# Incident Response and Log Analysis

Dr Christopher D. McDermott

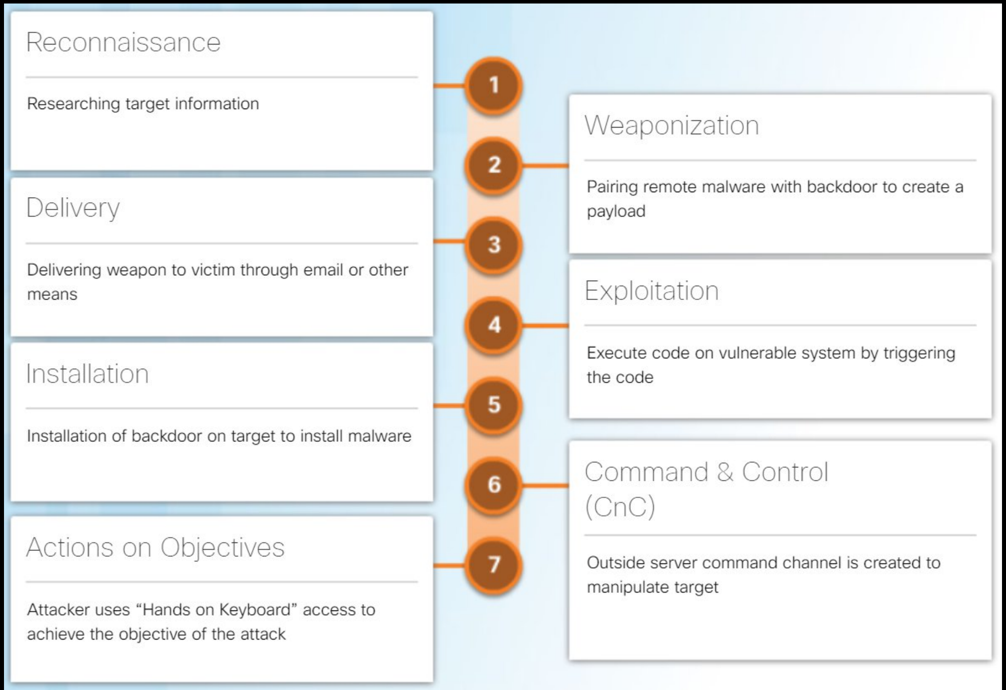
# Outline

1. Introduction
2. Incident Response Frameworks
   - Cyber Kill Chain
   - The Diamond Model
   - Other prominent frameworks
3. NIST & NCSC Guidance
4. Log Analysis
   - Security Onion

# Cyber Kill Chain

# Steps of the Cyber Kill Chain®

- Developed by **Lockheed Martin** to identify and prevent cyber intrusions.
    - The steps of the Cyber Kill Chain help analysts understand the techniques, tools, and procedures of threat actors.
    - The threat actor gains more access to the target as they progress through the steps.
    - The goal is to stop them as early as possible to lessen the damage done.



1 Reconnaissance

Researching target information

2 Weaponization

Pairing remote malware with backdoor to create a payload

3 Delivery

Delivering weapon to victim through email or other means

4 Exploitation

Execute code on vulnerable system by triggering the code

5 Installation

Installation of backdoor on target to install malware

6 Command & Control (CnC)

Outside server command channel is created to manipulate target

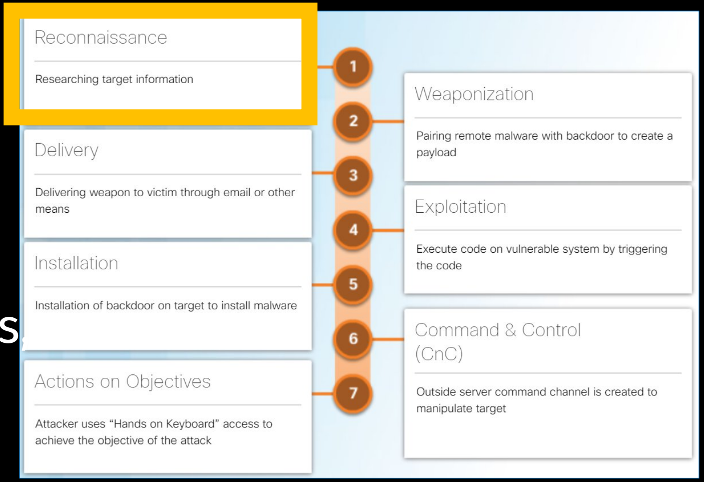
7 Actions on Objectives

Attacker uses “Hands on Keyboard” access to achieve the objective of the attack

The Cyber Kill Chain

# Reconnaissance

- **Reconnaissance** is when the threat actor performs research, gathers intelligence, and selects targets.
- Organizations may provide information on websites public-facing network devices, in news articles, conference proceedings, and social media outlets.



Reconnaissance

Researching target information

Weaponization

Pairing remote malware with backdoor to create a payload

Delivery

Delivering weapon to victim through email or other means

Exploitation

Execute code on vulnerable system by triggering the code

Installation

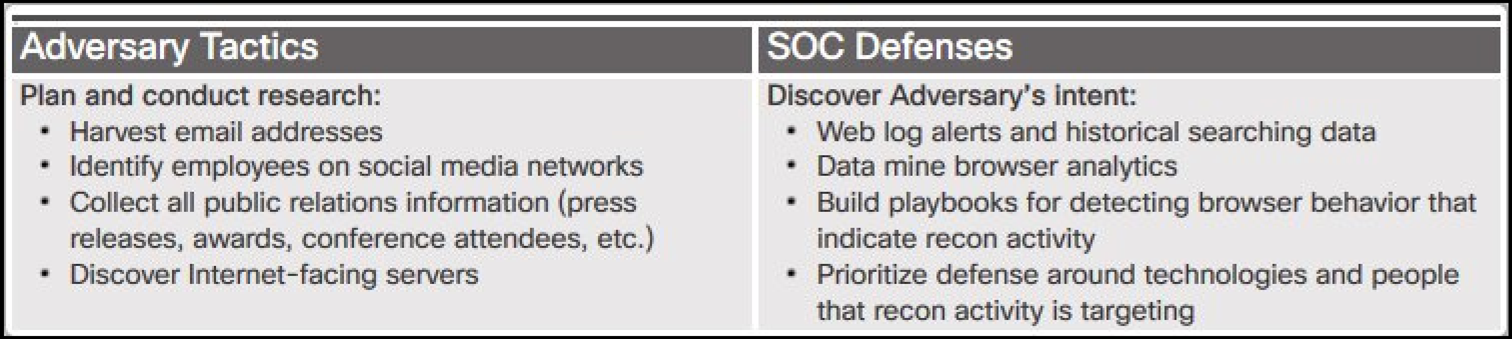
Installation of backdoor on target to install malware

Command & Control (CnC)

Outside server command channel is created to manipulate target

Actions on Objectives

Attacker uses "Hands on Keyboard" access to achieve the objective of the attack

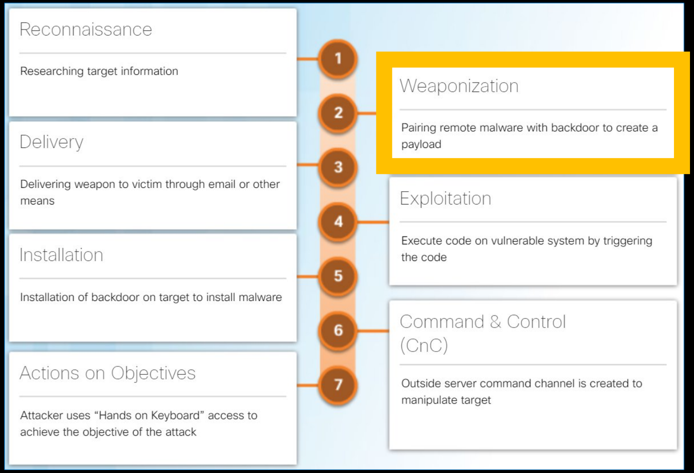


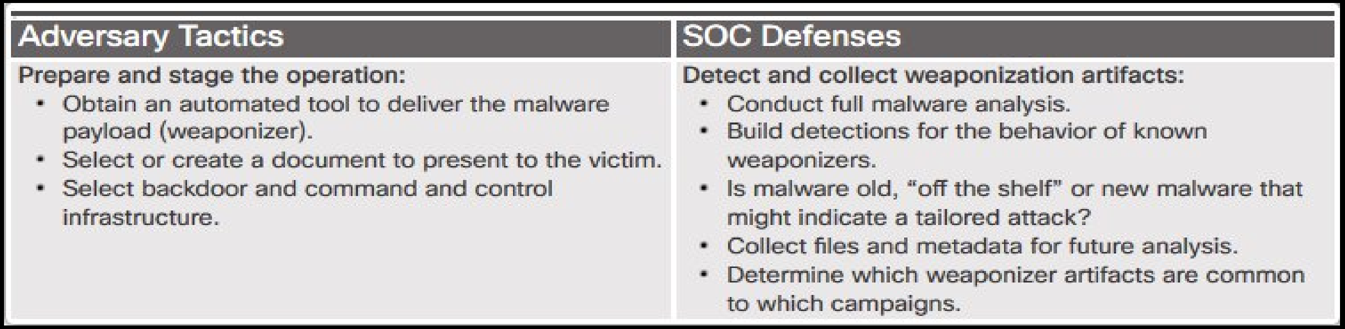
| Adversary Tactics | SOC Defenses |
|---|---|
| Plan and conduct research:<br>• Harvest email addresses<br>• Identify employees on social media networks<br>• Collect all public relations information (press releases, awards, conference attendees, etc.)<br>• Discover Internet-facing servers | Discover Adversary's intent:<br>• Web log alerts and historical searching data<br>• Data mine browser analytics<br>• Build playbooks for detecting browser behavior that indicate recon activity<br>• Prioritize defense around technologies and people that recon activity is targeting |

The Cyber Kill Chain

# Weaponization

- **Weaponization** uses the vulnerability information gathered in the reconnaissance step to identify and develop a weapon against specific targeted systems in the organization.


Reconnaissance
Researching target information
Delivery
Delivering weapon to victim through email or other means
Installation
Installation of backdoor on target to install malware
Actions on Objectives
Attacker uses "Hands on Keyboard" access to achieve the objective of the attack
1
2
3
4
5
6
7
Weaponization
Pairing remote malware with backdoor to create a payload
Exploitation
Execute code on vulnerable system by triggering the code
Command & Control (CnC)
Outside server command channel is created to manipulate target

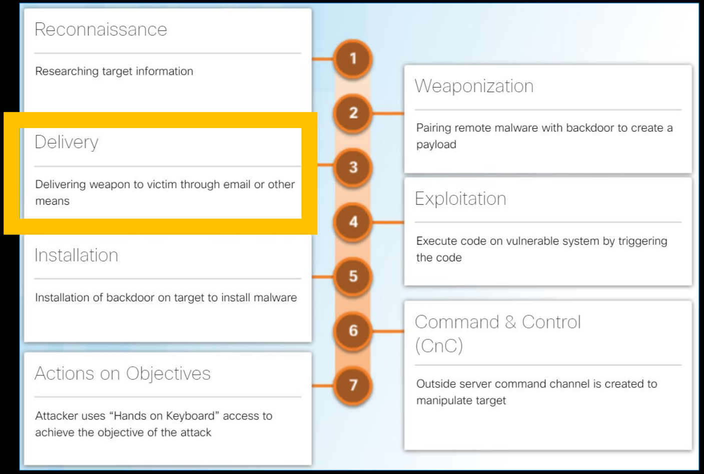


| Adversary Tactics | SOC Defenses |
|---|---|
| Prepare and stage the operation:<br>• Obtain an automated tool to deliver the malware payload (weaponizer).<br>• Select or create a document to present to the victim.<br>• Select backdoor and command and control infrastructure. | Detect and collect weaponization artifacts:<br>• Conduct full malware analysis.<br>• Build detections for the behavior of known weaponizers.<br>• Is malware old, "off the shelf" or new malware that might indicate a tailored attack?<br>• Collect files and metadata for future analysis.<br>• Determine which weaponizer artifacts are common to which campaigns. |

The Cyber Kill Chain

# Delivery

- **Delivery** is when the threat actor delivers the developed weapon using either a website, a removable USB media, or an email attachment.



Reconnaissance

Researching target information

Weaponization

Pairing remote malware with backdoor to create a payload

Delivery

Delivering weapon to victim through email or other means

Exploitation

Execute code on vulnerable system by triggering the code

Installation

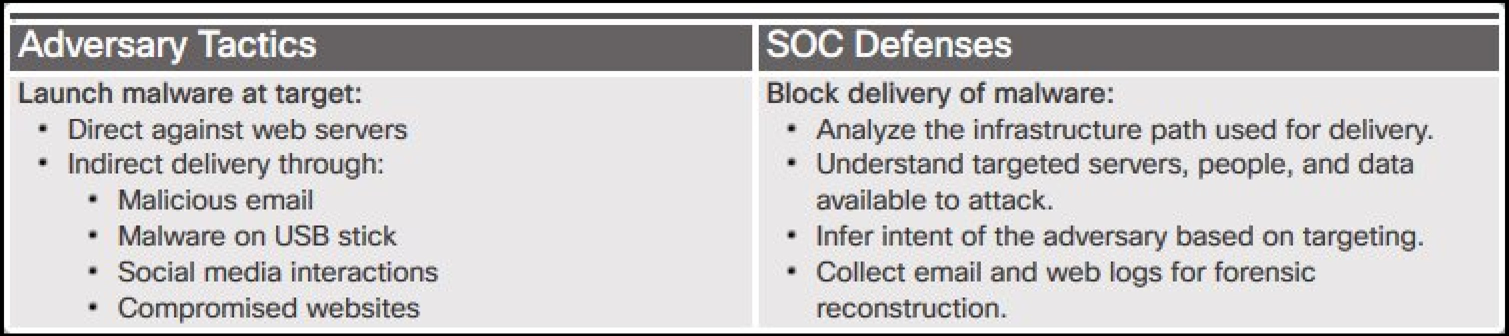
Installation of backdoor on target to install malware

Command & Control (CnC)

Outside server command channel is created to manipulate target

Actions on Objectives

Attacker uses "Hands on Keyboard" access to achieve the objective of the attack

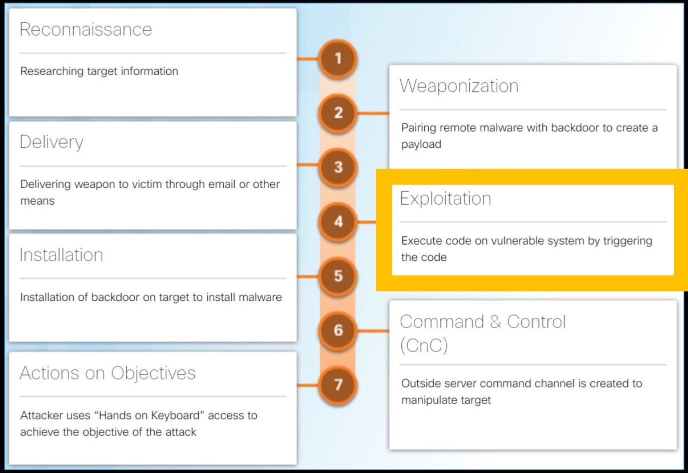


| Adversary Tactics | SOC Defenses |
|---|---|
| Launch malware at target:<br>• Direct against web servers<br>• Indirect delivery through:<br>  • Malicious email<br>  • Malware on USB stick<br>  • Social media interactions<br>  • Compromised websites | Block delivery of malware:<br>• Analyze the infrastructure path used for delivery.<br>• Understand targeted servers, people, and data available to attack.<br>• Infer intent of the adversary based on targeting.<br>• Collect email and web logs for forensic reconstruction. |

The Cyber Kill Chain

# Exploitation

- **Exploitation** is when the threat actor triggers the weapon and executes it to compromise the vulnerability and gain control of the target.



1 Reconnaissance: Researching target information

2 Weaponization: Pairing remote malware with backdoor to create a payload

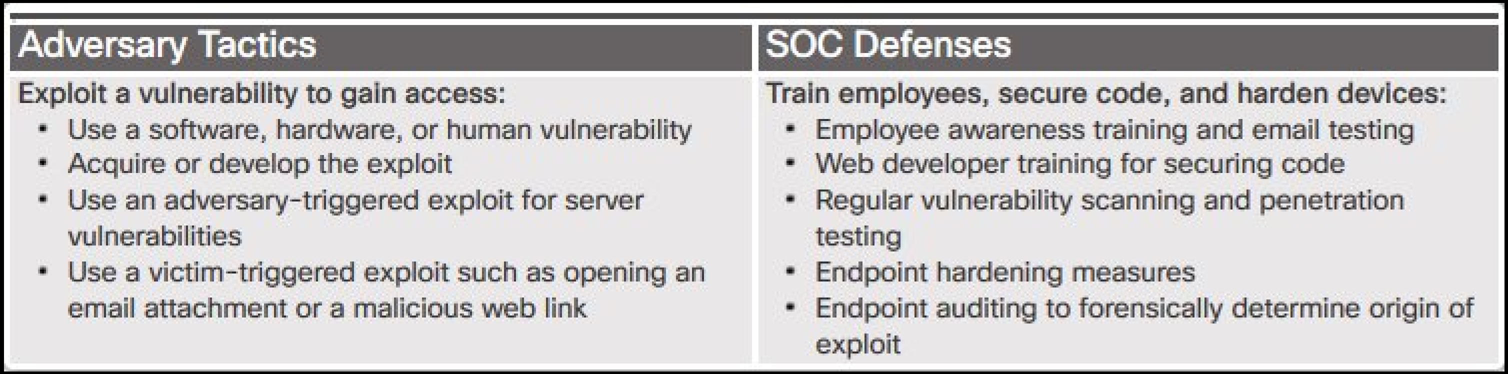
3 Delivery: Delivering weapon to victim through email or other means

4 Exploitation: Execute code on vulnerable system by triggering the code

5 Installation: Installation of backdoor on target to install malware

6 Command & Control (CnC): Outside server command channel is created to manipulate target

7 Actions on Objectives: Attacker uses "Hands on Keyboard" access to achieve the objective of the attack

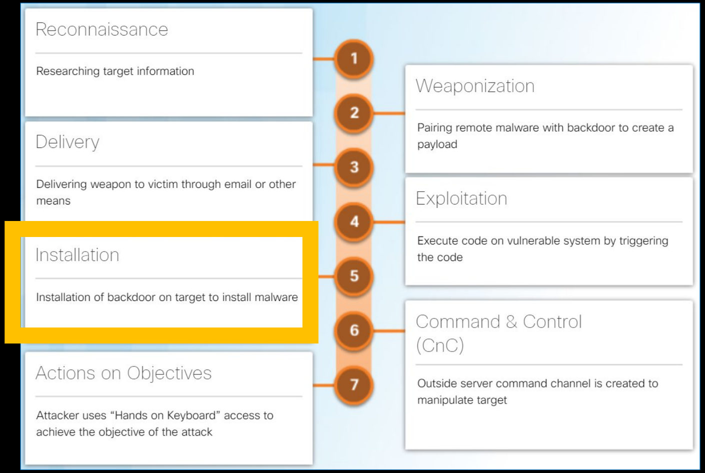


| Adversary Tactics | SOC Defenses |
|---|---|
| Exploit a vulnerability to gain access:<br>• Use a software, hardware, or human vulnerability<br>• Acquire or develop the exploit<br>• Use an adversary-triggered exploit for server vulnerabilities<br>• Use a victim-triggered exploit such as opening an email attachment or a malicious web link | Train employees, secure code, and harden devices:<br>• Employee awareness training and email testing<br>• Web developer training for securing code<br>• Regular vulnerability scanning and penetration testing<br>• Endpoint hardening measures<br>• Endpoint auditing to forensically determine origin of exploit |

The Cyber Kill Chain

# Installation

- **Installation** is when the threat actor establishes a back door into the system to allow for continued access to the target.



1 Reconnaissance

Researching target information

2 Weaponization

Pairing remote malware with backdoor to create a payload

3 Delivery

Delivering weapon to victim through email or other means

4 Exploitation

Execute code on vulnerable system by triggering the code

5 Installation

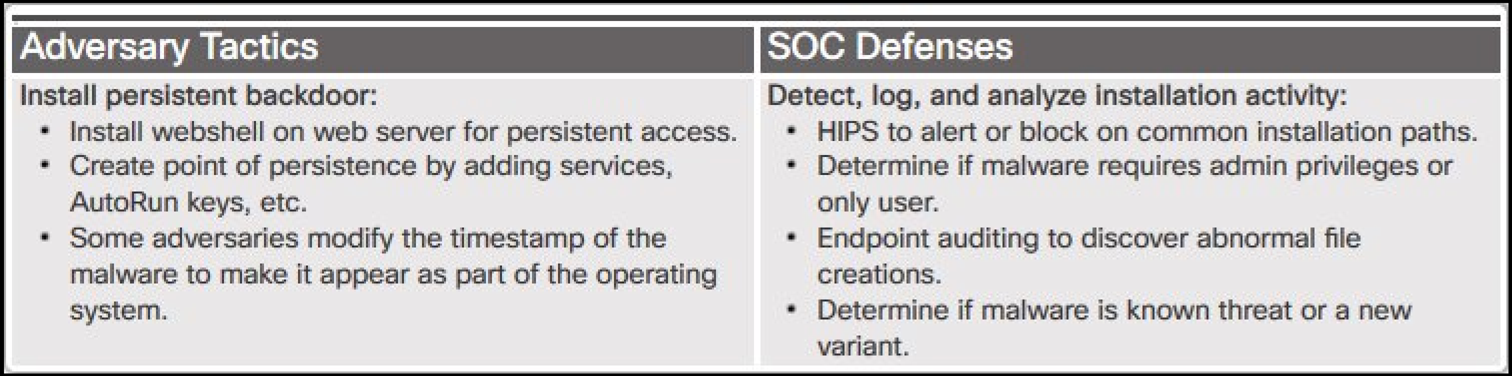
Installation of backdoor on target to install malware

6 Command & Control (CnC)

Outside server command channel is created to manipulate target

7 Actions on Objectives

Attacker uses “Hands on Keyboard” access to achieve the objective of the attack

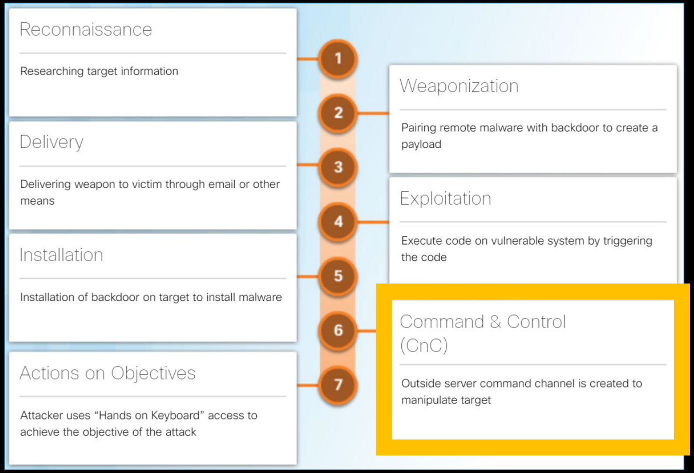


| Adversary Tactics | SOC Defenses |
|---|---|
| Install persistent backdoor:<br>• Install webshell on web server for persistent access.<br>• Create point of persistence by adding services, AutoRun keys, etc.<br>• Some adversaries modify the timestamp of the malware to make it appear as part of the operating system. | Detect, log, and analyze installation activity:<br>• HIPS to alert or block on common installation paths.<br>• Determine if malware requires admin privileges or only user.<br>• Endpoint auditing to discover abnormal file creations.<br>• Determine if malware is known threat or a new variant. |

The Cyber Kill Chain

# Command and Control

- **Command & Control** (CnC or C2) is when an outside server channel is used by the threat actor to manipulate a target by issuing commands to the software that they installed on the target.



1 Reconnaissance

Researching target information

2 Weaponization

Pairing remote malware with backdoor to create a payload

3 Delivery

Delivering weapon to victim through email or other means

4 Exploitation

Execute code on vulnerable system by triggering the code

5 Installation

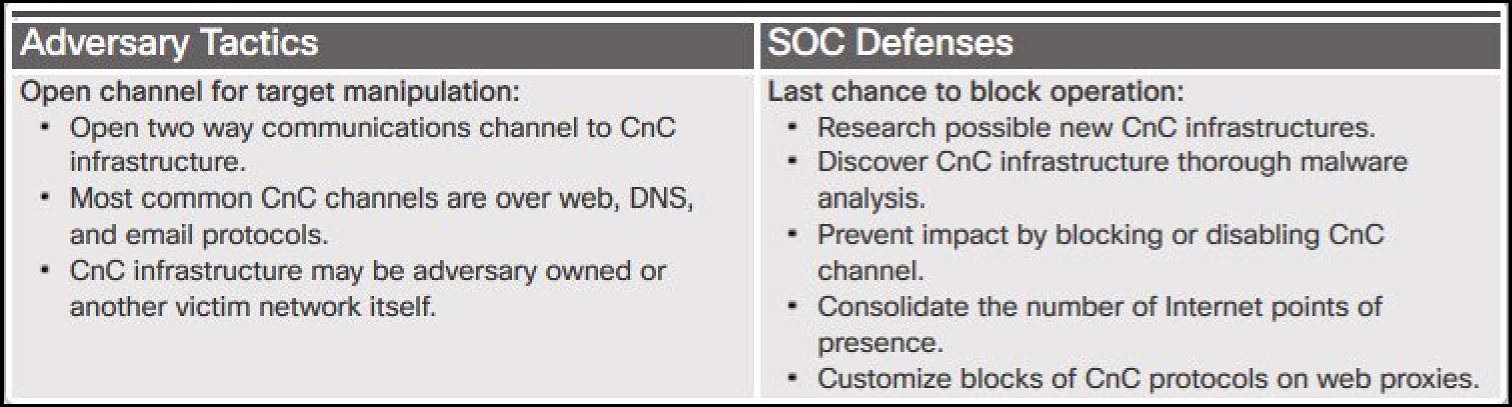
Installation of backdoor on target to install malware

6 Command & Control (CnC)

Outside server command channel is created to manipulate target

7 Actions on Objectives

Attacker uses "Hands on Keyboard" access to achieve the objective of the attack

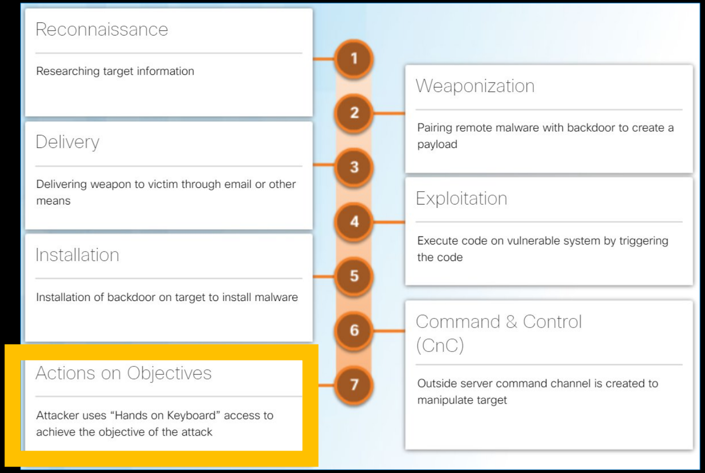


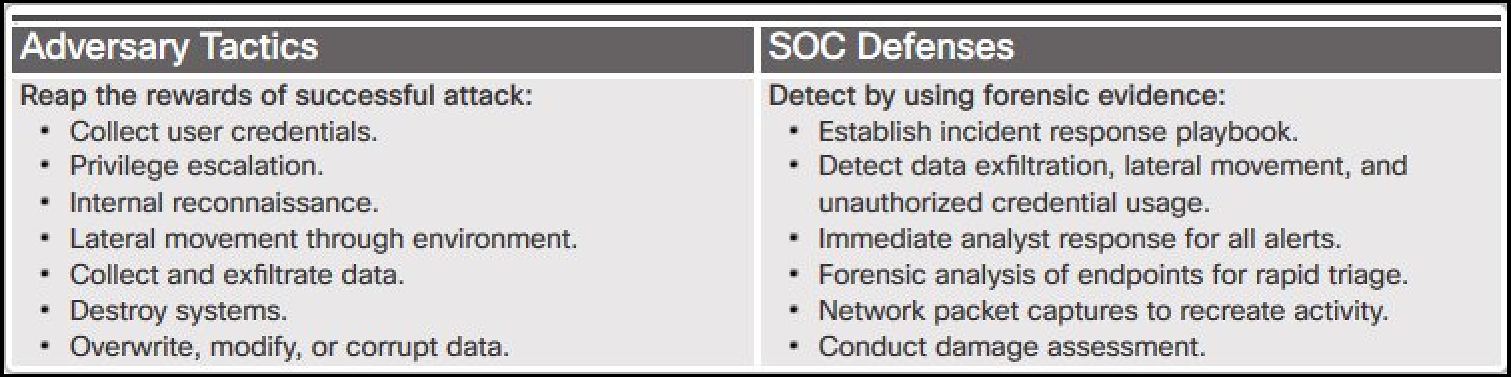
| Adversary Tactics | SOC Defenses |
|---|---|
| Open channel for target manipulation:<br>• Open two way communications channel to CnC infrastructure.<br>• Most common CnC channels are over web, DNS, and email protocols.<br>• CnC infrastructure may be adversary owned or another victim network itself. | Last chance to block operation:<br>• Research possible new CnC infrastructures.<br>• Discover CnC infrastructure thorough malware analysis.<br>• Prevent impact by blocking or disabling CnC channel.<br>• Consolidate the number of Internet points of presence.<br>• Customize blocks of CnC protocols on web proxies. |

The Cyber Kill Chain

# Actions on Objectives

- **Actions on Objectives** is the final step of the kill chain and is when the attacker achieves attack objective.
  - Can be used for data theft, performing a DDoS attack, or using the compromised network to create and send spam.
  - Threat actor is deeply rooted in the systems of the organization and may be extremely difficult to remove from the network.


1
Reconnaissance
Researching target information
2
Weaponization
Pairing remote malware with backdoor to create a payload
3
Delivery
Delivering weapon to victim through email or other means
4
Exploitation
Execute code on vulnerable system by triggering the code
5
Installation
Installation of backdoor on target to install malware
6
Command & Control (CnC)
Outside server command channel is created to manipulate target
7
Actions on Objectives
Attacker uses "Hands on Keyboard" access to achieve the objective of the attack



| Adversary Tactics | SOC Defenses |
|---|---|
| Reap the rewards of successful attack:<br>• Collect user credentials.<br>• Privilege escalation.<br>• Internal reconnaissance.<br>• Lateral movement through environment.<br>• Collect and exfiltrate data.<br>• Destroy systems.<br>• Overwrite, modify, or corrupt data. | Detect by using forensic evidence:<br>• Establish incident response playbook.<br>• Detect data exfiltration, lateral movement, and unauthorized credential usage.<br>• Immediate analyst response for all alerts.<br>• Forensic analysis of endpoints for rapid triage.<br>• Network packet captures to recreate activity.<br>• Conduct damage assessment. |

# The Cyber Kill Chain
# Defensive Measures Mapped to CKC

| Cyber Kill Chain Stage | Defensive Measures |
|---|---|
| **1. Reconnaissance** | OSINT reduction, firewalls, IDS, employee awareness |
| **2. Weaponization** | Threat intelligence, patching, sandbox analysis |
| **3. Delivery** | Email security, web filtering, USB controls |
| **4. Exploitation** | EDR, application whitelisting, exploit mitigations |

# The Cyber Kill Chain
## Defensive Measures Mapped to CKC

| Cyber Kill Chain Stage | Defensive Measures |
|---|---|
| 5. Installation | NGAV, FIM, registry monitoring, endpoint hardening |
| 6. Command & Control | Network segmentation, traffic filtering, honeypots |
| 7. Actions on Objectives | Data encryption, DLP, backups, incident response |

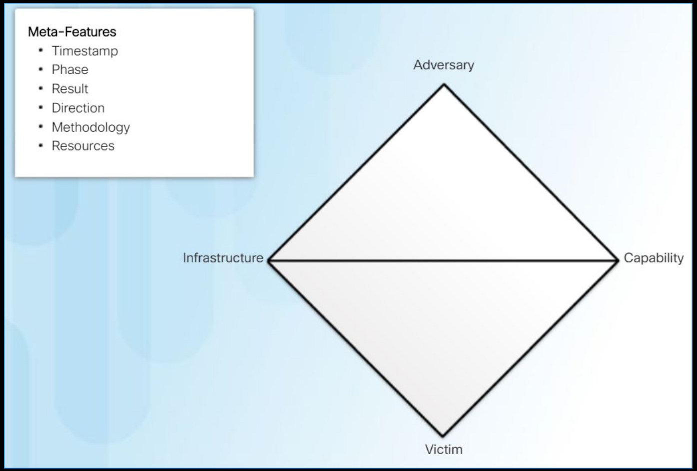
# Diamond Model

The Diamond Model of Intrusion

# Diamond Model Overview

- The Diamond Model identifies four parts involved in a security incident.

Meta-features expand the model to include important elements.



- **Adversary** – Parties responsible for the intrusion.
- **Capability** – Tool or technique used by the threat actor.
- **Infrastructure** – The network path(s) used by the threat actor  to establish and maintain command and control.
- **Victim** – The target of the attack. The victim could then used as part of the infrastructure to launch other attacks.

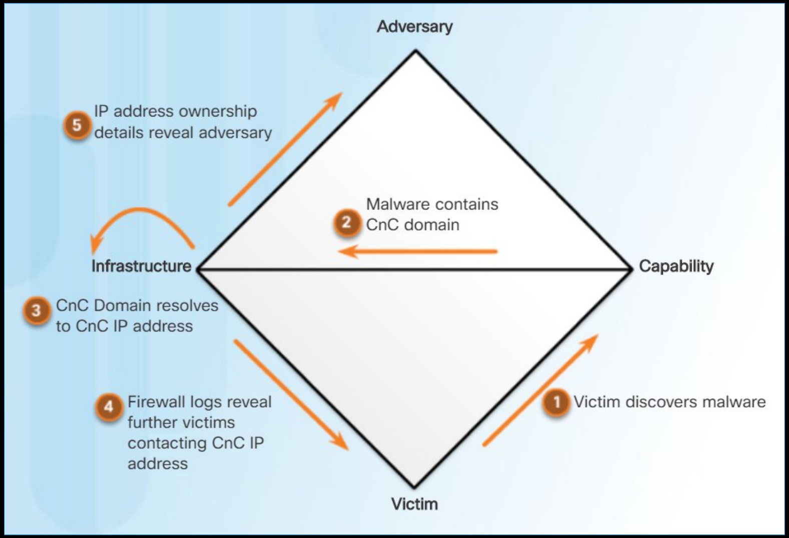
- The ***adversary*** uses ***capabilities*** over ***infrastructure*** to attack the ***victim***.
  - Each line in the model shows how each part reached the other.

The Diamond Model of Intrusion

# Pivoting Across the Diamond Model

- The Diamond Model is ideal for illustrating how the adversary pivots from one event to the next.

**Example**


Adversary
5 IP address ownership details reveal adversary
2 Malware contains CnC domain
Infrastructure
Capability
3 CnC Domain resolves to CnC IP address
4 Firewall logs reveal further victims contacting CnC IP address
1 Victim discovers malware
Victim

1) An employee reports that his computer is acting abnormally and a scan indicates the computer is infected with malware.
2) An analysis of the malware reveals that the malware contains a list of CnC domain names.
3) These domain names resolve to a list of IP addresses.
4) These IP addresses are used to investigate logs to determine if other victims in the organization are using the CnC channel.
5) The IP addresses are also used to identify the adversary.

# The Diamond Model and the Cyber Kill Chain

- The example illustrates the process used by an adversary as they **traverse the Cyber Kill Chain**.

Reconnaissance

Weaponization

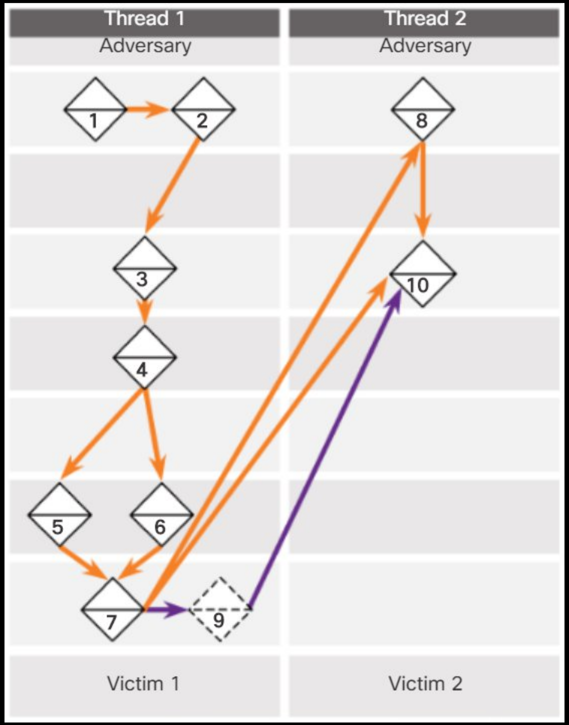
Delivery

Exploitation

Installation

C2

Action on Objectives

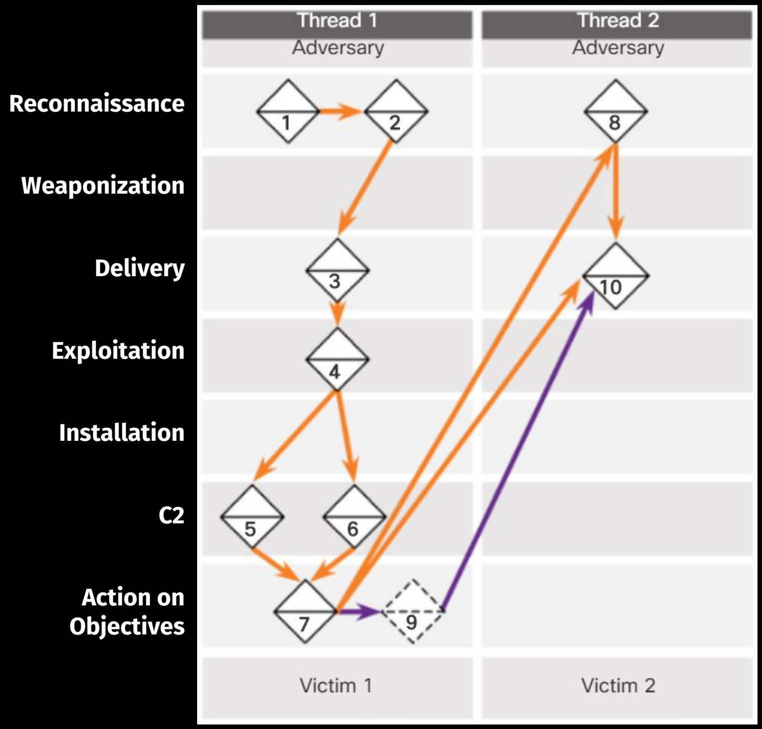


1) Adversary conducts a web search for victim company Gadgets, Inc. receiving as part of the results their domain gadgets.com.
2) Adversary searches "network administrator gadget.com" and discovers the network administrators' email addresses.
3) Adversary sends phishing emails with a Trojan horse attached to the network administrators.
4) One network administrator (NA1) opens the malicious attachment which executes the enclosed exploit.
5) NA1's host registers with a CnC controller by sending an HTTP Post message and receiving an HTTP Response in return.
6) Analysis of the malware identifies additional backup IP addresses.
7) Through a CnC HTTP response message sent to NA1's host, the malware begins to act as a proxy for new TCP connections.

The Diamond Model of Intrusion

# The Diamond Model and the Cyber Kill Chain (Cont.)

- The example illustrates the process used by an adversary as they traverse the Cyber Kill Chain.



8) Through the proxy established on NA1's host, Adversary does a web search for "most important research ever" and finds Victim 2, Interesting Research Inc.

9) Adversary checks NA1's email contact list for any contacts from Interesting Research Inc. and discovers the contact for the Interesting Research Inc. Chief Research Officer.

10) Chief Research Officer of Interesting Research Inc. receives a spear-phish email from Gadget Inc.'s NA1's email address sent from NA1's host with the same payload as observed in Event 3.

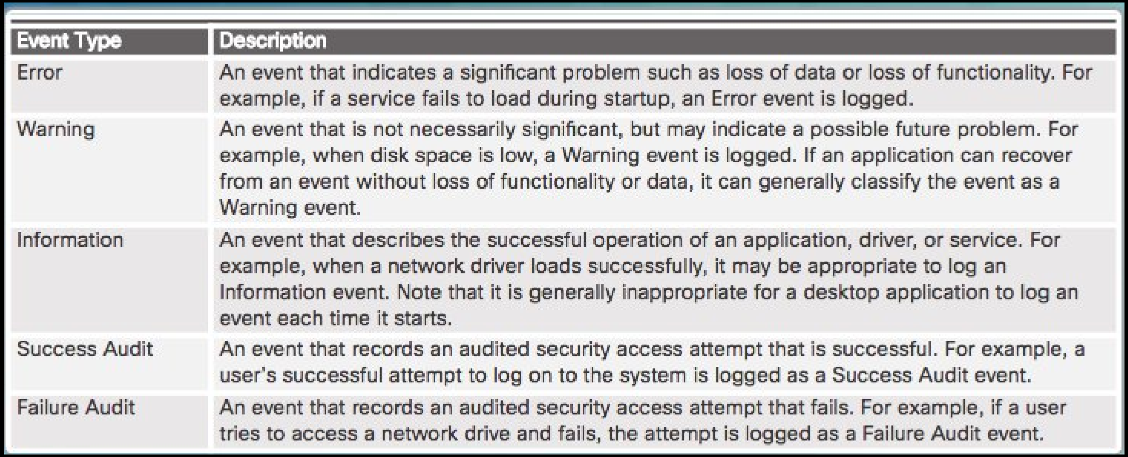
The adversary now has two compromised victims from which additional attacks can be launched.

# Log Analysis

### End Device Logs

# Host Logs

- Host-based intrusion protection (HIDS) runs on individual hosts.
  - HIDS not only detects intrusions, but in the form of host-based firewalls, can also prevent intrusion.
  - Creates logs and stores them on the host.
  - Microsoft Windows host logs are visible locally through Event Viewer.
  - Event Viewer keeps four types of logs: Application logs, System logs, Setup logs, and Security logs.



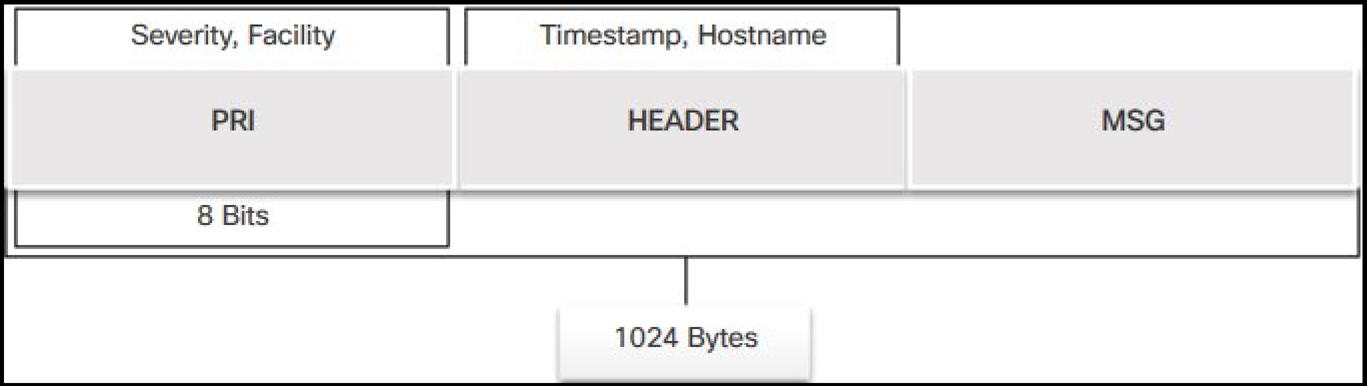
| Event Type | Description |
|---|---|
| Error | An event that indicates a significant problem such as loss of data or loss of functionality. For example, if a service fails to load during startup, an Error event is logged. |
| Warning | An event that is not necessarily significant, but may indicate a possible future problem. For example, when disk space is low, a Warning event is logged. If an application can recover from an event without loss of functionality or data, it can generally classify the event as a Warning event. |
| Information | An event that describes the successful operation of an application, driver, or service. For example, when a network driver loads successfully, it may be appropriate to log an Information event. Note that it is generally inappropriate for a desktop application to log an event each time it starts. |
| Success Audit | An event that records an audited security access attempt that is successful. For example, a user's successful attempt to log on to the system is logged as a Success Audit event. |
| Failure Audit | An event that records an audited security access attempt that fails. For example, if a user tries to access a network drive and fails, the attempt is logged as a Failure Audit event. |

**Windows Host Log Event Types**

## End Device Logs

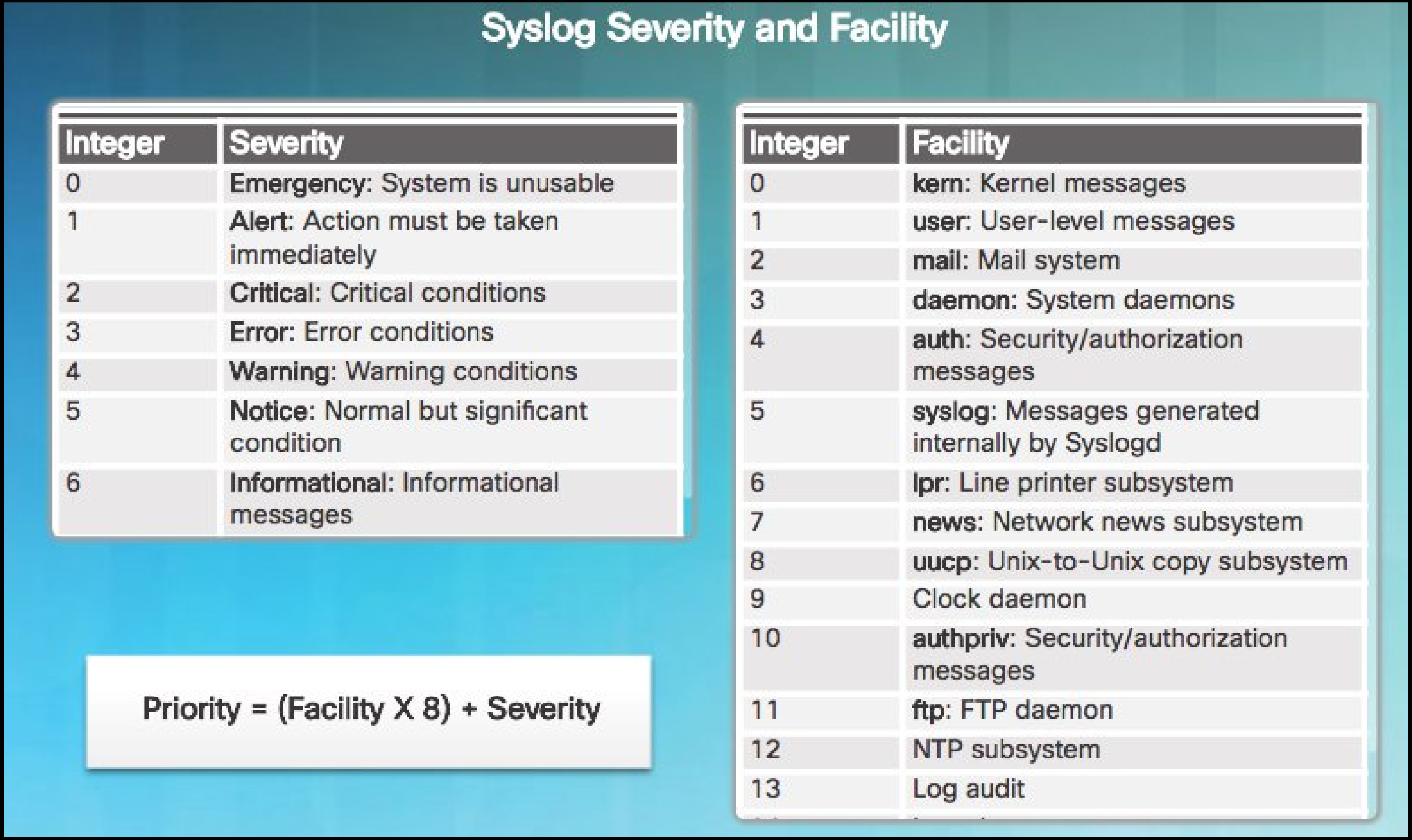
# Syslog

- Many types of network devices can be configured to log events to syslog servers.
  - Client/server protocol
  - Syslog messages have three parts: PRI (priority), HEADER, and MSG (message text).
    - PRI consists of two elements, the Facility and Severity of the message.
    - Facility consists of broad categories of sources that generated the message, such as the system, process, or application, directs message to appropriate log file.
    - Severity is a value from 0-7 that defines the severity of the message.



End Device Logs

# Syslog (Cont.)



## Syslog Severity and Facility

| Integer | Severity |
|---|---|
| 0 | **Emergency:** System is unusable |
| 1 | **Alert:** Action must be taken immediately |
| 2 | **Critical:** Critical conditions |
| 3 | **Error:** Error conditions |
| 4 | **Warning:** Warning conditions |
| 5 | **Notice:** Normal but significant condition |
| 6 | **Informational:** Informational messages |

Priority = (Facility X 8) + Severity

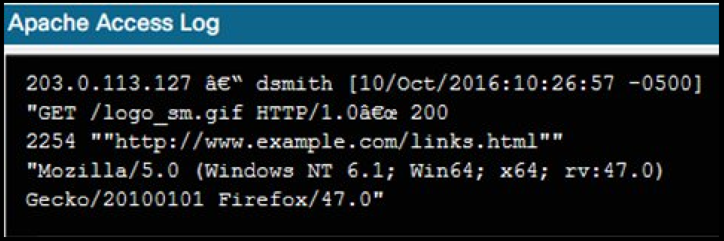
| Integer | Facility |
|---|---|
| 0 | **kern:** Kernel messages |
| 1 | **user:** User-level messages |
| 2 | **mail:** Mail system |
| 3 | **daemon:** System daemons |
| 4 | **auth:** Security/authorization messages |
| 5 | **syslog:** Messages generated internally by Syslogd |
| 6 | **lpr:** Line printer subsystem |
| 7 | **news:** Network news subsystem |
| 8 | **uucp:** Unix-to-Unix copy subsystem |
| 9 | Clock daemon |
| 10 | **authpriv:** Security/authorization messages |
| 11 | **ftp:** FTP daemon |
| 12 | NTP subsystem |
| 13 | Log audit |

End Device Logs

# Server Logs

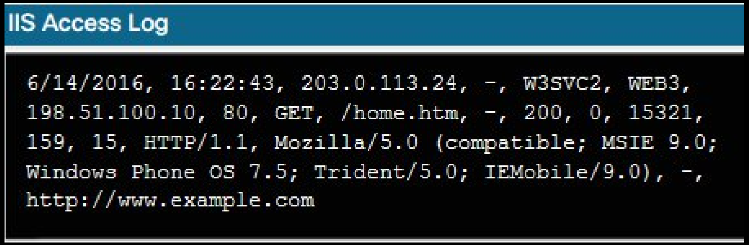
- Server Logs are an essential source of data for network security monitoring.
  - Email and web servers keep access and error logs.
  - DNS proxy server logs document all DNS queries and responses that occur on the network.
  - DNS proxy logs can identify hosts that visited dangerous websites and identify DNS data exfiltration and connections to malware CnC servers.

Web Server Logs



Apache Access Log

```
203.0.113.127 â€“ dsmith [10/Oct/2016:10:26:57 -0500]
"GET /logo_sm.gif HTTP/1.0â€œ 200
2254 ""http://www.example.com/links.html""
"Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:47.0)
Gecko/20100101 Firefox/47.0"
```



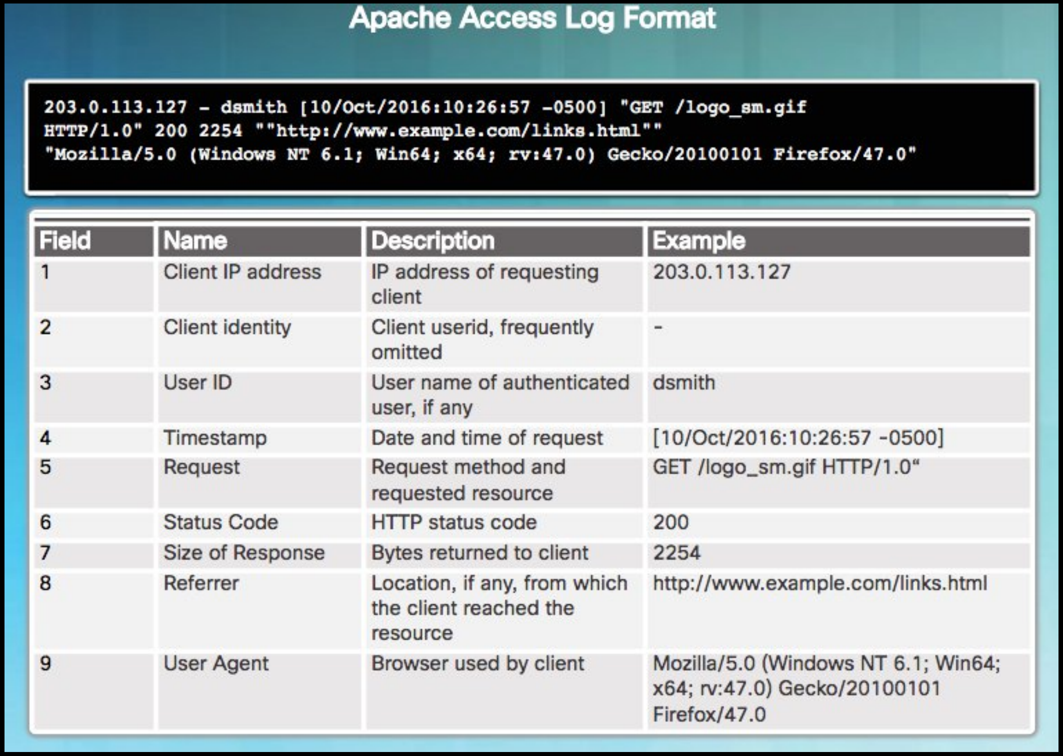
IIS Access Log

```
6/14/2016, 16:22:43, 203.0.113.24, -, W3SVC2, WEB3,
198.51.100.10, 80, GET, /home.htm, -, 200, 0, 15321,
159, 15, HTTP/1.1, Mozilla/5.0 (compatible; MSIE 9.0;
Windows Phone OS 7.5; Trident/5.0; IEMobile/9.0), -,
http://www.example.com
```

End Device Logs

# Apache Webserver Access Logs

- Apache Webserver access logs record the requests for resources from clients to the server.
  - Two log formats
    - Common log format (CLF)
    - Combined log format, which is CLF with the addition of the referrer and user agent fields



Apache Access Log Format

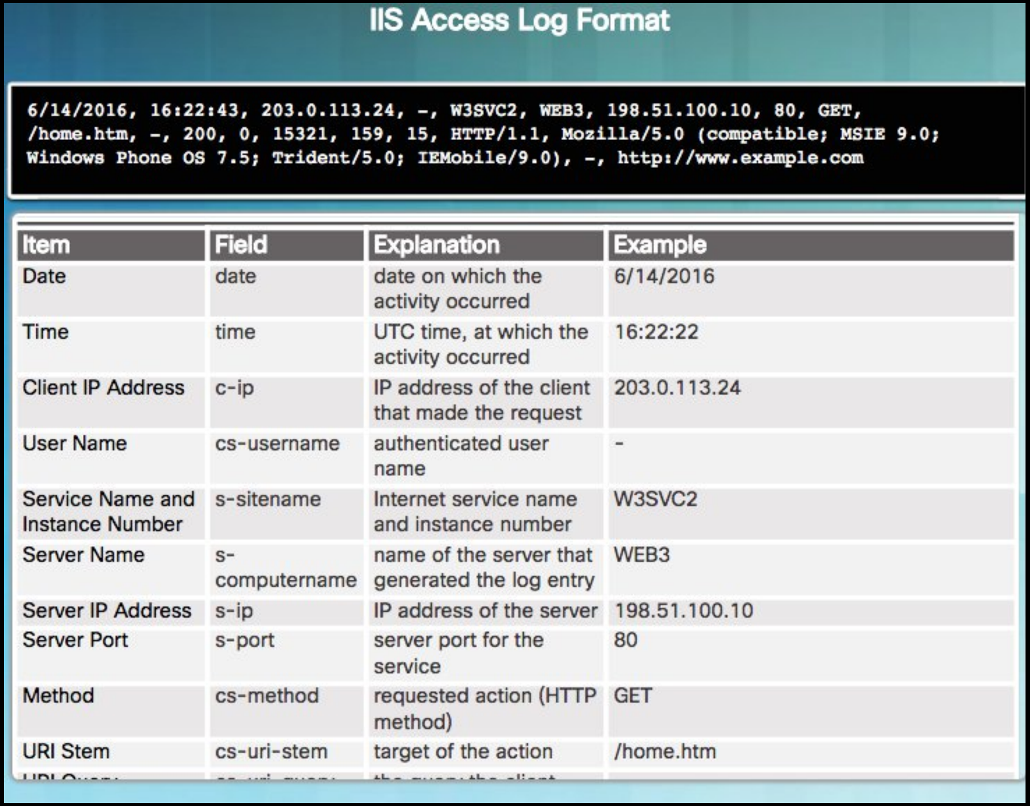
```
203.0.113.127 - dsmith [10/Oct/2016:10:26:57 -0500] "GET /logo_sm.gif
HTTP/1.0" 200 2254 ""http://www.example.com/links.html""
"Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:47.0) Gecko/20100101 Firefox/47.0"
```

| Field | Name | Description | Example |
|---|---|---|---|
| 1 | Client IP address | IP address of requesting client | 203.0.113.127 |
| 2 | Client identity | Client userid, frequently omitted | - |
| 3 | User ID | User name of authenticated user, if any | dsmith |
| 4 | Timestamp | Date and time of request | [10/Oct/2016:10:26:57 -0500] |
| 5 | Request | Request method and requested resource | GET /logo_sm.gif HTTP/1.0“ |
| 6 | Status Code | HTTP status code | 200 |
| 7 | Size of Response | Bytes returned to client | 2254 |
| 8 | Referrer | Location, if any, from which the client reached the resource | http://www.example.com/links.html |
| 9 | User Agent | Browser used by client | Mozilla/5.0 (Windows NT 6.1; Win64; x64; rv:47.0) Gecko/20100101 Firefox/47.0 |

End Device Logs

# IIS Access Logs

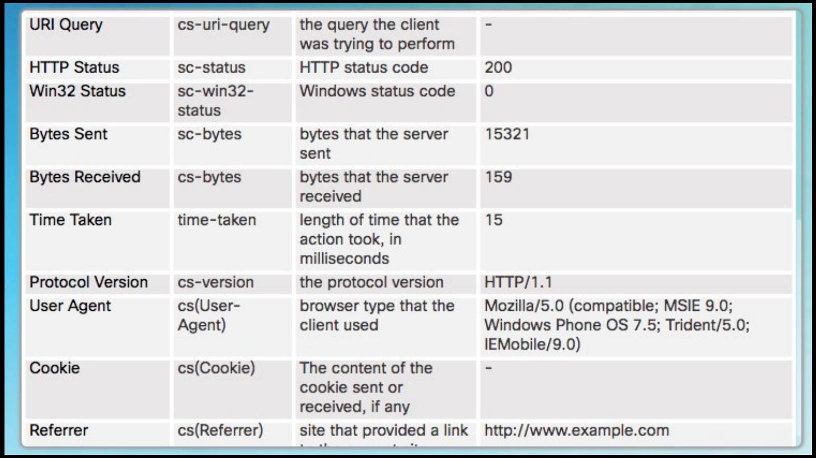
- Microsoft IIS creates access logs that can be viewed from the server with Event Viewer.



IIS Access Log Format

```
6/14/2016, 16:22:43, 203.0.113.24, -, W3SVC2, WEB3, 198.51.100.10, 80, GET,
/home.htm, -, 200, 0, 15321, 159, 15, HTTP/1.1, Mozilla/5.0 (compatible; MSIE 9.0;
Windows Phone OS 7.5; Trident/5.0; IEMobile/9.0), -, http://www.example.com
```

| Item | Field | Explanation | Example |
|---|---|---|---|
| Date | date | date on which the activity occurred | 6/14/2016 |
| Time | time | UTC time, at which the activity occurred | 16:22:22 |
| Client IP Address | c-ip | IP address of the client that made the request | 203.0.113.24 |
| User Name | cs-username | authenticated user name | - |
| Service Name and Instance Number | s-sitename | Internet service name and instance number | W3SVC2 |
| Server Name | s-computername | name of the server that generated the log entry | WEB3 |
| Server IP Address | s-ip | IP address of the server | 198.51.100.10 |
| Server Port | s-port | server port for the service | 80 |
| Method | cs-method | requested action (HTTP method) | GET |
| URI Stem | cs-uri-stem | target of the action | /home.htm |



| | | | |
|---|---|---|---|
| URI Query | cs-uri-query | the query the client was trying to perform | - |
| HTTP Status | sc-status | HTTP status code | 200 |
| Win32 Status | sc-win32-status | Windows status code | 0 |
| Bytes Sent | sc-bytes | bytes that the server sent | 15321 |
| Bytes Received | cs-bytes | bytes that the server received | 159 |
| Time Taken | time-taken | length of time that the action took, in milliseconds | 15 |
| Protocol Version | cs-version | the protocol version | HTTP/1.1 |
| User Agent | cs(User-Agent) | browser type that the client used | Mozilla/5.0 (compatible; MSIE 9.0; Windows Phone OS 7.5; Trident/5.0; IEMobile/9.0) |
| Cookie | cs(Cookie) | The content of the cookie sent or received, if any | - |
| Referrer | cs(Referrer) | site that provided a link | http://www.example.com |

### Network Logs

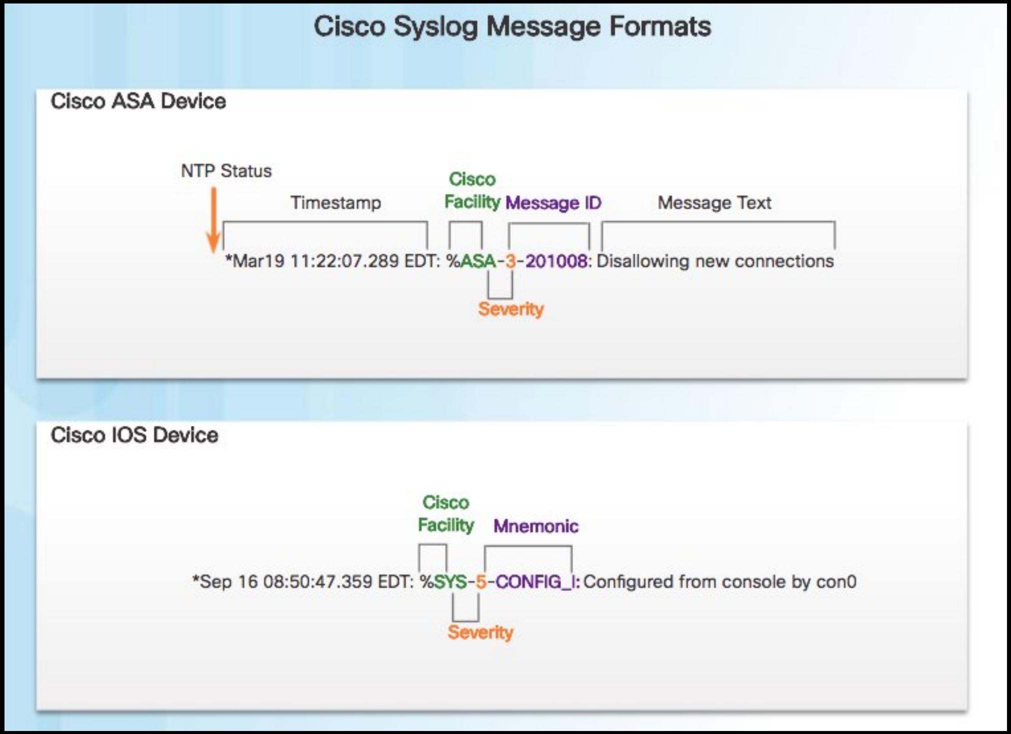
# Proxy Logs

- Proxy servers contain valuable logs that are a primary source of data for network security monitoring.
    - Proxy servers make requests for resources and return them to the client.
    - Generate logs of all requests and responses.
    - Can be analyzed to determine which hosts are making the requests, whether the destinations are safe or potentially malicious, and to gain insights into the kind of resources that have been downloaded.
- Web proxies provide data that helps determine whether responses from the web were generated in response to legitimate requests or only appear to be responses.
- Open DNS offers a hosted DNS service that extends the capability of DNS to include security enhancements.
    - DNS super proxy
    - Apply real-time threat intelligence to managing DNS access and the security of DNS records

Network Logs

# Logging from Cisco Devices

- Cisco devices can be configured to submit events and alerts to security management platforms using SNMP or syslog.



```
*Mar19 11:22:07.289 EDT: %ASA-3-201008: Disallowing new connections
```

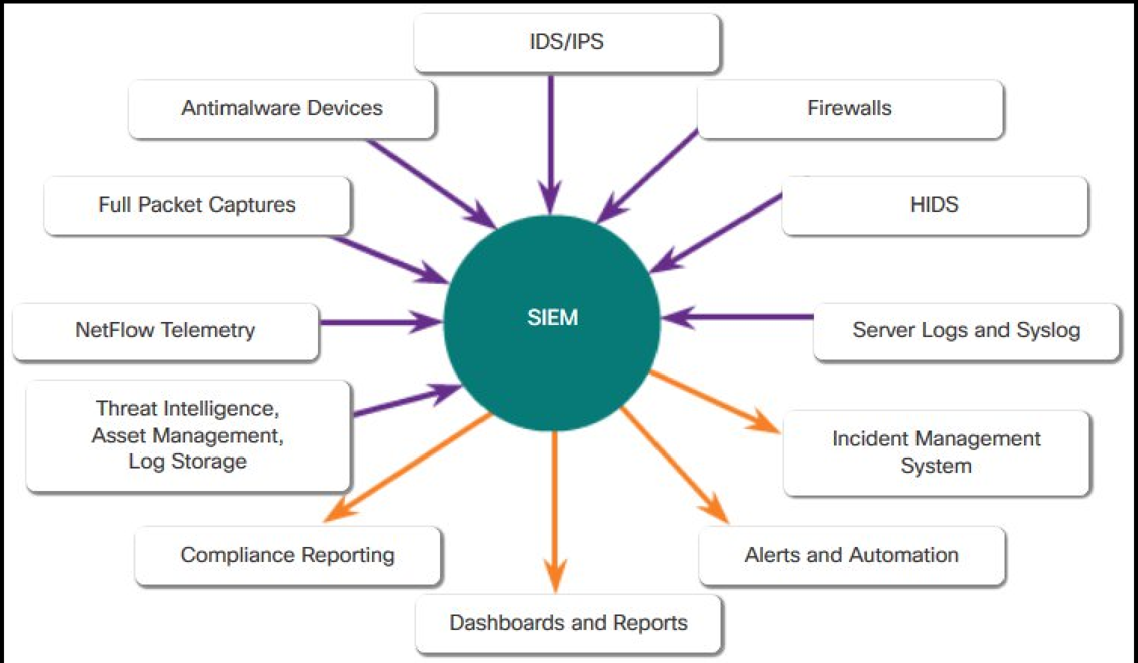
```
*Sep 16 08:50:47.359 EDT: %SYS-5-CONFIG_I: Configured from console by con0
```

End Device Logs

# SIEM and Log Collection

- Security Information and Event Management (SIEM) technology
  - Provides real-time reporting and long-term analysis of security events.
  - Uses the following functions: Log collection, Normalization, Correlation, Aggregation, Reporting, Compliance
  - A popular SIEM is Splunk..

SIEM Components



# Security Information Event Management (SIEM)

**Tasks and Capabilities:**

- **Log Data Aggregation** – (linux, apps, database, switches, routers)
- **Log Forensics** – search raw and formatted logs
- **Event Correlation and Alerting** – set rules, thresholds
- **File integrity Monitoring** – track changes
- **Log Analysis & Dashboard** – real-time log data in chart, graphs, reports
- **User Monitoring** – track suspicious user activity or privilege escalation
- **Object Access Auditing** – tracking access, editing and deletion of files
- **Compliance Reports** – pre-defined reports
- **Log Data Retention** – internal audits, historical data

Commercial Examples:

IBM Qradar
LogRhythm
Splunk

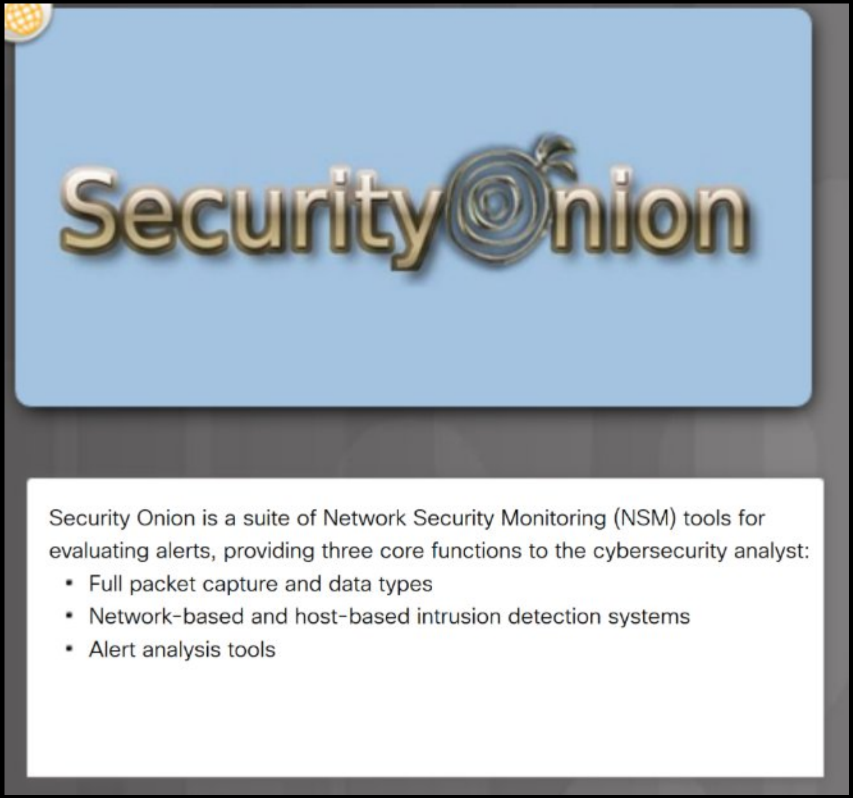
Open Source:

Loggly
Sumo Logic
Logzilla
Wazuh

Sources of Alerts

# Security Onion

- Security Onion is an open-source suite of Network Security Monitoring (NSM) tools that run on an Ubuntu Linux distribution.
- Some components of Security Onion are owned and maintained by corporations, such as Cisco and Riverbend Technologies, but are made available as open source.



Security Onion is a suite of Network Security Monitoring (NSM) tools for evaluating alerts, providing three core functions to the cybersecurity analyst:

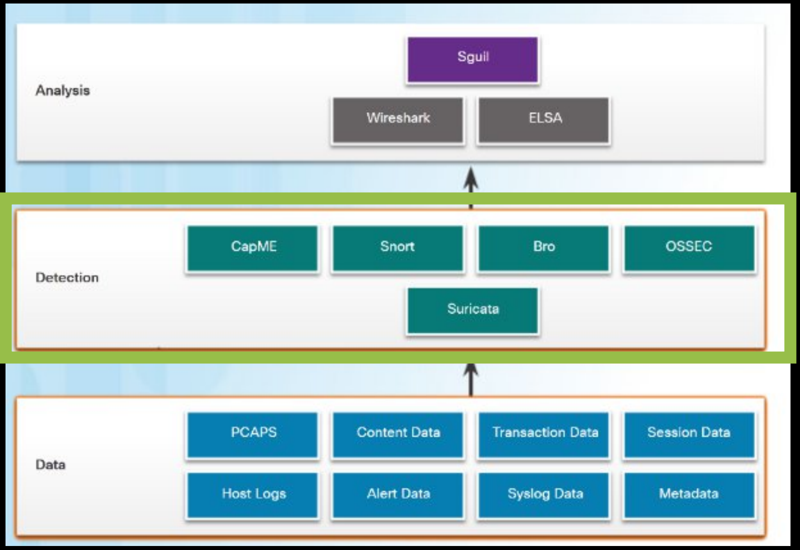
- Full packet capture and data types
- Network-based and host-based intrusion detection systems
- Alert analysis tools

Sources of Alerts

# Detection Tools for Collection

- **CapME** provides the cybersecurity analyst with an easy-to-read means of viewing an entire Layer 4 session.
- **Snort** uses rules and signatures to generate alerts.
- **Bro** uses policies, in the form of scripts that determine what data to log and when to issue alert notifications.
- **OSSEC** actively monitors host system operations, including conducting file integrity monitoring, local log monitoring, system process monitoring, and rootkit detection.
- **Suricata** uses native multithreading, which allows the distribution of packet stream processing across multiple processor cores.

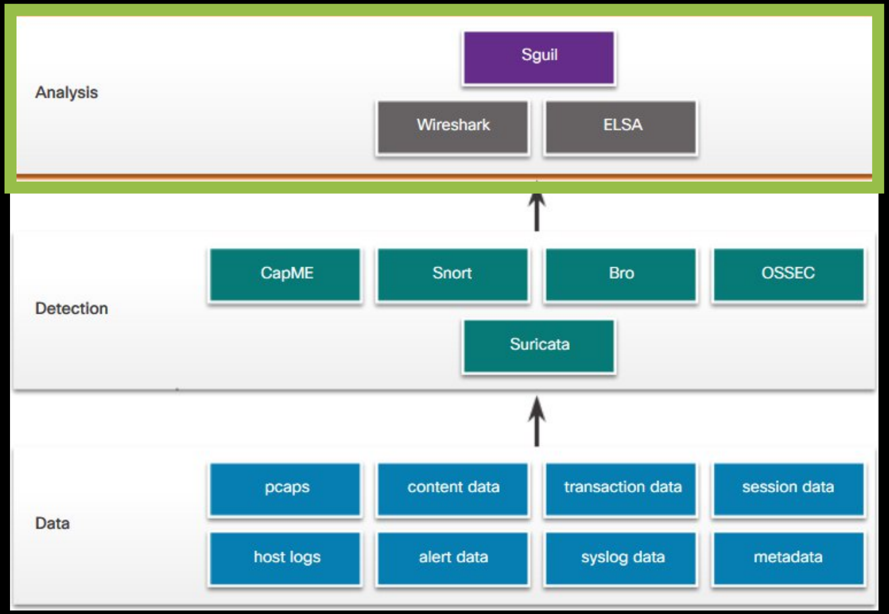
## A Security Onion Architecture



Sources of Alerts

# Analysis Tools

A Security Onion Architecture



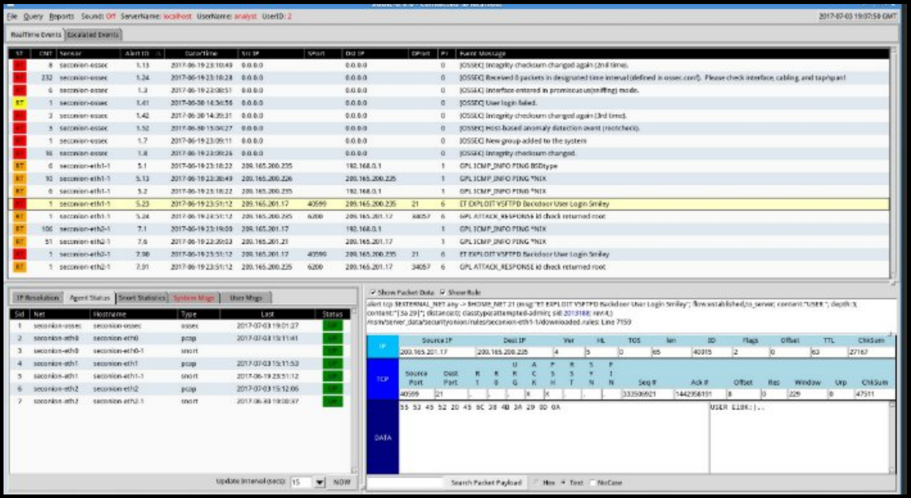
- **Sguil** – This provides a high-level cybersecurity analysts' console for investigating security alerts from a wide variety of sources.
- **ELSA** –replaced by ELK
- **Wireshark** – This is a packet capture application that is integrated into the Security Onion suite.

## Sources of Alerts

# Alert Generation

- Alerts are generated in Security Onion by many sources including Snort, Bro, Suricata, and OSSEC, among others.
- Sguil provides a console that integrates alerts from multiple sources into a timestamped queue.
- Alerts will generally include the following five-tuples information:
    - SrcIP - the source IP address for the event.
    - SPort - the source (local) Layer 4 port for the event.
    - DstIP - the destination IP for the event.
    - DPort - the destination Layer 4 port for the event.
    - Pr - the IP protocol number for the event.

**Sguil Window**



File Query Reports Sound: Off ServerName: localhost UserName: analyst UserID: 2
2017-07-03 19:07:59 GMT
RealTime Events | Escalated Events

| ST | CNT | Sensor | Alert ID | Date/Time | Src IP | SPort | Dst IP | DPort | Pr | Event Message |
|---|---|---|---|---|---|---|---|---|---|---|
| RT | 8 | seconion-ossec | 1.13 | 2017-06-19 23:10:49 | 0.0.0.0 | | 0.0.0.0 | | 0 | [OSSEC] Integrity checksum changed again (2nd time). |
| RT | 232 | seconion-ossec | 1.24 | 2017-06-19 23:10:28 | 0.0.0.0 | | 0.0.0.0 | | 0 | [OSSEC] Received 0 packets in designated time interval (defined in ossec.conf). Please check interface, cabling, and tap/span! |
| RT | 6 | seconion-ossec | 1.3 | 2017-06-19 23:08:51 | 0.0.0.0 | | 0.0.0.0 | | 0 | [OSSEC] Interface entered in promiscuous(sniffing) mode. |
| RT | 1 | seconion-ossec | 1.41 | 2017-06-30 14:34:56 | 0.0.0.0 | | 0.0.0.0 | | 0 | [OSSEC] User login failed. |
| RT | 3 | seconion-ossec | 1.42 | 2017-06-30 14:39:31 | 0.0.0.0 | | 0.0.0.0 | | 0 | [OSSEC] Integrity checksum changed again (3rd time). |
| RT | 5 | seconion-ossec | 1.52 | 2017-06-30 15:04:27 | 0.0.0.0 | | 0.0.0.0 | | 0 | [OSSEC] Host-based anomaly detection event (rootcheck). |
| RT | 1 | seconion-ossec | 1.7 | 2017-06-19 23:09:11 | 0.0.0.0 | | 0.0.0.0 | | 0 | [OSSEC] New group added to the system |
| RT | 16 | seconion-ossec | 1.8 | 2017-06-19 23:09:26 | 0.0.0.0 | | 0.0.0.0 | | 0 | [OSSEC] Integrity checksum changed. |
| RT | 6 | seconion-eth1-1 | 5.1 | 2017-06-19 23:18:22 | 209.165.200.235 | | 192.168.0.1 | | 1 | GPL ICMP_INFO PING BSDtype |
| RT | 10 | seconion-eth1-1 | 5.13 | 2017-06-19 23:38:49 | 209.165.200.226 | | 209.165.200.235 | | 1 | GPL ICMP_INFO PING *NIX |
| RT | 6 | seconion-eth1-1 | 5.2 | 2017-06-19 23:18:22 | 209.165.200.235 | | 192.168.0.1 | | 1 | GPL ICMP_INFO PING *NIX |
| RT | 1 | seconion-eth1-1 | 5.23 | 2017-06-19 23:51:12 | 209.165.201.17 | 40599 | 209.165.200.235 | 21 | 6 | ET EXPLOIT VSFTPD Backdoor User Login Smiley |
| RT | 1 | seconion-eth1-1 | 5.24 | 2017-06-19 23:51:12 | 209.165.200.235 | 6200 | 209.165.201.17 | 34057 | 6 | GPL ATTACK_RESPONSE id check returned root |
| RT | 106 | seconion-eth2-1 | 7.1 | 2017-06-19 23:19:00 | 209.165.201.17 | | 192.168.0.1 | | 1 | GPL ICMP_INFO PING *NIX |
| RT | 51 | seconion-eth2-1 | 7.6 | 2017-06-19 22:20:03 | 209.165.201.21 | | 209.165.201.17 | | 1 | GPL ICMP_INFO PING *NIX |
| RT | 1 | seconion-eth2-1 | 7.90 | 2017-06-19 23:51:12 | 209.165.201.17 | 40599 | 209.165.200.235 | 21 | 6 | ET EXPLOIT VSFTPD Backdoor User Login Smiley |
| RT | 1 | seconion-eth2-1 | 7.91 | 2017-06-19 23:51:12 | 209.165.200.235 | 6200 | 209.165.201.17 | 34057 | 6 | GPL ATTACK_RESPONSE id check returned root |

IP Resolution | Agent Status | Snort Statistics | System Msgs | User Msgs

| Sid | Net | Hostname | Type | Last | Status |
|---|---|---|---|---|---|
| 1 | seconion-ossec | seconion-ossec | ossec | 2017-07-03 19:01:27 | UP |
| 2 | seconion-eth0 | seconion-eth0 | pcap | 2017-07-03 15:11:41 | UP |
| 3 | seconion-eth0 | seconion-eth0-1 | snort | | UP |
| 4 | seconion-eth1 | seconion-eth1 | pcap | 2017-07-03 15:11:53 | UP |
| 5 | seconion-eth1 | seconion-eth1-1 | snort | 2017-06-19 23:51:12 | UP |
| 6 | seconion-eth2 | seconion-eth2 | pcap | 2017-07-03 15:12:06 | UP |
| 7 | seconion-eth2 | seconion-eth2-1 | snort | 2017-06-30 19:00:37 | UP |

Update Interval (secs): 15 NOW

☑ Show Packet Data ☑ Show Rule

```
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"ET EXPLOIT VSFTPD Backdoor User Login Smiley"; flow:established,to_server; content:"USER "; depth:5; content:"|3a 29|"; distance:0; classtype:attempted-admin; sid:2013188; rev:4;)
/nsm/server_data/securityonion/rules/seconion-eth1-1/downloaded.rules: Line 7159
```

| IP | Source IP | Dest IP | Ver | HL | TOS | len | ID | Flags | Offset | TTL | ChkSum |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | 209.165.201.17 | 209.165.200.235 | 4 | 5 | 0 | 65 | 40015 | 2 | 0 | 63 | 27167 |

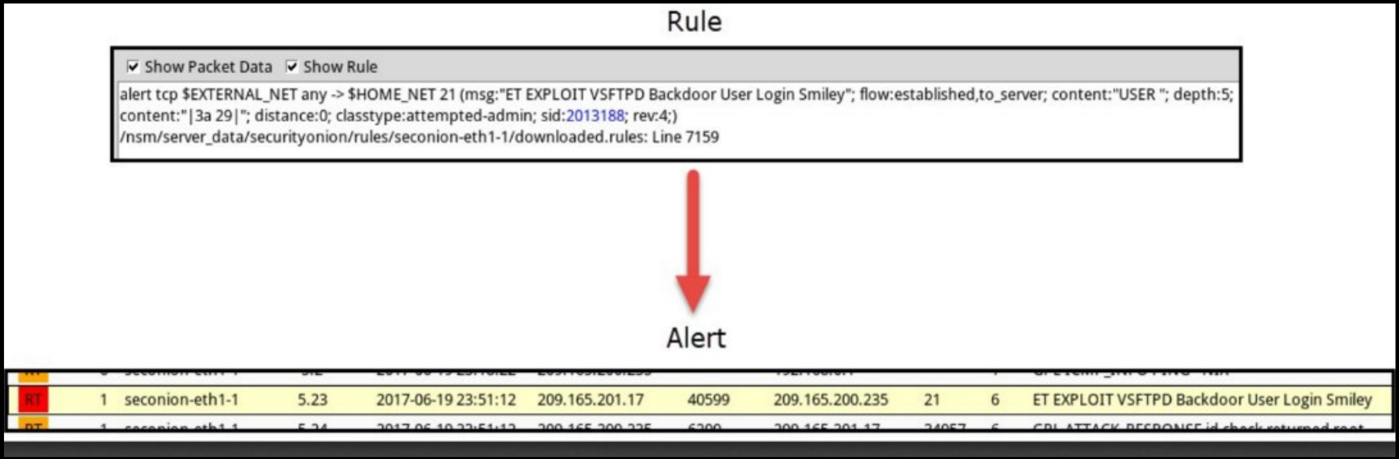
| TCP | Source Port | Dest Port | R 1 | R 0 | URG | ACK | PSH | RST | SYN | FIN | Seq # | Ack # | Offset | Res | Window | Urp | ChkSum |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | 40599 | 21 | . | . | . | X | X | . | . | . | 333506921 | 1442958191 | 8 | 0 | 229 | 0 | 47511 |

```
DATA
55 53 45 52 20 45 6C 38 4B 3A 29 0D 0A        USER E18K:)..
```

Sources of Alerts

# Rules and Alerts

- Alerts can come from a number of sources:
  - NIDS - Snort, Bro and Suricata
  - HIDS – OSSEC
  - Asset management and monitoring - Passive Asset Detection System (PADS)
  - HTTP, DNS, and TCP transactions - Recorded by Bro and pcaps
  - Syslog messages - Multiple sources



```
alert tcp $EXTERNAL_NET any -> $HOME_NET 21 (msg:"ET EXPLOIT VSFTPD Backdoor User Login Smiley"; flow:established,to_server; content:"USER "; depth:5;
content:"|3a 29|"; distance:0; classtype:attempted-admin; sid:2013188; rev:4;)
/nsm/server_data/securityonion/rules/seconion-eth1-1/downloaded.rules: Line 7159
```

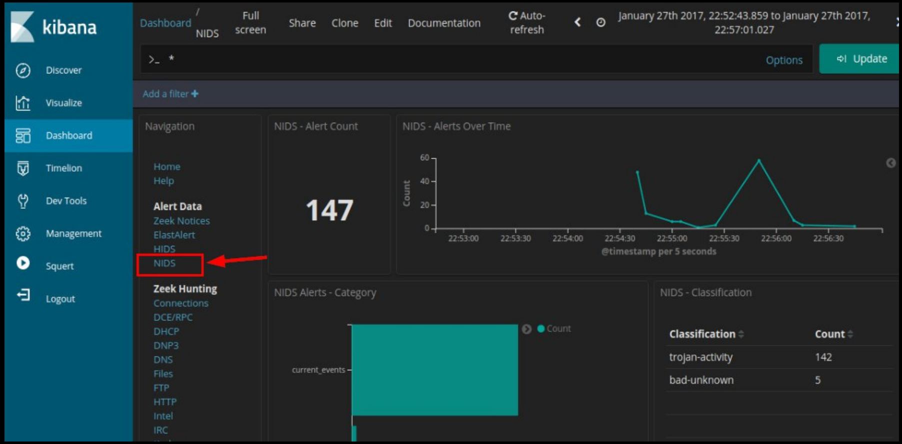
```
RT   1   seconion-eth1-1   5.23   2017-06-19 23:51:12   209.165.201.17   40599   209.165.200.235   21   6   ET EXPLOIT VSFTPD Backdoor User Login Smiley
```

Sources of Alerts

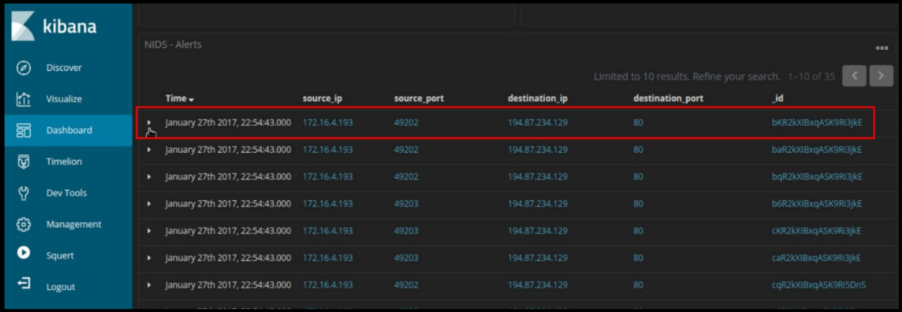
# Threat Analysis

- Alerts and events can be visualised in Kibana, capME

Kibana capME

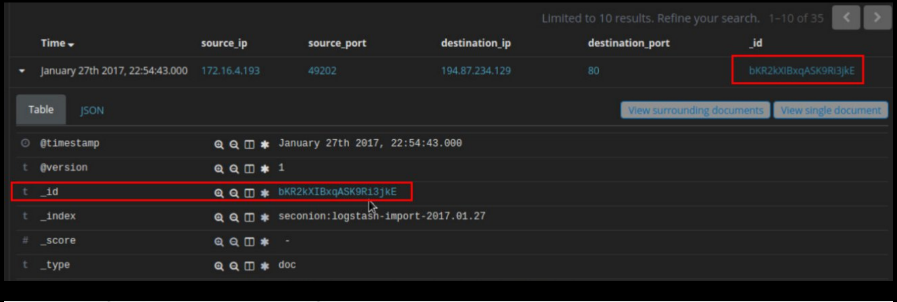


| Classification | Count |
|---|---|
| trojan-activity | 142 |
| bad-unknown | 5 |



Limited to 10 results. Refine your search. 1–10 of 35

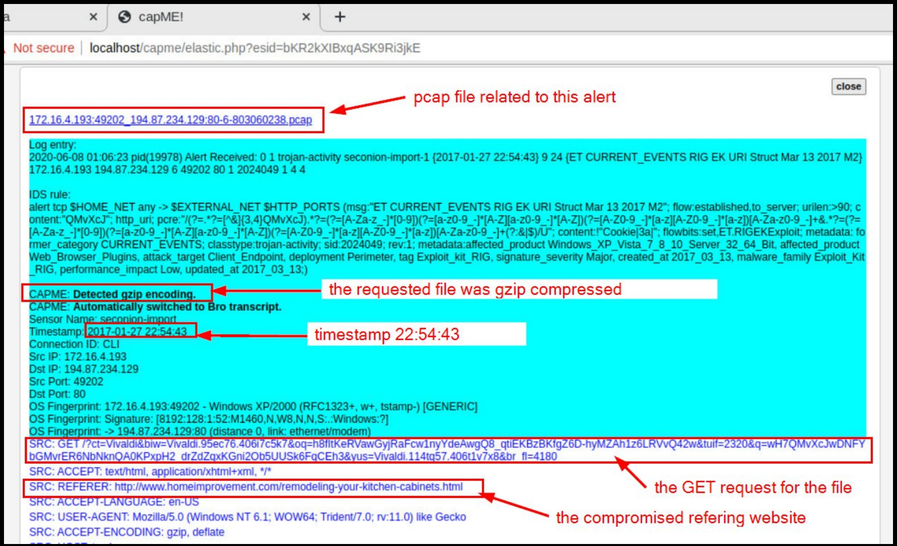
| Time | source_ip | source_port | destination_ip | destination_port | _id |
|---|---|---|---|---|---|
| January 27th 2017, 22:54:43.000 | 172.16.4.193 | 49202 | 194.87.234.129 | 80 | bKR2kXIBxqASK9Ri3jkE |
| January 27th 2017, 22:54:43.000 | 172.16.4.193 | 49202 | 194.87.234.129 | 80 | baR2kXIBxqASK9Ri3jkE |
| January 27th 2017, 22:54:43.000 | 172.16.4.193 | 49202 | 194.87.234.129 | 80 | bqR2kXIBxqASK9Ri3jkE |
| January 27th 2017, 22:54:43.000 | 172.16.4.193 | 49203 | 194.87.234.129 | 80 | b6R2kXIBxqASK9Ri3jkE |
| January 27th 2017, 22:54:43.000 | 172.16.4.193 | 49203 | 194.87.234.129 | 80 | cKR2kXIBxqASK9Ri3jkE |
| January 27th 2017, 22:54:43.000 | 172.16.4.193 | 49203 | 194.87.234.129 | 80 | caR2kXIBxqASK9Ri3jkE |
| January 27th 2017, 22:54:43.000 | 172.16.4.193 | 49202 | 194.87.234.129 | 80 | cqR2kXIBxqASK9Ri5DnS |



Limited to 10 results. Refine your search. 1–10 of 35

| Time | source_ip | source_port | destination_ip | destination_port | _id |
|---|---|---|---|---|---|
| January 27th 2017, 22:54:43.000 | 172.16.4.193 | 49202 | 194.87.234.129 | 80 | bKR2kXIBxqASK9Ri3jkE |

| | |
|---|---|
| @timestamp | January 27th 2017, 22:54:43.000 |
| @version | 1 |
| _id | bKR2kXIBxqASK9Ri3jkE |
| _index | seconion:logstash-import-2017.01.27 |
| _score | - |
| _type | doc |



Not secure | localhost/capme/elastic.php?esid=bKR2kXIBxqASK9Ri3jkE

172.16.4.193:49202_194.87.234.129:80-6-803060238.pcap

pcap file related to this alert

```
Log entry:
2020-06-08 01:06:23 pid(19978) Alert Received: 0 1 trojan-activity seconion-import-1 {2017-01-27 22:54:43} 9 24 {ET CURRENT_EVENTS RIG EK URI Struct Mar 13 2017 M2}
172.16.4.193 194.87.234.129 6 49202 80 1 2024049 1 4 4

IDS rule:
alert tcp $HOME_NET any -> $EXTERNAL_NET $HTTP_PORTS (msg:"ET CURRENT_EVENTS RIG EK URI Struct Mar 13 2017 M2"; flow:established,to_server; urilen:>90; c
ontent:"QMvXcJ"; http_uri; pcre:"/(?=.*?=[^&]{3,4}QMvXcJ).*?=(?=[A-Za-z_-]*[0-9])(?=[a-z0-9_-]*[A-Z][a-z0-9_-]*[A-Z])(?=[A-Z0-9_-]*[a-z][A-Z0-9_-]*[a-z])[A-Za-z0-9_-]+&.*?=(?=
[A-Za-z_-]*[0-9])(?=[a-z0-9_-]*[A-Z][a-z0-9_-]*[A-Z])(?=[A-Z0-9_-]*[a-z][A-Z0-9_-]*[a-z])[A-Za-z0-9_-]+(?:&|$)/U"; content:!"Cookie|3a|"; flowbits:set,ET.RIGEKExploit; metadata: fo
rmer_category CURRENT_EVENTS; classtype:trojan-activity; sid:2024049; rev:1; metadata:affected_product Windows_XP_Vista_7_8_10_Server_32_64_Bit, affected_product
Web_Browser_Plugins, attack_target Client_Endpoint, deployment Perimeter, tag Exploit_kit_RIG, signature_severity Major, created_at 2017_03_13, malware_family Exploit_Kit
_RIG, performance_impact Low, updated_at 2017_03_13;)

CAPME: Detected gzip encoding.
CAPME: Automatically switched to Bro transcript.
Sensor Name: seconion-import
Timestamp: 2017-01-27 22:54:43
Connection ID: CLI
Src IP: 172.16.4.193
Dst IP: 194.87.234.129
Src Port: 49202
Dst Port: 80
OS Fingerprint: 172.16.4.193:49202 - Windows XP/2000 (RFC1323+, w+, tstamp-) [GENERIC]
OS Fingerprint: Signature: [8192:128:1:52:M1460,N,W8,N,N,S:.:Windows:?]
OS Fingerprint: -> 194.87.234.129:80 (distance 0, link: ethernet/modem)
SRC: GET /?ct=Vivaldi&biw=Vivaldi.95ec76.406i7c5k7&oq=h8fltKeRVawGyjRaFcw1nyYdeAwgQ8_qtiEKBzBKfgZ6D-hyMZAh1z6LRVvQ42w&tuif=2320&q=wH7QMvXcJwDNFY
bGMvrER6NbNknQA0KPxpH2_drZdZqxKGni2Ob5UUSk6FqCEh3&yus=Vivaldi.114tq57.406t1v7x8&br_fl=4180
SRC: ACCEPT: text/html, application/xhtml+xml, */*
SRC: REFERER: http://www.homeimprovement.com/remodeling-your-kitchen-cabinets.html
SRC: ACCEPT-LANGUAGE: en-US
SRC: USER-AGENT: Mozilla/5.0 (Windows NT 6.1; WOW64; Trident/7.0; rv:11.0) like Gecko
SRC: ACCEPT-ENCODING: gzip, deflate
```

the requested file was gzip compressed

timestamp 22:54:43

the GET request for the file

the compromised refering website

# Sources

Bhuyan MH, Bhattacharyya DK, Kalita JK (2014) Network anomaly detection: methods, systems and tools. IEEE Communications Surveys & Tutorials 16(1):303–336

K. Riesen and H. Bunke, "IAM graph database repository for graph based pattern recognition and machine learning," in Structural, syntactic, and statistical pattern recognition: joint IAPR international workshop, SSPR & SPR 2008, Orlando, USA, December 4–6, 2008. Proceedings, N. da Vitoria Lobo et al., Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 287–297

Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: an update. ACM SIGKDD explorations newsletter 11(1):10–18

Kenkre PS, Pai A, Colaco L (2015b) Real Time Intrusion Detection and Prevention System. Springer International Publishing, Cham, pp 405–411

Bhuyan MH, Bhattacharyya DK, Kalita JK (2014) Network anomaly detection: methods, systems and tools. IEEE Communications Surveys & Tutorials 16(1):303–336

A. Abbasi, J. Wetzels, W. Bokslag, E. Zambon, and S. Etalle, "On emulation-based network intrusion detection systems," in Research in attacks, intrusions and defenses: 17th international symposium, RAID 2014, Gothenburg, Sweden, September 17–19, 2014. Proceedings, A. Stavrou, H. Bos, and G. Portokalidis, Eds. Cham: Springer International Publishing, 2014, pp. 384–404