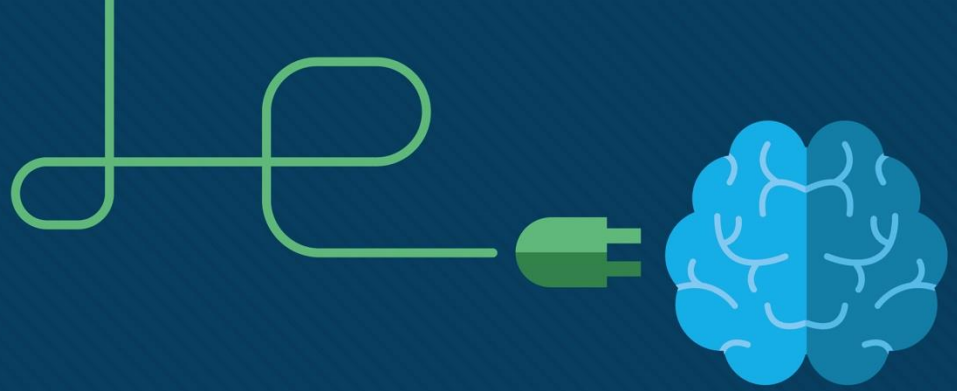


Module 26: Evaluating Alerts

CyberOps Associate v1.0

Module Objectives

Module Title: Evaluating Alerts

Module Objective: Explain the process of evaluating alerts

Topic Title	Topic Objective
Source of Alerts	Identify the structure of alerts.
Overview of Alert Evaluation	Explain how alerts are classified.

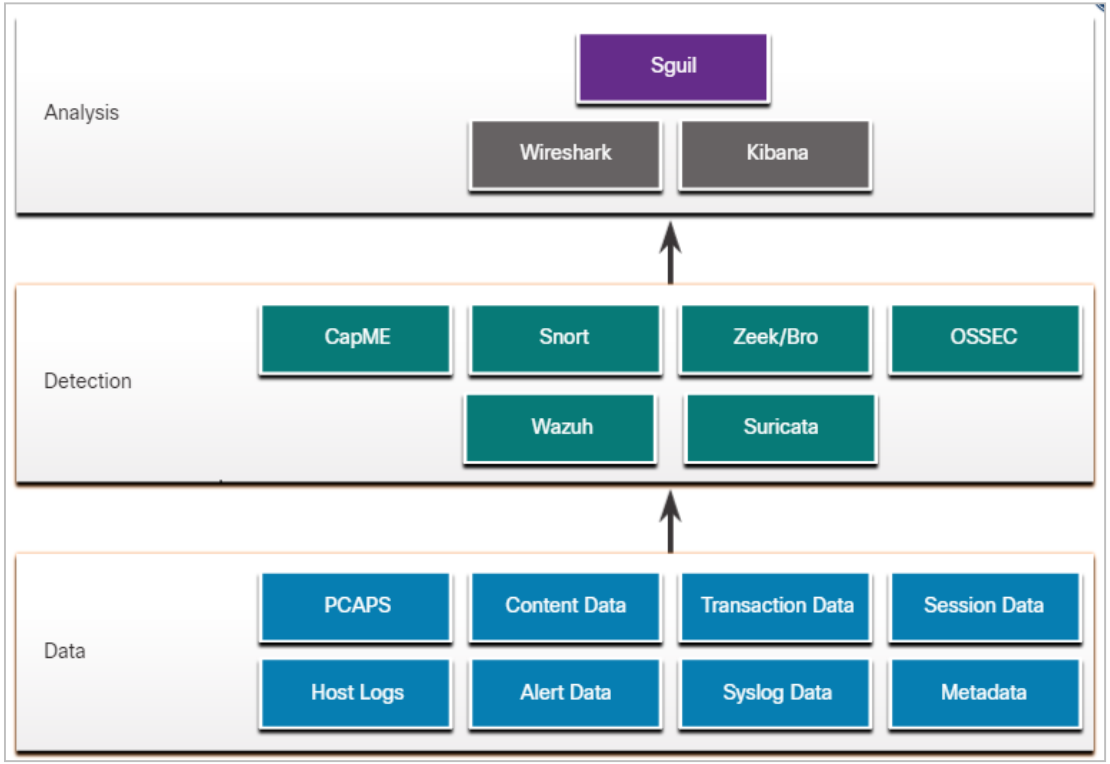
26.1 Sources of Alerts

Security Onion

- Security Onion is an open-source suite of Network Security Monitoring (NSM) tools that run on an Ubuntu Linux distribution.
- Security Onion tools provides three core functions for the cybersecurity analyst such as full packet capture and data types, network-based and host-based intrusion detection systems, and alert analyst tools.
- Security Onion can be installed as a standalone installation or as a sensor and server platform.
- Some components of Security Onion are owned and maintained by corporations, such as Cisco and Riverbend Technologies, but are made available as open source.

Detection Tools for Collecting Alert Data

- Security Onion contains many components. It is an integrated environment which is designed to simplify the deployment of a comprehensive NSM solution.
- The figure illustrates the way in which components of the Security Onion work together.



A Security Onion Architecture

Detection Tools for Collecting Alert Data (Contd.)

The following table lists the detection tools of the Security Onion:

Components	Description
CapME	This is a web application that allows viewing of pcap transcripts rendered with the tcpflow or Zeek tools.
Snort	This is a Network Intrusion Detection System (NIDS). It is an important source of alert data that is indexed in the Sguil analysis tool.
Zeek	Formerly known as Bro. This is a NIDS that uses more of a behavior-based approach to intrusion detection.
OSSEC	This is a host-based intrusion detection system (HIDS) that is integrated into Security Onion.
Wazuh	It is a full-featured solution that provides a broad spectrum of endpoint protection mechanisms including host logfile analysis, file integrity monitoring, vulnerability detection, configuration assessment, and incident response.
Suricata	This is a NIDS that uses a signature-based approach. It can also be used for inline intrusion prevention.

Analysis Tools

Security Onion integrates these various types of data and Intrusion Detection System (IDS) logs into a single platform through the following tools:

- **Sguil:** This provides a high-level console for investigating security alerts from a wide variety of sources. Sguil serves as a starting point in the investigation of security alerts. Many data sources are available by pivoting directly from Sguil to other tools.
- **Kibana:** It is an interactive dashboard interface to Elasticsearch data. It allows querying of NSM data and provides flexible visualizations of that data. It is possible to pivot from Sguil directly into Kibana to see contextualized displays.
- **Wireshark:** It is a packet capture application that is integrated into the Security Onion suit. It can be opened directly from other tools and display full packet captures relevant to an analysis.
- **Zeek:** This is a network traffic analyzer that serves as a security monitor. It inspects all traffic on a network segment and enables in-depth analysis of that data. Pivoting from Sguil into Zeek provides access to very accurate transaction logs, file content, and customized output.

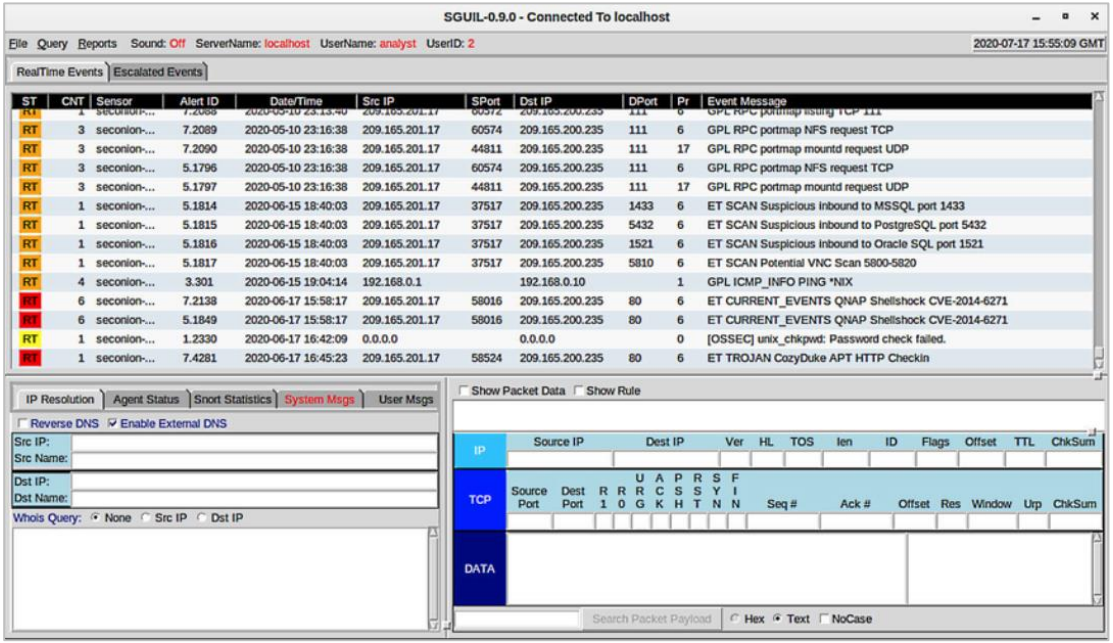
Alert Generation

- Security alerts are notification messages that are generated by NSM tools, systems, and security devices. Alerts can come in many forms depending on the source.
- In Security Onion, Sguil provides a console that integrates alerts from multiple sources into a timestamped queue.
- A cybersecurity analyst works through the security queue investigating, classifying, escalating, or retiring alerts.
- Alerts will generally include five-tuples information, as well as timestamps and information identifying which device or system generated the alert.
  - **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.

Alert Generation (Contd.)

The figure shows the Sguil application window with the queue of alerts that are waiting to be investigated in the top portion of the interface. The fields available for the real-time events are as follows:

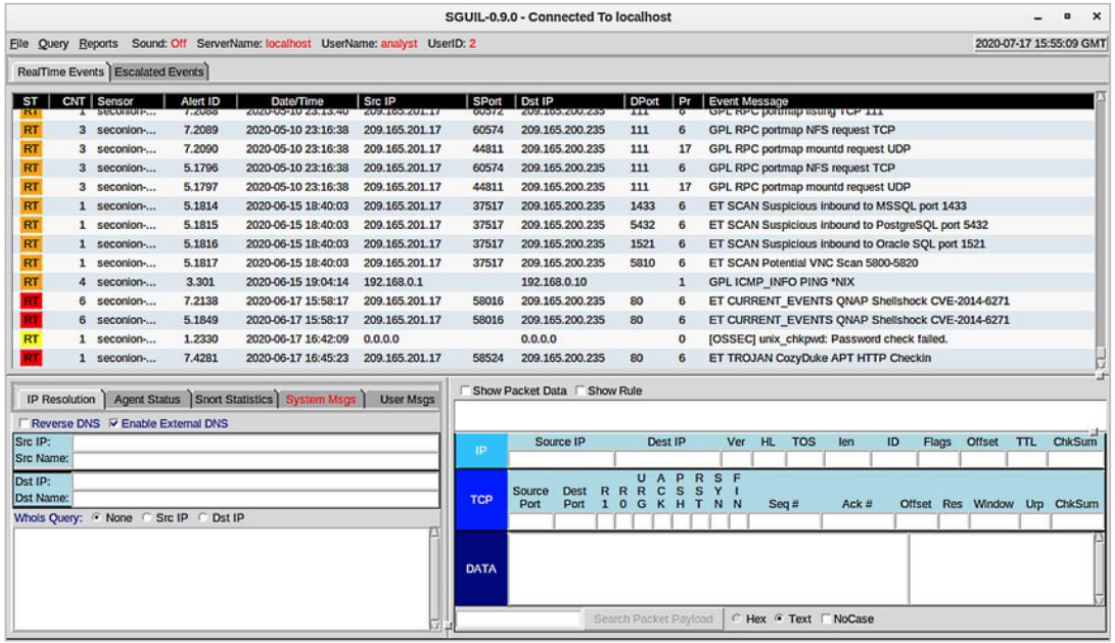
- **ST** - This is the status of the event. The event is color-coded by priority based on the category of the alert. There are four priority levels: very low, low, medium, and high and the colors range from light yellow to red as the priority increases.
- **CNT** - This is the count for the number of times this event has been detected for the same source and destination IP address. The system has determined that this set of events is correlated.
- **Sensor** - This is the agent reporting the event. The available sensors and their identifying numbers can be found in the Agent Status tab of the pane which appears below the events window on the left.



Sguil Window

Alert Generation (Contd.)

- **Alert ID** - This two-part number represents the sensor that has reported the problem and the event number for that sensor.
- **Date/Time** - This is the timestamp for the event. In the case of correlated events, it is the timestamp for the first event.
- **Event Message** - This is the identifying text for the event. his is configured in the rule that triggered the alert. The associated rule can be viewed in the right-hand pane, just above the packet data. To display the rule, the **Show Rule** checkbox must be selected.



Sguil Window

Rules and Alerts

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

- The information found in the alerts that are displayed in Sguil will differ in message format because they come from different sources.
- The Sguil alert in the figure was triggered by a rule that was configured in Snort.

Snort Rule Structure

Snort rules consist of two sections, as shown in the figure: the rule header and the rule options. Rule Location is sometimes added by Sguil.

```
alert ip any any -> any any (msg:"GPL ATTACK_RESPONSE id check returned root";
content:"uid=0|28|root|29|"; fast_pattern:only; classtype:bad-unknown; sid:2100498;
rev:8;)
/nsm/server_data/securityonion/rules/seconion-eth1-1/downloaded.rules:Line 692
```

Component	Example (shortened…)	Explanation
rule header	alert ip any any -> any any	Contains the action to be taken, source and destination addresses and port, and the direction of traffic flow
rule options	(msg:”GPL ATTACK_RESPONSE ID CHECK RETURNED ROOT”;…)	Includes the message to be displayed, details of packet content, alert type, source ID, and additional details, such as a reference for the rule or vulnerability
rule location	/nsm/server_data/securityonion/rules/…	Added by Sguil to indicate the location of the rule in the Security Onion file structure and in the specified rule file

Snort Rule Structure (Contd.)

The Rule Header

The rule header contains the action, protocol, addressing, and port information, as shown in the figure. The structure of the header portion is consistent between Snort alert rule. Snort can be configured to use variables to represent internal and external IP addresses.

```
alert ip any any -> any any (msg:"GPL ATTACK_RESPONSE id check returned root";
content:"uid=0|28|root|29|"; fast_pattern:only; classtype:bad-unknown; sid:2100498;
rev:8;)
/nsm/server_data/securityonion/rules/seconion-eth1-1/downloaded.rules:Line 692
```

Component	Explanation
alert	the action to be taken is to issue an alert, other actions are log and pass
ip	the protocol
any any	the specified source is any IP address and any Layer 4 port
->	the direction of flow is from the source to the destination
any any	the specified destination is any IP address and any Layer 4 port

Snort Rule Structure (Contd.)

The Rule Options

- The structure of the options section of the rule is variable. It is the portion of the rule that is enclosed in parenthesis, as shown in the figure. It contains the text message that identifies the alert. It also contains metadata about the alert, such as a URL.
- Snort rule messages may include the source of the rule. Three common sources for Snort rules are:
  - **GPL** - Older Snort rules that were created by Sourcefire and distributed under a GPLv2. The GPL ruleset is not Cisco Talos certified. The GPL ruleset is can be downloaded from the Snort website, and it is included in Security Onion.
  - **ET** - Snort rules from Emerging Threats which is a collection point for Snort rules from multiple sources. The ET ruleset contains rules from multiple categories. A set of ET rules is included with Security Onion. Emerging Threats is a division of Proofpoint, Inc.
  - **VRT** - These rules are immediately available to subscribers and are released to registered users 30 days after they were created, with some limitations. They are now created and maintained by Cisco Talos.

Snort Rule Structure (Contd.)

```
alert ip any any -> any any (msg:"GPL ATTACK_RESPONSE id check returned root";
content:"uid=0|28|root|29|"; fast_pattern:only; classtype:bad-unknown; sid:2100498;
rev:8;)
/nsm/server_data/securityonion/rules/seconion-eth1-1/downloaded.rules:Line 692
```

Component	Explanation
msg:	Text that describes the alert.
content:	Refers to content of the packet. In this case, an alert will be sent if the literal text “uid=0(root)” appears anywhere in the packet data. Values specifying the location of the text can be provided.
reference:	This is not shown in the figure. It is often a link to a URL that provides more information on the rule. In this case, the sid is hyperlinked to the source of the rule on the internet.
classtype:	A category for the attack. Snort includes a set of default categories that have one of four priority values.
sid:	A unique numeric identifier for the rule.
rev:	The revision of the rule that is represented by the sid.

Lab - Snort and Firewall Rules

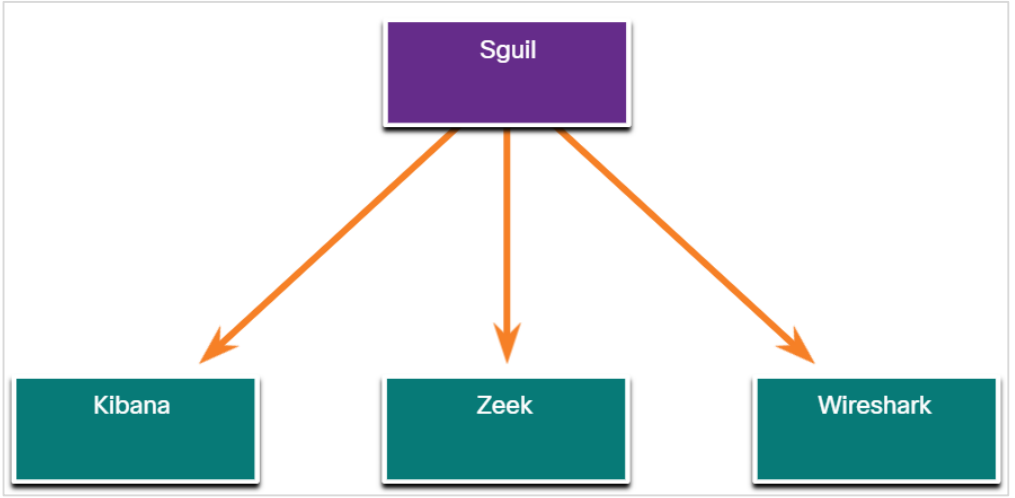
In this lab, you will complete the following objectives:

- Perform live monitoring of IDS and events.
- Configure your own customized firewall rule to stop internal hosts from contacting a malware-hosting server.
- Craft a malicious packet and launch it against an internal target.
- Create a customized IDS rule to detect the customized attack and issue an alert based on it.

26.2 Overview of Alert Evaluation

The Need for Alert Evaluation

- The threat landscape is constantly changing as new vulnerabilities and threats are discovered. As user and organizational needs change, so also does the attack surface.
- Threat actors have learned how to quickly vary features of their exploits in order to evade detection.
- It is better to have alerts that are sometimes generated by innocent traffic, than it is to have rules that miss malicious traffic.
- It is necessary to have skilled cybersecurity analysts investigate alerts to determine if an exploit has actually occurred.
- Tier 1 cybersecurity analysts will work through queues of alerts in a tool like Sguil, pivoting to tools like Zeek, Wireshark, and Kibana to verify that an alert represents an actual exploit.



Primary Tools for the Tier 1 Cybersecurity Analyst

Evaluating Alerts

- Security incidents are classified using a scheme borrowed from medical diagnostics. This classification scheme is used to guide actions and to evaluate diagnostic procedures. The concern is that either diagnosis can be accurate, or true, or inaccurate, or false.
- In network security analysis, the cybersecurity analyst is presented with an alert. The cybersecurity analyst needs to determine if this diagnosis is true.
- Alerts can be classified as follows:
  - **True Positive**: The alert has been verified to be an actual security incident.
  - **False Positive**: The alert does not indicate an actual security incident. Benign activity that results in a false positive is sometimes referred to as a benign trigger.
- An alternative situation is that an alert was not generated. The absence of an alert can be classified as:
  - **True Negative**: No security incident has occurred. The activity is benign.
  - **False Negative**: An undetected incident has occurred.

Evaluating Alerts (Contd.)

When an alert is issued, it will receive one of four possible classifications:

	True	False
Positive (Alert exists)	Incident occurred	No incident occurred
Negative (No alert exists)	No incident occurred	Incident occurred

- **True positives** are the desired type of alert. They mean that the rules that generate alerts have worked correctly.
- **False positives** are not desirable. Although they do not indicate that an undetected exploit has occurred, they are costly because cybersecurity analysts must investigate false alarms.
- **True negatives** are desirable. They indicate that benign normal traffic is correctly ignored, and erroneous alerts are not being issued.
- **False negatives** are dangerous. They indicate that exploits are not being detected by the security systems that are in place.

Note: *"True" events are desirable. "False" events are undesirable and potentially dangerous.*

Evaluating Alerts (Contd.)

- Benign events are those that should not trigger alerts. Excess benign events indicate that some rules or other detectors need to be improved or eliminated.
- When true positives are suspected, a cybersecurity analyst is required to escalate the alert to a higher level for investigation. The investigator will move forward with the investigation in order to confirm the incident and identify any potential damage that may have been caused.
- A cybersecurity analyst may also be responsible for informing security personnel that false positives are occurring to the extent that the cybersecurity analyst's time is seriously impacted.
- False negatives may be discovered well after an exploit has occurred. This can happen through retrospective security analysis (RSA). RSA can occur when newly obtained rules or other threat intelligence is applied to archived network security data.
- For this reason, it is important to monitor threat intelligence to learn of new vulnerabilities and exploits and to evaluate the likelihood that the network was vulnerable to them at some time in the past.

Deterministic Analysis and Probabilistic Analysis

- Deterministic analysis evaluates risk based on what is known about a vulnerability. This type of risk analysis can only describe the worst case.
- Probabilistic analysis estimates the potential success of an exploit by estimating the likelihood that if one step in an exploit has successfully been completed that the next step will also be successful.
- In a deterministic analysis, all of the information to accomplish an exploit is assumed to be known.
- In probabilistic analysis, it is assumed that the port numbers that will be used can only be predicted with some degree of confidence.
- The two approaches are summarized below.
  - **Deterministic Analysis** - For an exploit to be successful, all prior steps in the exploit must also be successful. The cybersecurity analyst knows the steps for a successful exploit.
  - **Probabilistic Analysis** - Statistical techniques are used to determine the probability that a successful exploit will occur based on the likelihood that each step in the exploit will succeed.

26.3 Evaluating Alerts Summary

What Did I Learn in this Module?

- Security Onion is an open-source suite of Network Security Monitoring (NSM) tools that run on an Ubuntu Linux distribution.
- Security Onion tools provide three core functions for the cybersecurity analyst: full packet capture and data types, network-based and host-based intrusion detection systems, and alert analyst tools.
- Security Onion integrates the data and IDS logs into a single platform through the following tools:
  - Sguil - serves as a starting point in the investigation of security alerts.
  - Kibana - It is an interactive dashboard interface to Elasticsearch data.
  - The Wireshark packet capture application is integrated into the Security Onion suite.
  - Zeek is a network traffic analyzer that serves as a security monitor.

What Did I Learn in this Module? (Contd.)

- Snort is a Network Intrusion Detection System (NIDS). It is an important source of the alert data that is indexed in the Sguil analysis tool.
- Alerts can be classified as True Positive (The alert has been verified to be an actual security incident) or False Positive (The alert does not indicate an actual security incident).
- An alternative situation is that an alert was not generated. The absence of an alert can be classified as: True Negative (No security incident has occurred. The activity is benign.) and False Negative (An undetected incident has occurred).
- Deterministic analysis evaluates risk based on what is known about a vulnerability.
- Probabilistic analysis estimates the potential success of an exploit by estimating the likelihood that if one step in an exploit has successfully been completed that the next step will also be successful.


CISCO