modification times**: Inconsistent timestamps or creation times that don't align with normal user activity may suggest unauthorized changes or malicious files.
    - **File origin details**: Metadata showing where a file originated (e.g., downloads from suspicious URLs) can provide critical clues.

- Example: A malicious Word document with metadata showing it was created by an attacker's tool can provide important forensic evidence.

4. Traffic Volume Logs:

- Traffic volume logs track the amount of data being sent and received by a network device. **Monitoring for spikes or drops in traffic volume can help detect data exfiltration or DDoS attacks**.
- What to look for:
    - Sudden spikes in outbound traffic: Unusual amounts of data being uploaded could indicate an ongoing data exfiltration attempt.
    - Increased inbound traffic: This could be a sign of a DDoS attack or scanning activity from an attacker probing the network.
- Example: A significant increase in traffic volume from a workstation at an unusual time could suggest an insider threat or compromised machine exfiltrating sensitive data.

5. Traffic Patterns Logs:

- Traffic pattern logs provide insights into the behavior of network traffic over time. **Analyzing these patterns can help identify suspicious activity such as scanning, lateral movement, or connections to unexpected services**.
- What to look for:
    - **Unusual connection destinations**: Devices communicating with unknown or foreign IP addresses.
    - **Odd timing patterns**: Traffic occurring during off-hours, weekends, or holidays, when legitimate activity is low.
    - **Scanning patterns**: Repeated attempts to connect to multiple ports across many IPs could indicate a port scan or lateral movement attempt.
- Example: Multiple failed connections to internal IP addresses in quick succession may be an indicator of an internal reconnaissance or lateral movement attack.

6. Execution Logs:

- Execution logs capture details about the processes and commands being executed on a system. Monitoring these logs **can help detect unauthorized or malicious activity such as malware execution or privilege escalation attempts**.
- What to look for:
    - **Suspicious or unauthorized command execution**: Unusual commands being executed (e.g., privilege escalation attempts with sudo, or running scripts from unexpected locations).
    - **Execution of known malicious files**: Log entries showing that malware binaries (e.g., ransomware.exe) have been run on the system.
    - **Abnormal process activity**: Processes that are typically not run by normal users, or processes starting at odd times, might indicate malicious activity.
- Example: Detecting the execution of PowerShell scripts or binaries downloaded from the internet outside of normal administrative tasks can be a strong indicator of compromise.

Summary:

- DNS Queries to Suspicious Domains: Look for queries to known malicious domains or **randomized domain names (DGA)**.
- HTTP Headers: Monitor for unusual or malformed HTTP headers (e.g., odd User-Agent strings or missing headers).
- Metadata of Files: Use file metadata to track suspicious file creation, modification, or distribution.
- Traffic Volume: Unusual spikes or drops in data volume could signal exfiltration or DDoS attacks.
- Traffic Patterns: Look for abnormal traffic behaviors such as off-hour connections or scanning attempts.
- Execution Logs: Analyze process execution logs for unauthorized commands, malware execution, or suspicious process behavior.

> By carefully monitoring these logs, organizations can detect potential threats early, investigate suspicious behavior, and respond to security incidents more effectively.

Detection Related Tools

In cybersecurity, **detection tools are used to monitor, analyze, and alert on potential threats in real-time or during investigations**. These tools help security teams detect anomalies, attacks, and breaches by analyzing logs, network traffic, and system behaviors. Below is an overview of some key detection-related tools and their functions:

1. Splunk:

- Type: **SIEM (Security Information and Event Management) / Log Management**
- Description: Splunk is a powerful platform for searching, monitoring, and analyzing machine-generated data, including logs, metrics, and events from various systems. It provides real-time visibility into an organization's IT infrastructure and can be used for security monitoring, threat detection, and incident response.
- Features:
    - Centralized log management and aggregation.
    - Real-time search and analysis.
    - Dashboards for visualizing security metrics and alerts.
    - Integrates with various data sources, making it versatile for detecting security incidents.

2. ArcSight:

- Type: **SIEM**
- Description: ArcSight (by Micro Focus) is an enterprise-level SIEM platform that helps organizations detect, respond to, and mitigate security threats. It collects and correlates data from various sources to provide comprehensive threat detection capabilities.
- Features:
    - Advanced correlation engine for detecting security incidents.
    - Centralized event management and real-time alerting.
    - Scalable architecture for large organizations.
    - Used to perform forensic analysis and compliance reporting.

3. QRadar:

- Type: **SIEM**
- Description: IBM QRadar is another leading SIEM tool used for log management, threat detection, and security intelligence. It analyzes security data from various sources and applies advanced analytics to detect anomalies and attacks in real-time.
- Features:
    - Collects data from logs, network flows, and user activities.
    - Uses machine learning and behavioral analytics for threat detection.
    - Automatically correlates events to provide actionable insights.
    - Supports incident investigation and forensics by correlating events across different data sources.

4. Darktrace:

- Type: AI-based **Network Detection and Response (NDR)**
- Description: Darktrace is a cybersecurity platform that uses artificial intelligence (AI) to detect threats in real-time across enterprise networks. It builds a model of “normal” network behavior and detects anomalous activity that deviates from this baseline.
- Features:
    - Uses unsupervised machine learning for detecting unknown threats.
    - Monitors network traffic to detect internal and external threats.
    - Capable of detecting insider threats, malware, and zero-day attacks.
    - Provides autonomous responses to threats through its Antigena module, automatically containing threats.

5. Tcpdump:

- Type: **Packet Sniffer / Network Traffic Analysis**
- Description: Tcpdump is a **command-line tool** used to capture and analyze network traffic at the packet level. It is widely used by security professionals for network troubleshooting and incident investigation.
- Features:
    - Captures network packets in real-time and allows deep inspection of network traffic.
    - Supports **filtering by protocol, IP addresses, ports, etc.**
    - Useful for detecting traffic anomalies, unauthorized communications, and potential attacks.
    - Often used in conjunction with other tools like Wireshark for deeper analysis.

6. Wireshark:

- Type: **Packet Analyzer**
- Description: Wireshark is one of **the most popular and powerful GUI-based packet analyzers**. It captures and analyzes network traffic in real-time and provides a detailed view of packet-level data.
- Features:
    - Captures packets from live network traffic and provides deep inspection capabilities.
    - Supports a wide variety of network protocols, allowing detailed traffic analysis.
    - Provides filtering, searching, and visualization features to detect anomalies and investigate suspicious activities.
    - Useful for detecting attacks like man-in-the-middle, DNS poisoning, and traffic injection.

7. Zeek (formerly known as Bro):

- Type: **Network Security Monitor**
- Description: Zeek is an open-source network monitoring and traffic analysis framework designed to detect intrusions by inspecting network traffic. Unlike packet analyzers, Zeek focuses more on analyzing higher-level events and producing structured logs.
- Features:
    - Monitors network traffic and generates logs with detailed metadata about network activities.
    - Detects network intrusions by analyzing patterns of traffic behavior.
    - Provides event-based logs, which are easier to correlate and analyze than raw packet captures.
    - Useful for detecting threats like scanning, DDoS, lateral movement, and command-and-control (C2) traffic.

Summary of Detection Tools:

- Splunk, ArcSight, and QRadar: SIEM tools that aggregate and correlate logs from multiple sources, providing real-time threat detection and incident response capabilities.
- Darktrace: AI-driven platform that detects network anomalies and threats by modeling normal behavior and identifying deviations.
- Tcpdump and Wireshark: Packet capture and analysis tools used to inspect network traffic at a granular level, identifying suspicious or malicious network activity.
- Zeek: Network security monitor that analyzes traffic patterns and generates logs to detect network intrusions and suspicious activities.

> Each of these tools plays a critical role in detecting, analyzing, and responding to potential threats in a cybersecurity environment, providing insights into logs, network traffic, and system behaviors.