Chapter 10: Intrusion Data Analysis

Information Security

Dr. Ayman Aljarbouh

10.1 Evaluating Alerts

Module Objectives

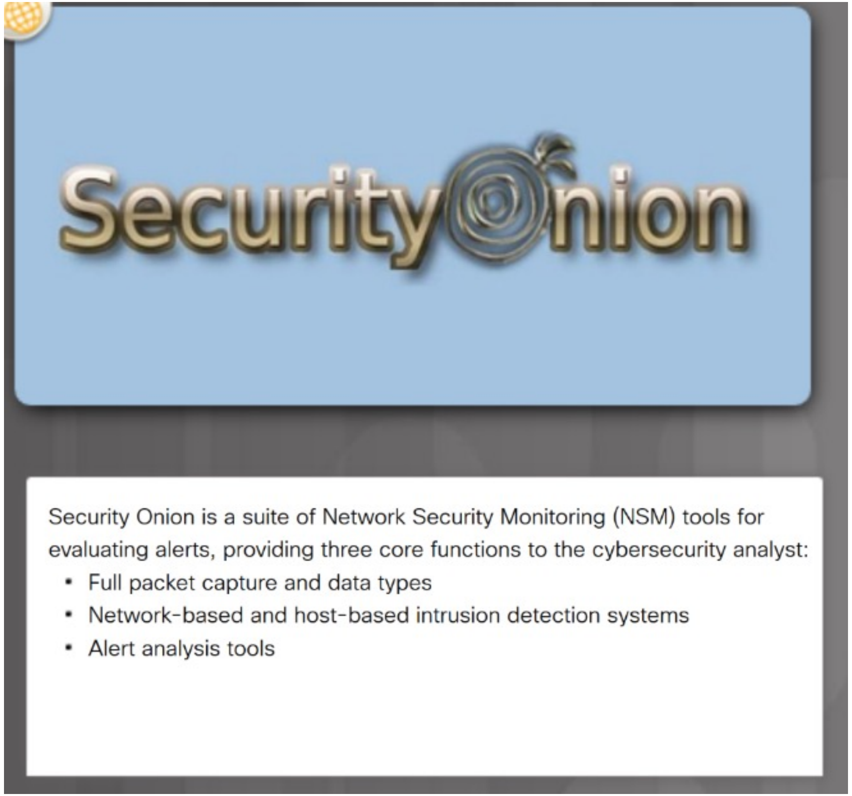
Module Title: Evaluating Alerts

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.

Security Onion

- Security Onion is an open-source suite of Network Security Monitoring (NSM) tools that run on an Ubuntu Linux distribution.
- 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.



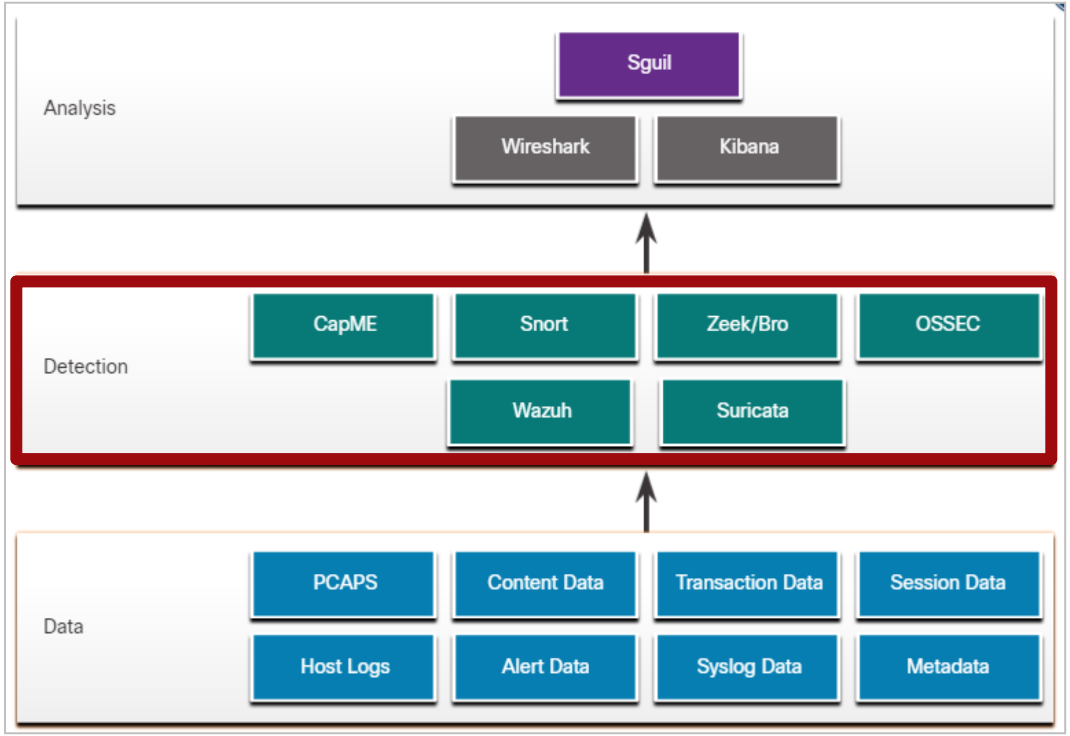
Security Onion is a suite of Network Security Monitoring (NSM) tools for evaluating alerts, providing three core functions to the cybersecurity analyst:

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

Sources of Alerts

Detection Tools for 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.

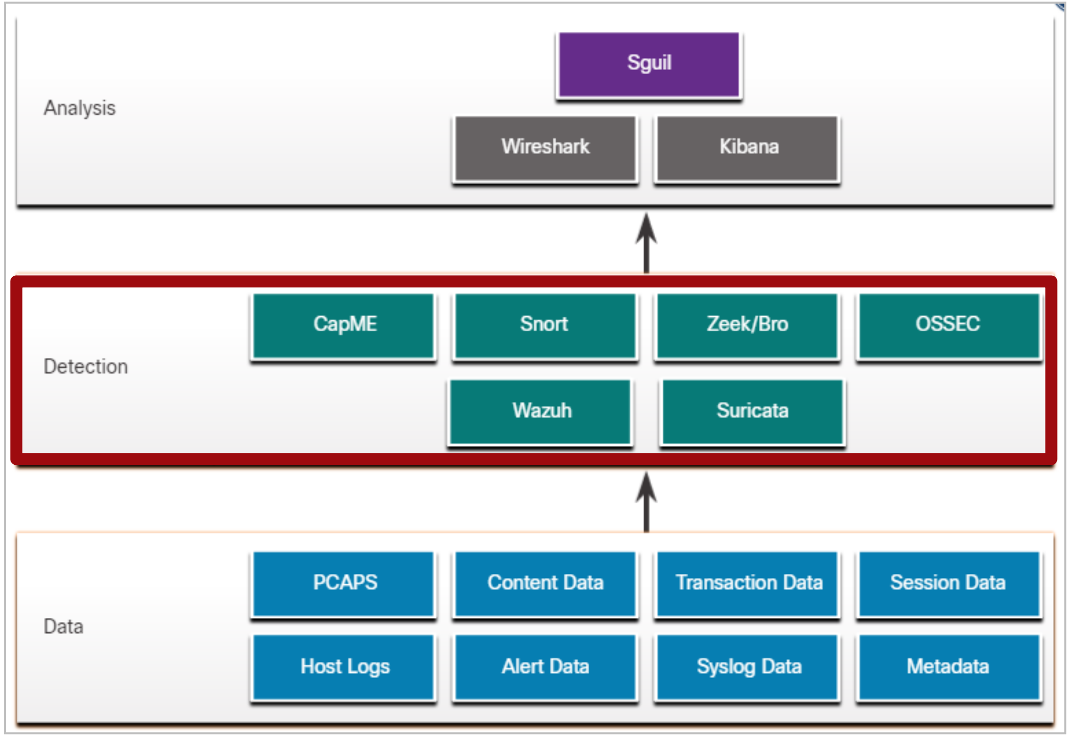


A Security Onion Architecture

Sources of Alerts

Detection Tools for Collecting Alert Data (Contd.)

- **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** – This is a NIDS that uses more of a behavior-based approach to intrusion detection.

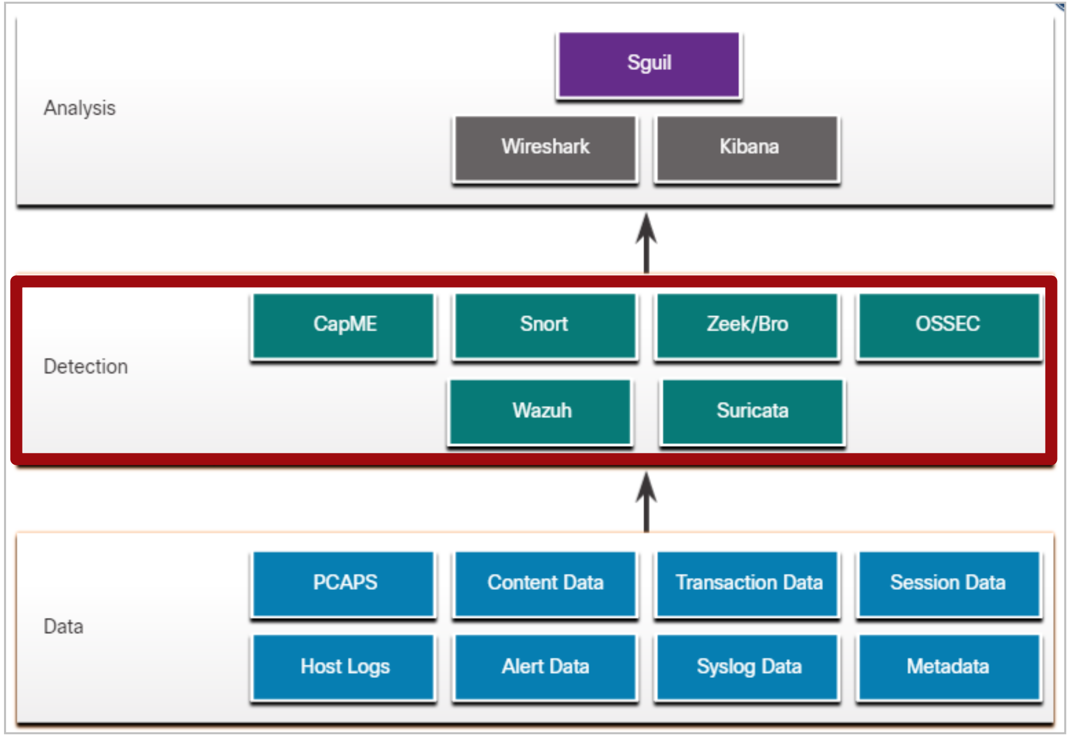


A Security Onion Architecture

Sources of Alerts

Detection Tools for Collecting Alert Data (Contd.)

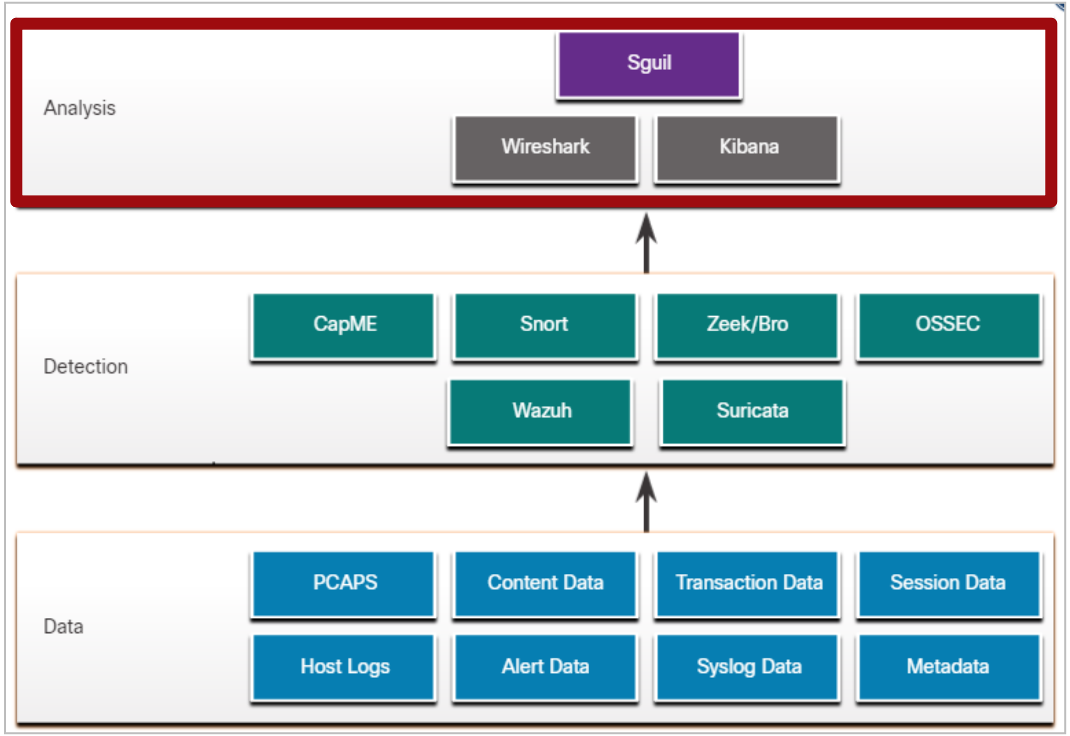
- **OSSEC** – This is a host-based intrusion detection system (HIDS) that is integrated into Security Onion.
- **Wazuh** – It provides host logfile analysis, file integrity monitoring, vulnerability detection, configuration assessment, and incident response.
- **Suricata** – This is a NIDS that uses a signature-based approach. It can also be used for inline intrusion prevention.



A Security Onion Architecture

Analysis Tools

- **Sguil** – This provides a high-level cybersecurity analysts' console for investigating security alerts from a wide variety of sources.
- **Kibana** – It is an interactive dashboard interface to Elasticsearch data. It allows querying of NSM data and provides flexible visualizations of that data.
- **Wireshark** – This is a packet capture application that is integrated into the Security Onion suite.



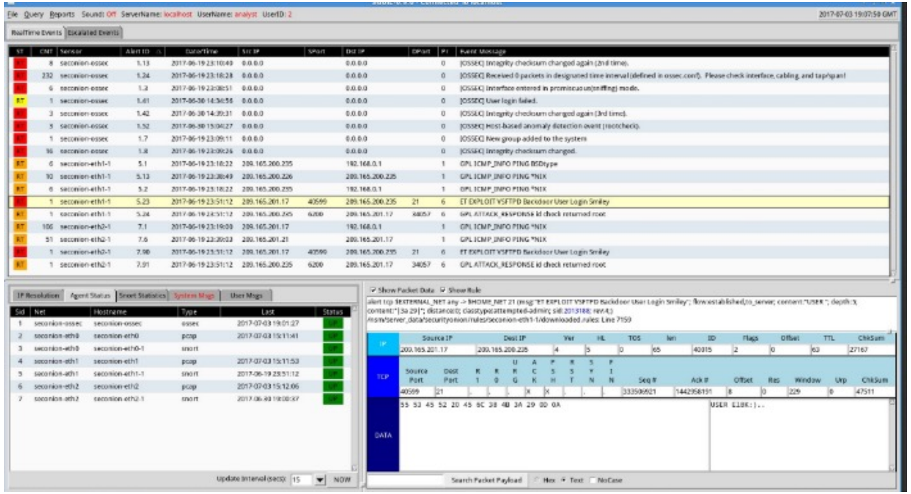
A Security Onion Architecture

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 23:18:28	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 23:39:03	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	40315	2	0	63	27167

TCP	Source Port	Dest Port	R 1	R 0	U R G	A C K	P S H	R S T	S Y N	F I N	Seq #	Ack #	Offset	Res	Window	Urp	ChkSum
	40599	21	.	.	.	X	X	.	.	.	333506921	1442958191	8	0	229	0	47511

DATA

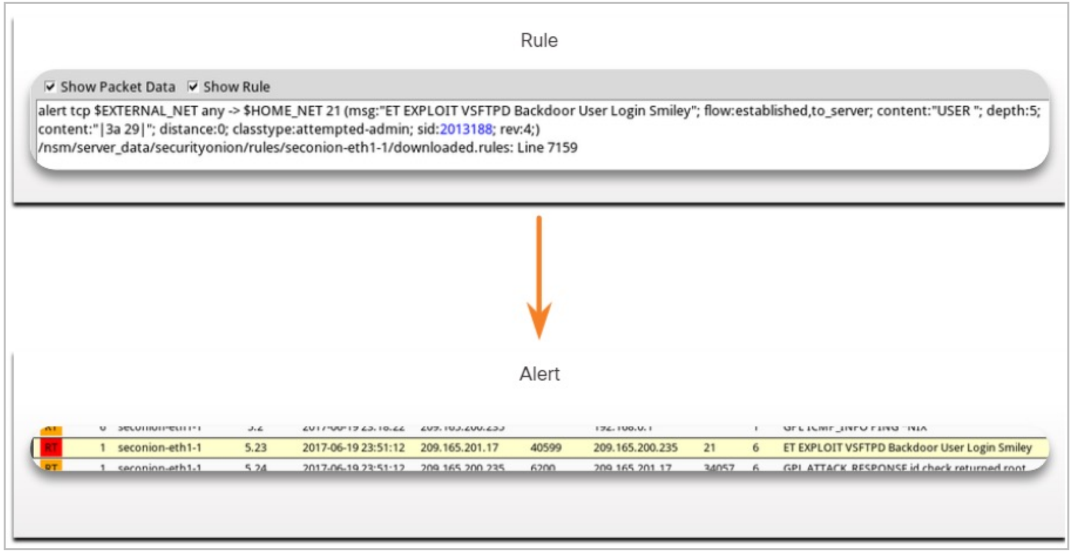
```
55 53 45 52 20 45 6C 38 4B 3A 29 0D 0A          USER El8K:)..
```

Search Packet Payload ○ Hex ● Text □ NoCase

Sources of Alerts

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



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
```

Alert

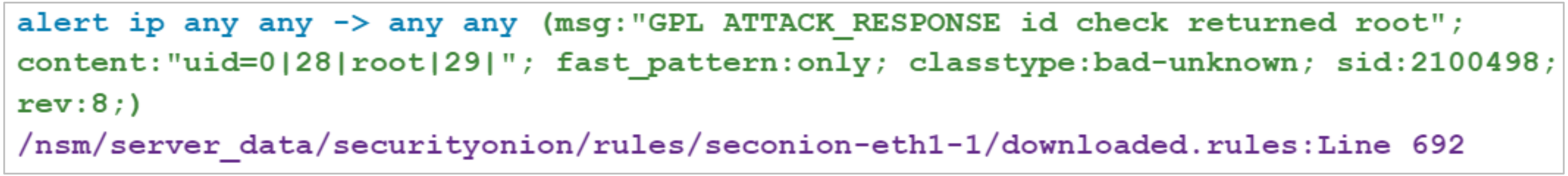
```
RT  0  seconion-eth1-1  5.2   2017-06-19 23:18:22  209.165.200.235         192.168.0.1            1  GPL ICMP_INFO PING *NIX
RT  1  seconion-eth1-1  5.23  2017-06-19 23:51:12  209.165.201.17   40599  209.165.200.235  21    6  ET EXPLOIT VSFTPD Backdoor User Login Smiley
RT  1  seconion-eth1-1  5.24  2017-06-19 23:51:12  209.165.200.235  6200   209.165.201.17   34057 6  GPL ATTACK_RESPONSE id check returned root
```

- 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.

Sources of Alerts

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
```

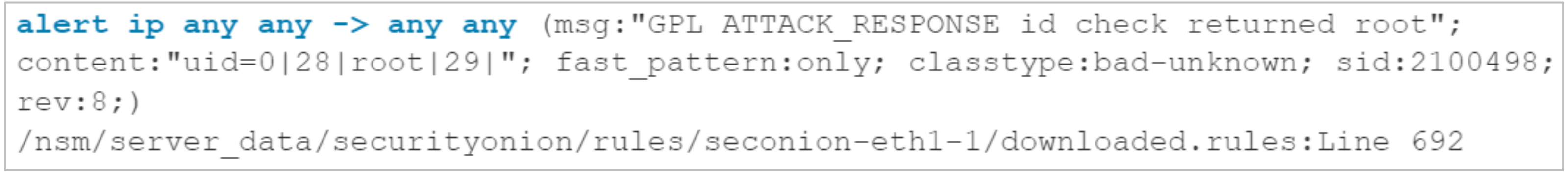
Component	Example (shortened…)	Explanation
rule header	alert ip any any -> any any	Contains the action to be taken, source and destination addresses and port, and the direction of traffic flow
rule options	(msg:”GPL ATTACK_RESPONSE ID CHECK RETURNED ROOT”;…)	Includes the message to be displayed, details 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

Sources of Alerts

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

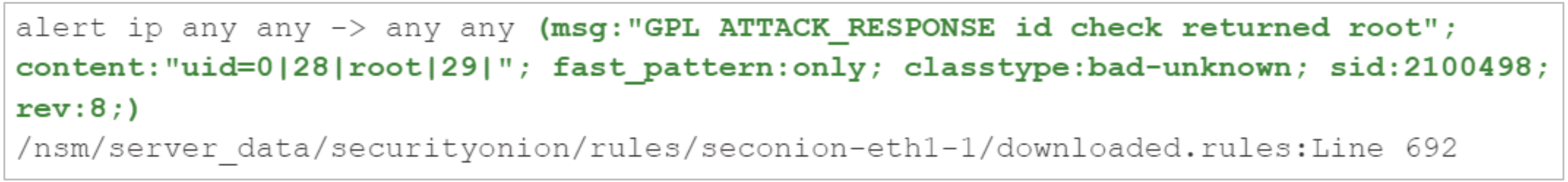
Snort Rule Structure (Contd.)

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.

Sources of Alerts

Snort Rule Structure (Contd.)

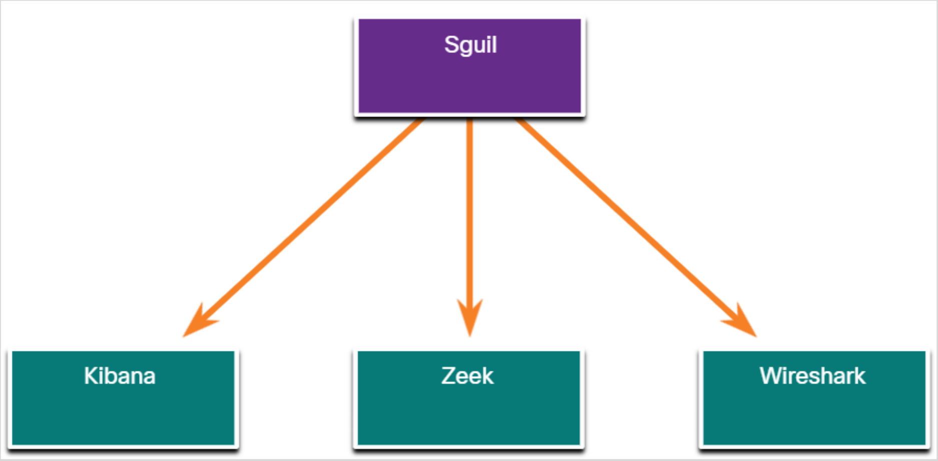


```
alert ip any any -> any any (msg:"GPL ATTACK_RESPONSE id check returned root";
content:"uid=0|28|root|29|"; fast_pattern:only; classtype:bad-unknown; sid:2100498;
rev:8;)
/nsm/server_data/securityonion/rules/seconion-eth1-1/downloaded.rules:Line 692
```

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.

Overview of Alert Evaluation

The Need for Alert Evaluation



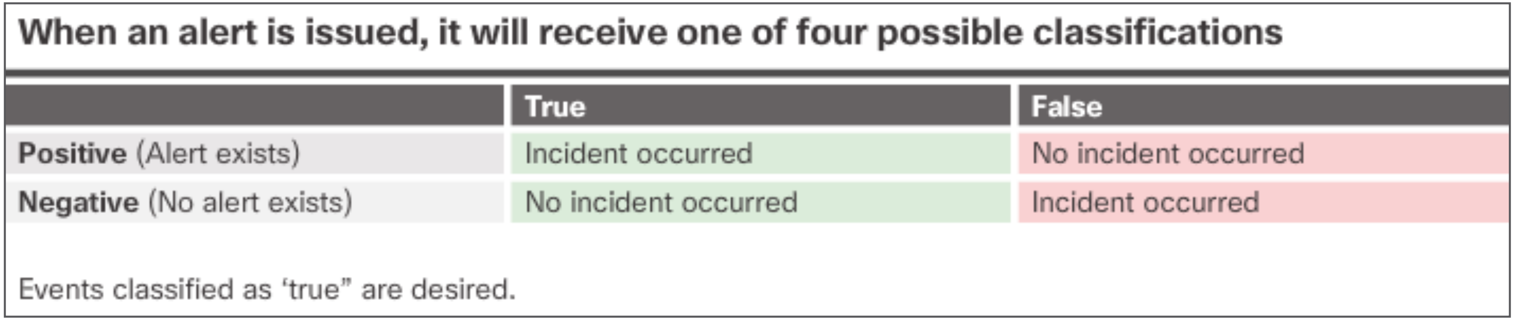
Primary Tools for the Tier 1 Cybersecurity Analyst

- Exploits will inevitably evade protection measures, no matter how sophisticated they may be.
- Detection rules should be overly conservative.
- 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.

Overview of Alert Evaluation

Evaluating Alerts

- 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.
  - **True Negative**: No security incident has occurred.
  - **False Negative**: An undetected incident has occurred.



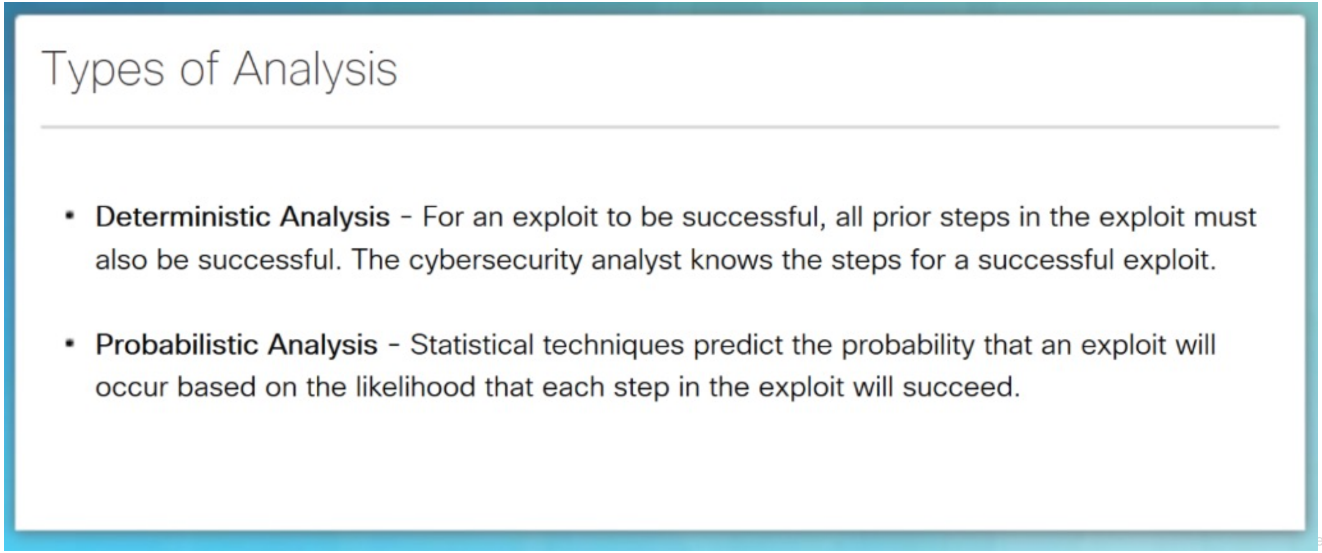
When an alert is issued, it will receive one of four possible classifications

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

Events classified as ‘true” are desired.

Deterministic Analysis and Probabilistic Analysis

- Statistical techniques can be used to evaluate the risk that exploits will be successful in a given network.
  - **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.



Types of Analysis

- **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 predict the probability that an exploit will occur based on the likelihood that each step in the exploit will succeed.

Intrusion Data Analysis

New Terms and Commands

- Security Onion
- Network Security Monitoring (NSM)
- CapME
- Snort
- Zeek
- OSSEC
- Wazuh
- Suricata
- Passive Asset Detection System (PADS)
- Retrospective Security Analysis (RSA)
- Deterministic Analysis
- Probabilistic Analysis
- Sguil
- Kibana
- Wireshark

Intrusion Data Analysis

Lab 34 - 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.