INCIDENT RESPONSE TRAINING

INTRODUCTION TO INCIDENT RESPONSE

This training plan is intended for analysts who have completed review of our Cyber Incident Response Plan (CISP) and Corporate Information Security Policy (CISP).

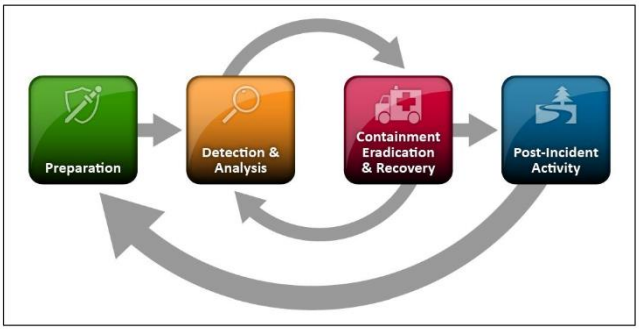
The generally accepted lifecycle for responding to incidents is structured into several key phases[1]:

1. **Preparation**: This phase involves establishing an IR policy, defining the team's roles and responsibilities, and identifying critical assets and potential threats. It includes creating incident detection and reporting procedures, as well as ensuring resources are available for effective response.

2. **Detection and Analysis**: In this phase, incidents are detected through monitoring and analysis of network traffic, system logs, and other sources. Once detected, incidents are categorized based on severity and impact, and a preliminary assessment is conducted to understand the nature and scope of the incident.

3. **Containment, Eradication, and Recovery**: The focus here is on containing the incident to prevent further damage, eradicating the threat from affected systems, and restoring operations to normalcy. This phase involves executing mitigation strategies, applying patches or fixes, and restoring data from backups if necessary.

4. **Post-Incident Activity**: After the incident is contained and systems are restored, post-incident activities include documenting the incident, conducting a lessons learned review, and updating incident response procedures based on identified weaknesses or improvements.



As a Digital Forensics and Incident Response (DFIR) Analyst, you're expected to adeptly manage every phase of the response lifecycle response lifecycle, from creating detections to writing incident reports, all with the aim of minimizing the impact of threats within the organization.

Now, let's delve deeper into a few core components of incident response: Host-based Analysis, Network-based Analysis, and Endpoint Detection and Response (EDR) / Antivirus Intrusion Detection Systems. These areas will equip you with the essential skills to effectively detect, investigate, and mitigate security incidents across a diverse digital environment.

HOST-BASED ANALYSIS

Host-based analysis is a critical aspect of cybersecurity that involves examining and investigating individual systems (hosts) to identify, understand, and mitigate security incidents. The primary purpose of host-based analysis is to detect and respond to malicious activities, investigate potential compromises, and ensure the integrity and security of host systems. This training will cover the essential techniques and tools used for host-based analysis, including log analysis, file integrity monitoring, and malware detection.

In this training, our focus will be specifically on Windows operating systems. By delving into host-based analysis on Windows OS, trainees will gain essential skills in identifying malicious activity, analyzing system artifacts, and implementing effective mitigation strategies tailored to this prevalent OS platform.

STRUCTURE OF THE WINDOWS OPERATING SYSTEM

The Windows operating system comprises several key components, each generating unique artifacts that can be crucial for forensic analysis:

FILE SYSTEM:

- **NTFS (New Technology File System):** The primary file system used in modern Windows systems. It supports various features such as encryption, permissions, and journaling.
- **FAT (File Allocation Table):** An older file system still used in some removable storage devices and legacy systems. *Out of Scope*

REGISTRY:

- The Windows registry stores configuration settings and options for the operating system and installed applications. It is divided into several hives, each containing a set of keys and values.

  Certain directories, files, and registry hives are critical for forensic analysis as they often contain artifacts related to user activity, system configuration, and installed software.

  Adversaries manipulate the Windows Registry to store hidden settings, delete traces of their activity, or ensure their malicious code runs every time the system starts. Access to different parts of the Registry depends on account privileges, with some areas requiring administrator rights. Tools like the built-in Windows command-line utility 'Reg' allow for local or remote Registry changes.

DIRECTORIES:

- **C:\Windows:** The directory where the Windows operating system files are stored.
- **C:\Users:** Contains user profiles and associated data.
- **C:\Program Files & C:\Program Files (x86):** Default directories for installed applications.
- **C:\ProgramData:** Contains application data that is shared among users.
- **C:\Windows\System32:** Critical system files and drivers necessary for Windows functionality, protected against unauthorized modifications.
- **C:\Windows\Temp (or %Temp%):** Used for temporary storage,

frequently targeted by malware for hiding and executing malicious code.

FILES:

- **Pagefile.sys:** A file used for virtual memory, which can contain fragments of memory.
- **Hiberfil.sys:** A file used when the system hibernates, which can contain the contents of RAM.
- **Prefetch Files:** Located in C:\Windows\Prefetch, these files help speed up application launch times and can provide information about program execution.

REGISTRY HIVES:

- **HKEY_LOCAL_MACHINE (HKLM):** Contains system-wide settings.
- **HKEY_CURRENT_USER (HKCU):** Contains settings for the currently logged-in user.
- **HKEY_CLASSES_ROOT (HKCR):** Stores information about registered applications.
- **HKEY_USERS (HKU):** Contains user-specific settings for all users on the system.

PROCESSES

Processes are instances of executable programs running on a computer system, each with its own memory space and resources.

In Windows, processes can spawn other processes, creating a parent-child relationship. Understanding this hierarchy is essential for tracing the execution flow of malicious activities.

1. **Parent Process:** The original process that initiates another process. For example, a command shell (cmd.exe) launching a script.
2. **Child Process:** The process that is initiated by another process. For example, a PowerShell script (powershell.exe) started by cmd.exe.

HASHES

Hashes are cryptographic representations of data and are used to verify the integrity of files. In forensics, hashes help ensure that evidence has not been tampered with.

1. **File Hashes:** Used to uniquely identify files. Common algorithms include MD5, SHA-1, and SHA-256.
2. **Process Hashes:** Hashes of executable files that help identify known good or malicious files. Comparing hashes against databases of known malware (e.g., VirusTotal) can quickly determine if a file is malicious.

USER ACCOUNTS

In Windows operating systems, user accounts serve as the primary method of identifying and authenticating users. They dictate the level of access individuals have to files, settings, and applications on a computer or across a network. Properly managing these accounts ensures that individuals have appropriate access rights aligned with their roles and responsibilities, while also mitigating the risk of unauthorized access and potential security breaches.

Here are a few account types to know:

Local User Accounts: Specific to the local machine, used for individual access with limited permissions. While local accounts exist, organizations typically favor domain

accounts for centralized management and enhanced security controls.

A local account may be indication of an attacker's attempt to carry out various objectives, such as establishing persistence, exfiltrating data, or bypassing security controls. These accounts can provide a foothold for further compromise within the local system or network environment.

Domain Accounts: Managed centrally within a Windows domain, allowing access across multiple domain-joined machines.

Attackers target domain accounts to escalate privileges and move laterally across the network, leveraging the broader access these accounts have to organizational resources.

Administrative Privileges: Accounts with elevated rights to perform system-level tasks like software installation and configuration changes.

Administrators are prime targets for attackers due to their extensive system privileges. Attackers aim to compromise these accounts to gain full control over systems and network resources. They may install backdoors, alter configurations, or exfiltrate sensitive information, posing significant threats to organizational security.

WINDOWS EVENT LOGS

Windows Event Logs are files that record information about a Windows operating system's security, system, and application events. These logs are crucial for tracking system events, security-related activities, and user actions, providing a detailed record of system and application activity. Log files can help determine the cause of a security breach or error because they record data concurrently with the system's activities. This information is essential for forensic analysis and incident response and can be manually reviewed (for example, sending to a SIEM like Splunk), or automatically ingested by security tools for monitoring.

LOG TYPES

Windows Event Logs consist of several types, each focusing on different aspects of system operation and security

APPLICATION LOGS

Contains events logged by applications or programs running on the system; these can be events related to application errors, warnings, and informational messages.

Application logs are useful for identifying issues within specific applications, such as crashes, unexpected behavior, or application-specific errors.

SYSTEM LOGS

Events related to system stability, performance, and hardware interactions logged by Windows system components.

Examples:

- A system service failing to start.
- A device driver encountered an error.
- System boot and shutdown events.

These logs help in identifying unauthorized changes or failures induced by malicious activities.

SECURITY LOGS

Windows Security Event Logs are vital tools for an analyst. They store all the pertinent information concerning the security of the operating system. These logs include a wide range of information, such as successful and failed login attempts, system changes, account modifications, and policy updates. Each event recorded in the logs contains information about the event type, the date and time it occurred, the user involved, and the success or failure of the event.

Security logs are important because they provide a chronological record of activities that can help trace and understand the

sequence of actions during a security incident. By analyzing these logs, analysts can identify patterns of malicious behavior, detect breaches, investigate the scope of an attack, and understand the methods used by attackers. Additionally, many alerts and detections are built solely from the logging available in Seurity Logs, making **Preparation** and **Detection** phases possible. Together, this information is essential for effectively responding to incidents, mitigating damage, and enhancing the security posture of the organization to prevent future attacks.

Event IDs & Timestamps

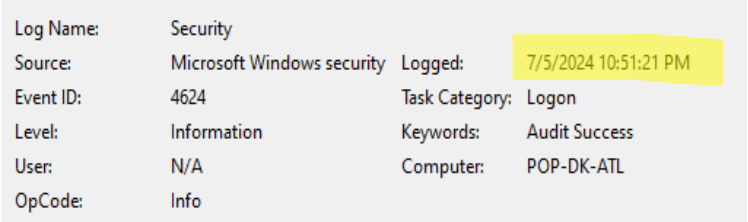
Event IDs in Windows Security Event Logs are unique identifiers assigned to specific types of events recorded by the system. Each event ID corresponds to a particular event type, such as a user logging in, a file being accessed, or a network connection being established. These identifiers help categorize and filter events for more straightforward analysis.

Event ID	What it means
4624	Successful account log on
4625	Failed account log on
4634	An account logged off
4648	A logon attempt was made with explicit credentials
4719	System audit policy was changed.
4964	A special group has been assigned to a new log on
1102	Audit log was cleared. This can relate to a potential attack
4720	A user account was created
4722	A user account was enabled
4723	An attempt was made to change the password of an account
4725	A user account was disabled

For example, Event ID 4624 indicates a successful logon, while Event ID 4740 denotes a user account being locked out after too many failed login attempts.

Timestamps in event logs record the exact date and time when an event occurred. This temporal data is critical for creating a timeline of security events, which is essential for understanding the sequence and timing of actions during a security incident.

For example, in the following image you can see that I had a successful logon to my workstation around 10:51:21PM ET on July 5, 2024 or about 3:51:21 UTC



By analyzing Event IDs and timestamps across different systems and logs, analysts can correlate seemingly unrelated events to uncover patterns of malicious activities and lateral movements within the network.

Common Areas of Analysis

There are common analysis methods in Windows Security Event Logging that involve scrutinizing specific types of events to uncover suspicious activities or unauthorized access within a network. Each type of analysis targets a different aspect of system interaction and user behavior.

Logon and Logoff (**Event ID 4624 & 4634)** Analysis is essential for detecting anomalies in user access patterns. By examining the times and frequencies of user logons and logoffs, along with the success or failure of these attempts, analysts can identify potential unauthorized access attempts or brute force attacks. For instance, logons at unusual hours or high frequencies of failed logon attempts (**4740)** can be indicators of compromised credentials.

Process Creation Events (**4688**) analysis focuses on monitoring the creation of new processes on the system, which is crucial for identifying malware execution or unauthorized activities. Malicious actors often execute new processes to establish their presence or propagate within the system. Anomalies in process creation, such as processes starting from unusual locations or known bad paths, can signal the presence of malware or unauthorized software.

Object Access Events analysis (**4663**) helps in detecting potential data breaches or insider threats by tracking access to sensitive files and folders. This analysis reviews the event logs for entries that indicate modifications, deletions, or unauthorized viewing of critical data. Sudden changes in access patterns, such as a non-privileged user accessing confidential files, can be early warnings of data exfiltration or misuse.

Scheduled tasks are frequently used by attackers for persistence, as they allow malicious software to be executed automatically at specified times or under certain conditions. Monitoring events related to the creation, modification, or deletion of scheduled tasks is crucial for identifying unauthorized activities that could indicate system compromise or tampering. Unauthorized changes, such as new tasks being registered or existing tasks being altered without proper authorization, are red flags. A key Event ID to watch is **4698**, which logs the creation of a scheduled task and helps detect potential unauthorized persistence mechanisms.

These fundamentals of host-based analysis provide a foundational understanding of the activities occurring within a single system. However, to fully grasp the scope of an incident, it's often necessary to also consider network-based indicators. Analyzing network interactions and traffic is essential for piecing together a comprehensive view of how threats propagate and affect the broader network environment.

NETWORK-BASED ANALYSIS

Network-based analysis is another fundamental aspect of cybersecurity that focuses on examining and monitoring network traffic to identify, understand, and mitigate security incidents across an organization’s network infrastructure. The primary purpose of network-based analysis is to detect and respond to anomalies in network behavior, investigate potential compromises, and safeguard the integrity and security of network communications.

This section will cover essential techniques and tools used for network-based analysis, including IP address analysis, port scanning,

protocol understanding, and distinguishing between normal and abnormal traffic.

IP ADDRESSES

An IP (Internet Protocol) address is a unique identifier assigned to each device connected to a network that uses the Internet Protocol for communication. There are two main versions of IP addresses: IPv4 and IPv6.

- **IPv4**: Consists of four sets of numbers separated by dots (e.g., 192.168.1.1).
- **IPv6**: Uses a longer format with eight groups of hexadecimal numbers separated by colons (e.g., 2001:0db8:85a3:0000:0000:8a2e:0370:7334).

IP addresses can be assigned and managed in several ways within a network:

- **Static IP Assignment**: A fixed IP address is manually assigned to a device by an administrator. This is often used for servers, network printers, or other devices that need a consistent address.
- **Dynamic IP Assignment**: IP addresses are automatically assigned to devices using DHCP (Dynamic Host Configuration Protocol). This is common in-home networks and larger organizations, as it simplifies the management of IP addresses.

Many organizations operate behind Virtual Private Networks (VPNs) and use static IPs provided by their VPN service. This approach allows organizations to mask their internal IP addresses and present a single IP address to the outside world, enhancing privacy and security. Static IPs through a VPN can provide consistent access for remote users and facilitate secure communications by ensuring that only known, trusted IP addresses are used for sensitive operations.

SUBNET

A subnet, or subnetwork, is a segmented portion of a larger IP network. Subnetting divides a network into smaller, more manageable segments, which enhances performance and security by reducing broadcast traffic and improving network efficiency. Each subnet is identified by a unique subnet mask, defining the size and range of IP addresses within that subnet.

For example, a subnet mask of 255.255.255.0 (or /24 in CIDR notation) indicates that the first 24 bits are the network portion, and the remaining 8 bits are for hosts.

IP addresses and subnets are critical for analysts in network security and operations. It enables analysts to trace the origin and destination of network activity during security incidents. By identifying the IPs involved, analysts can pinpoint potential sources of attacks, track the spread of malware or unauthorized access attempts, and correlate activity across systems. This information is essential for reconstructing the timeline of events, understanding the scope of a breach, and implementing effective containment and mitigation strategies. Understanding subnetting helps analysts identify network segments that may be affected, assess the impact, and enforce targeted security measures to prevent further exploitation.

PORTS, PROTOCOLS, AND SERVICES

Ports are endpoints used by computers for communication over a network. They enable different applications and services to interact with each other by providing a specific channel through which data is transmitted. Ports facilitate the transmission and reception of data between computers and devices. Each port is associated with a specific protocol or service, enabling

computers to distinguish between different types of network traffic.

Protocols are standardized sets of rules that determine how data is transmitted between different devices over a network. They define the procedures for formatting, sending, and receiving data across network connections, ensuring that devices—regardless of their underlying architectures—can communicate effectively. Protocols operate at various layers of the OSI (Open Systems Interconnection) model, each layer addressing specific aspects of network communications.

The foundational protocols for the internet and major focus in network-based analysis are TCP and IP (Transmission Control Protocol &Internet Protocol). TCP ensures data is sent and received accurately, while IP handles the addressing and routing of packets to their destination as explained in the previous section. DNS is another important protocol used to translate human-readable domain names into IP addresses. This enables users to access websites using easy-to-remember names instead of numerical addresses (i.e google.com IP is 8.8.8.8).

Services, in a network context, refer to specific applications or processes that utilize network protocols to perform useful functions or provide functionalities to user applications. Services are often associated with specific protocols at the application layer of the OSI model, and they enable diverse network functionalities such as web browsing, file transfer, email communication, and more.

Here are a few common examples of Protocols/Services and the ports that are frequently used:

HTTP and HTTPS (Web Browsing)

- **Protocol:** HTTP (Hypertext Transfer Protocol) and HTTPS (HTTP Secure)
- **Port:** Typically, HTTP uses port 80, and HTTPS uses port 443.
- **Service:** Web browsing

  When you type a website URL into your browser, the browser uses HTTP or HTTPS to request the webpage from a server. HTTPS includes encryption with SSL/TLS to secure the communication. The server listens on ports 80 (HTTP) or 443 (HTTPS) to respond to these requests, delivering web pages back to the browser.

FTP (File Transfer)

- **Protocol:** FTP (File Transfer Protocol)
- **Port:** Commonly uses ports 20 and 21; port 21 for control (commands) and port 20 for data transfer.
- **Service:** Transferring files between computers

  FTP allows users to upload and download files from a server. A client connects to the server using port 21 to send commands, and the actual file data transfers through port 20.

SSH (Secure Remote Access)

- **Protocol:** SSH (Secure Shell)
- **Port:** Typically uses port 22.
- **Service:** Secure remote administration

  SSH is used for securely logging into remote machines, allowing administrators to manage systems via a secure channel. All data, including login credentials, is encrypted to prevent eavesdropping.
  SSH is not native to Windows Operating Systems and can often be an indication of suspicious activity.

RDP (Remote Desktop Protocol)

- **Protocol:** RDP (Remote Desktop Protocol)
- **Port:** Generally uses port 3389.
- **Service:** Remote desktop access

  RDP lets users connect to a remote computer and use the Windows graphical interface over a network connection. This service is crucial for accessing a workstation or server from a remote location.

DNS (Domain Name System)

- **Protocol:** DNS
- **Port:** Uses port 53.
- **Service:** Resolving domain names
- DNS translates human-readable domain names (like google.com) into IP addresses (like 8.8.8.8). When you enter a URL in your browser, your computer uses DNS to find the corresponding IP address of the server you're trying to reach.

SMB (Server Message Block)

- **Protocol:** SMB (Server Message Block)
- **Port:** Typically uses TCP ports 139 and 445.
- **Service:** Windows file sharing and inter-process communication

  SMB is used for providing shared access to files, printers, and serial ports among network devices. It's also used in Windows for network file system operations.

ENDPOINT DIVERSITY

In any organization, the network includes a diverse array of endpoints, each serving specific functions crucial for the overall IT infrastructure. Understanding these different types of endpoints and the anticipated traffic they receive are important when investing incidents, let's look at a few.

Workstations are the primary interfaces for employees, running daily operations on operating systems like Windows, macOS, or Linux. These endpoints are frequent targets due to their direct internet access and data sensitivity, requiring vigilance in monitoring for malware and phishing attacks.

Data Servers manage and store critical business data, making them key targets for breaches. Protecting these involves securing data integrity and guarding against unauthorized access.

File Share Servers enable collaborative file access and editing within the organization. They pose risks related to malware propagation and insider threats, emphasizing the importance of managing file permissions securely.

Application Servers host crucial applications like web services and business software, processing sensitive transactions. Security analysts must focus on safeguarding these servers from application-level exploits and ensuring robust security practices. Often times, the applications that are hosted are monitored by a separate Application Security team; these members will be responsible for Penetration testing or directing bug bounty programs with 3rd party penetration testers (pentesters).

Domain Controllers manage user authentication and enforce security policies across Windows networks. Given their role in network access control, protecting domain controllers is vital for preventing unauthorized access and network breaches. One of the key services provided by domain controllers is **LDAP (Lightweight Directory Access Protocol)**, which is used for accessing and managing directory information. In Windows environments, LDAP facilitates the querying and modification of user and computer objects within Active Directory. It supports a wide range of operations, from user authentication to directory searches.

Printers and Multifunction Devices, often overlooked in security planning, can store sensitive documents and connect directly to the network. Ensuring these devices are secure is crucial to prevent data leakage and exploit vulnerabilities. (ex. PrintNightmare)

Mobile Devices, including smartphones and tablets, are either company-issued or part of a Bring Your Own Device (BYOD) policy. These devices are at high risk of theft, loss, and cyber attacks, necessitating comprehensive security protocols to safeguard mobile access to corporate data.

IoT Devices represent a growing segment in networks and include devices like sensors and automated systems. Their often minimally built-in security makes them susceptible to attacks, which can serve as entry points into the network.

For security analysts in incident response, understanding the roles and vulnerabilities of each endpoint type allows for more effective monitoring, threat detection, and rapid response to incidents. This knowledge is critical for developing targeted security measures and response strategies that address the specific risks associated with each type of endpoint.

TRAFFIC ANALYSIS

Traffic analysis involves examining network traffic to understand the flow of data across the network, identify patterns, and detect potential security threats. This process is crucial for maintaining network security, performance, and integrity by identifying abnormal behaviors that may indicate malicious activities.

OBSERVING PATTERNS

Patterns can be categorized into normal and abnormal traffic, with the latter often signaling potential security issues. **Baseline traffic** represents the typical, day-to-day network activity that occurs under normal operating conditions. This includes regular user activities, scheduled backups, routine system updates, and normal application usage. To establish what normal network traffic looks like, analysts must monitor the network over a period of time, noting the volume of traffic, common protocols used, and the typical endpoints involved. Key components of baseline traffic can include, typical source and destination IP addresses, common ports and services used, and normal traffic flow patterns during different times of the day and week. **Anomalies** are deviations from the established baseline traffic patterns.

Unusual traffic spikes, unexpected protocol usage, communication with unknown or blacklisted IP addresses, and abnormal port activity are all examples of deviations an analyst needs to be aware of when monitoring traffic. For instance, a sudden increase in outbound traffic might suggest data exfiltration, while unusual inbound connections could indicate a scanning attempt or an active intrusion. Unusual user activities, such as access from unexpected locations or at unusual times, can signal compromised accounts or insider threats.

IDENTIFYING MALICIOUS ACTIVITIES

Putting it all together, network traffic analysis enables analysts to identify various stages of malicious activity by detecting distinctive patterns and behaviors within the network. Below are a few key stages that can be identified through thorough analysis.

Reconnaissance

Reconnaissance is the early stage of an attack where adversaries gather information about the network to identify potential vulnerabilities and plan their intrusion strategy. During this phase, attackers may perform various activities to map the network and gather critical information.

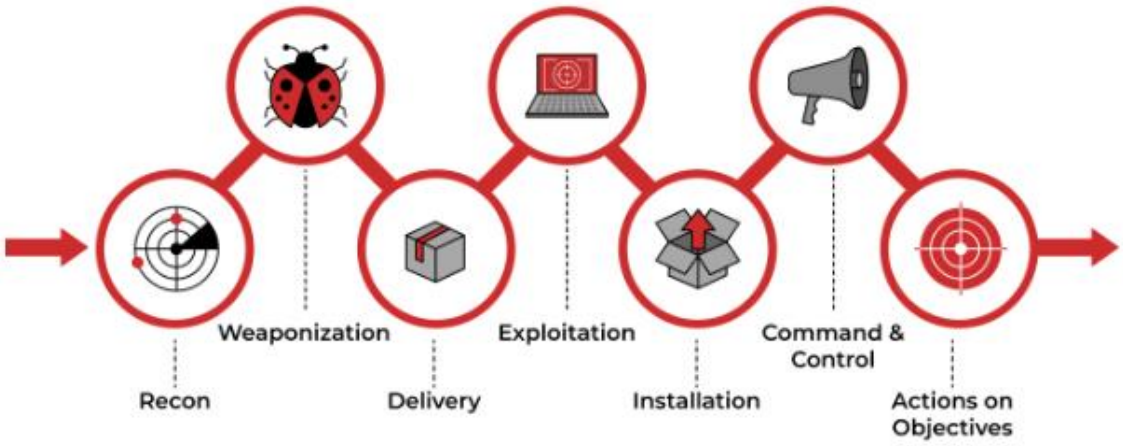
Indicators of reconnaissance include unusual port scans, where attackers probe different ports to find open ones that can be exploited. An increased volume of DNS queries, particularly those targeting internal domain names or attempting to resolve numerous external domains rapidly, can also signal reconnaissance. Network mapping efforts, often conducted using tools like Nmap, can generate abnormal patterns of network traffic, such as a high number of ICMP requests or responses (ping sweeps).

service accounts used inappropriately) can be signs of lateral movement.

Command and Control (C2)

Command and Control (C2) channels are used by attackers to communicate with compromised devices within the network. Through these channels, attackers can issue commands, control malware, and coordinate further actions.

Indicators of C2 activity include regular intervals of network traffic to known malicious IP addresses, which can suggest a



Lateral Movement

Lateral movement is when attackers move within the network to find and access key assets, such as confidential data or critical systems. This stage often involves compromising additional systems and escalating privileges to gain broader access.

Indicators of lateral movement include unexpected internal IP communications, where devices that typically do not communicate with each other start exchanging data. The use of uncommon protocols or services, such as remote desktop protocol (RDP) sessions initiated from unusual locations, can also be indicative of lateral movement. Additionally, repeated authentication attempts or the use of legitimate but unusual accounts (e.g.,

compromised device checking in with a remote server. The use of encrypted communications in unexpected contexts, such as encrypted traffic on ports typically used for unencrypted protocols, might also signal C2 activity. Additionally, domain generation algorithm (DGA) patterns, where compromised devices attempt to resolve numerous seemingly random domain names, can indicate efforts to establish or maintain C2 connections.

Exfiltration

Exfiltration refers to the unauthorized transfer of data out of the network. Attackers, having gained access to sensitive information, will attempt to move this data to an external location under their control.

Indicators of exfiltration include abnormal data transfer volumes, especially when directed to unknown or suspicious external IP addresses. Sudden spikes in outbound traffic or large data transfers during off-peak hours can also be red flags. Additionally, the use of uncommon protocols for data transfer or encryption methods that are not typically used within the organization might suggest exfiltration activities.

Network-based analysis encompasses a breadth of specialized areas that warrant deeper exploration. While some, like Email Security, will be within the scope of our training, others such as web application security, penetration testing, and cloud security, typically fall outside the direct purview of DFIR but will receive brief highlights throughout our sessions.

Next, we'll leverage the foundational skills in Host and Network-based analysis we've developed to discuss the common tools widely used across the industry. These tools are crucial for detecting and investigating security incidents, empowering analysts with the capabilities needed to effectively safeguard organizational assets.

EDR & AV

Endpoint Detection and Response (EDR) and Antivirus (AV) solutions are fundamental components of modern cybersecurity strategies; across the industry, DFIR analysts use various tools like these to protect endpoints/servers from a wide range of cyber threats.

ANTIVIRUS SOLUTIONS

AV solutions are essential tools for protecting endpoints from malware and other malicious threats. They employ a variety of detection techniques to identify and mitigate threats, with the primary methods being signature-based detection and heuristic and behavior-based detection.

Signature-based detection is the traditional and most widely used method in antivirus solutions. This technique relies on a database of known malware signatures—unique patterns or characteristics of malicious code. When an AV solution scans a file, application, or process, it compares the code against this database to look for matches. If a match is found, the antivirus flags the file as malicious and takes appropriate action, such as quarantining or deleting the file. The primary advantage of signature-based detection is its high accuracy for known threats, making it highly effective at identifying and removing malware that has been previously analyzed and cataloged.

However, signature-based detection has its limitations. It is ineffective against new threats, as it can only detect malware that is already in the signature database. New, unknown malware (zero-day threats) can evade detection until their signatures are added to the database. Moreover, the signature database must be constantly updated with new malware signatures to remain effective, which can be challenging for organizations to manage.

This need bred the development of heuristic and behavior-based detection. These techniques analyze the behavior and characteristics of files and processes to determine if they are malicious.

Heuristic detection involves analyzing the code of a file to look for suspicious structures or commands that are commonly associated with malware. It uses predefined rules or algorithms to evaluate the potential risk and can involve both static analysis (examining the code without executing it) and dynamic analysis (observing the behavior of code when executed in a controlled environment).

Behavior-based detection, on the other hand, monitors the actions of applications and processes in real-time. It looks for behaviors indicative of malicious activity, such as attempts to modify system files, create unauthorized network connections, or execute in unusual ways. This method often involves creating a baseline of normal behavior for the system and detecting deviations from this baseline that may indicate a threat. The key advantage of heuristic and behavior-based detection is their ability to detect unknown threats, providing protection against zero-day threats. These methods offer a more proactive defense by detecting and responding to suspicious activities before a known signature is available.

Despite their benefits, heuristic and behavior-based detection methods have some drawbacks. Because they rely on detecting suspicious behavior, they can sometimes flag legitimate activities as malicious, leading to **false positives**.

Some actions an analyst can expect / perform using Antivirus software:

Quarantine: Isolating suspicious files or programs to prevent them from executing or causing harm to the system.

Kill: Terminating processes associated with identified malware to halt their activity immediately.

Hash Blocking: Blocking known malicious files based on their cryptographic hash values to prevent them from being executed.

Containment: Isolating an entire workstation / server from the network to contain a potential infection and prevent further spread.

Policy Enforcement: Enforcing security policies that dictate acceptable behaviors and configurations such as, blocking Mouse Jigglers by Product and Vendor ID.

Custom Signatures: Creating specific signatures or policies within the AV solution to customize detection criteria and response actions based on organizational needs and threat intelligence.

Modern AV solutions are beginning to adopt features traditionally associated with EDR solutions (Extended Detection and Response – XDR). These include capabilities such as continuous monitoring, endpoint direct connection, and real-time threat hunting.

EDR

EDR represents an evolution in endpoint security. EDR systems provide advanced capabilities to detect, investigate, and respond to threats that evade traditional antivirus measures. EDR tools continuously monitor endpoint activities, capturing detailed information about system behaviors and network interactions. This real-time monitoring enables the identification of suspicious activities and potential breaches through behavioral analysis and anomaly detection.

DETECTION

A key aspect of EDR solutions is the ability to create detections based on defined threat indicators. Creating effective detections in involves defining rules and conditions that trigger alerts when suspicious activities are detected on endpoints. The process begins with understanding the specific environment and establishing a baseline of normal behavior, which includes typical applications, user activities, network connections, and processes. This knowledge is essential for accurately identifying deviations that may signal potential threats. After we baseline the normal, the next step is identifying Indicators of Compromise (IOCs), such as suspicious file hashes, IP addresses, domain names, URLs, registry keys, and unusual process behaviors. These IOCs help in crafting precise detection rules.

Let's say, for example, our EDR tool has an alert that searches for non-standard

applications creating network connections and it is alerting on the process `calc.exe` (Windows Calculator application) attempting to initiate an SSH connection.

As an analyst, the alarms should be sounding off in your head as this behavior is highly unusual and could indicate a process injection attack, where an attacker uses a legitimate process to execute malicious actions, bypassing traditional security measures. Now you can readily identify a threat in the environment and start stepping through the response plan. You may ask yourself:

“Should I blacklist (block) this hash or disconnect the workstation from the network? “

“What is the destination IP address this process is communicating with?”

“Does the hash for this process match for the known native calculator app?”

“Who or what initiated this process?”
“Are there any suspicious processes in the ancestry that led to this?”

EDR can help us answer these questions as well as take quick action. One of the immediate steps is to isolate the infected host from the network. Network isolation prevents the potential spread of the threat and limits the attacker's ability to communicate with the compromised device, thereby containing the threat. This action allows analysts to perform a more detailed investigation without the risk of further contamination or data exfiltration.

EDR solutions also offer capabilities to collect artifacts and perform file fetch operations. Analysts can remotely retrieve files, logs, and memory dumps from the affected system for forensic analysis. This helps in understanding the nature and extent of the compromise, identifying any malicious payloads, and gathering evidence for further investigation. The collected artifacts can be analyzed to determine how the attack occurred and what steps need to be taken to remediate the system fully.

THREAT HUNTING AND DEEP VISIBILITY

EDR solutions facilitate threat hunting, which involves proactively searching for indicators of compromise (IOCs) and anomalous activities across the network. Threat hunting can help identify other potentially compromised systems, uncover hidden threats, and provide insights into the attacker's tactics, techniques, and procedures (TTPs). By leveraging EDR's threat hunting capabilities, analysts can examine various categories of data collected from endpoints to identify potential threats. There is variance across EDR solutions, but here are a few more common categories:

PROCESS EVENTS

EDR solutions provide detailed logs of process events, allowing analysts to investigate the execution of applications and scripts. Key aspects to examine include:

- **Process Creation**: Identifying unusual or unauthorized processes being initiated, such as calc.exe attempting an SSH connection.
- **Parent-Child Relationships**: Analyzing the hierarchy of processes to detect abnormal parent-child relationships, such as a legitimate application spawning a known malicious process.
- **Execution Paths**: Reviewing the file paths from which processes are executed to spot processes running from suspicious or non-standard locations.
- **Command-line Arguments**: Inspecting the arguments used to launch processes, which can reveal malicious intent even if the process itself appears benign.

File Events

Tracking file events provides insights into file creation, modification, and deletion activities. Operations to monitor include:

- **File Access**: Monitoring read and write operations on sensitive files and directories to detect unauthorized access.
- **File Creation/Modification**: Detecting the creation or alteration of executables, scripts, or system files, which may indicate malware installation or tampering.
- **File Deletion**: Observing deletion patterns that may be used to cover tracks or remove evidence of malicious activities.

DNS Events

DNS query logs are valuable for identifying malicious network communications. Key areas to investigate include:

- **Domain Lookups**: Identifying queries to known malicious domains or newly registered domains often used in phishing and C2 (Command and Control) communications.
- **Frequency of Requests**: Spotting unusual patterns in DNS queries, such as high-frequency requests to the same domain, which could indicate data exfiltration or botnet activity.
- **NXDOMAIN Responses**: Monitoring failed DNS lookups, which can suggest the use of domain generation algorithms (DGAs) by malware.

Registry Events

Registry events are useful for detecting changes that can indicate persistence mechanisms and configuration changes. Important operations to consider include:

- **Registry Key Creation/Modification**: Identifying changes to registry keys and values, especially in areas commonly used for persistence, such as the Run and RunOnce keys.
- **Registry Key Deletion**: Monitoring deletions that may be used to disable security tools or remove evidence.
- **Sensitive Areas**: Focusing on changes within critical registry hives like HKEY_LOCAL_MACHINE and HKEY_CURRENT_USER that control startup processes and system configurations.

Driver Events

Driver events involve the loading and unloading of system drivers, which are crucial for the proper functioning of hardware and low-level system operations.

Driver Installation: Detecting the installation of unsigned or suspicious drivers that could indicate the presence of rootkits or other low-level malware.

For all categories, timestamps are critical for establishing timelines and correlating events. Analysts should review the timing of events to

detect patterns, such as multiple suspicious activities occurring within a short period.

Try to always cross-reference timestamps across different event categories (e.g., process creation and file modification) to build a comprehensive picture of an attack.

FALSE POSITIVES

A common challenge in cybersecurity that is another very important part of the job is managing False Positives. False positives occur when security solutions incorrectly identify benign activity as malicious. These erroneous alerts can stem from legitimate processes, applications, or user actions that mimic patterns typically associated with threats. High rates of false positives can overwhelm security teams, leading to alert fatigue and potentially causing real threats to be overlooked. Conversely, reducing false positives improves the signal-to-noise ratio, allowing analysts to focus on genuine threats and respond more quickly and effectively.

1. Review the context in which the alert was generated, including the behavior of the application or process, the user's actions, and the system's state at the time.
2. Duplicating efforts is never effective use of time, compare the current alert with historical data to determine if similar activity has previously been flagged as benign.
3. Tune Alerts: this is adjusting the detection rules and thresholds to minimize false positives without missing actual threats.
4. Suppress False Positives: Identify patterns in false positives and adjust detection logic to account for these common benign activities.

As new analysts, it may be challenging to understand what's normal vs. abnormal. It's never a bad idea to have a more senior level analyst review the criteria you wish to suppress before making the change; you're contributing to the improvement detection fidelity and enabling teams to respond more effectively.

REFERENCES

[1] **SANS Institute. (2023).** *FOR508: Advanced Incident Response, Threat Hunting, and Digital Forensics [Course Material].* SANS Institute.

[2] https://www.xplg.com/windows-server-security-events-list/

[3] https://attack.mitre.org/

[4] https://medium.com/@kumarishefu.4507/try-hack-me-cyber-kill-chain-write-up-d076d323fdc8

[5] https://www.trellix.com/security-awareness/endpoint/what-is-endpoint-detection-and-response/