

Information Security Operations Center

Prepared by
Dr. Ibrahim Gomaa

جامعة الجميع الذكية

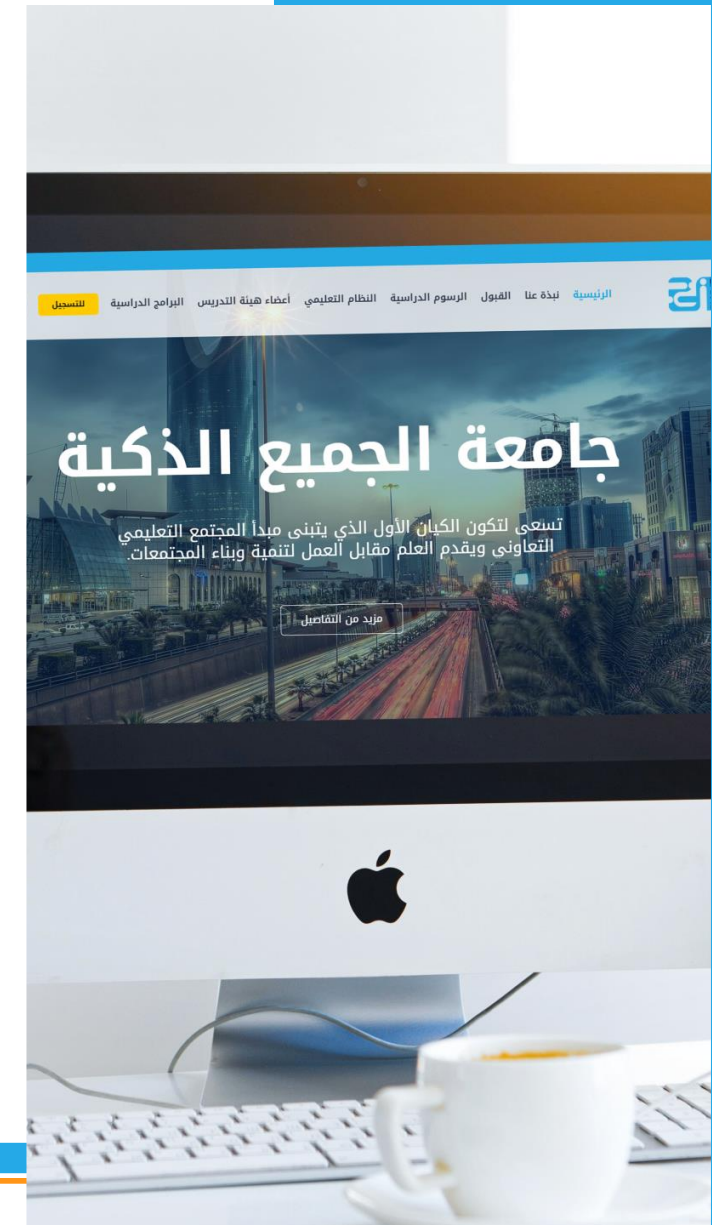
تسعى لتكون الكيان الأول الذي يتبنى مبدأ المجتمع التعليمي
التعاوني ويقدم العلم مقابل العمل لتنمية وبناء المجتمعات.

Lecture 8: Evaluate network security alerts

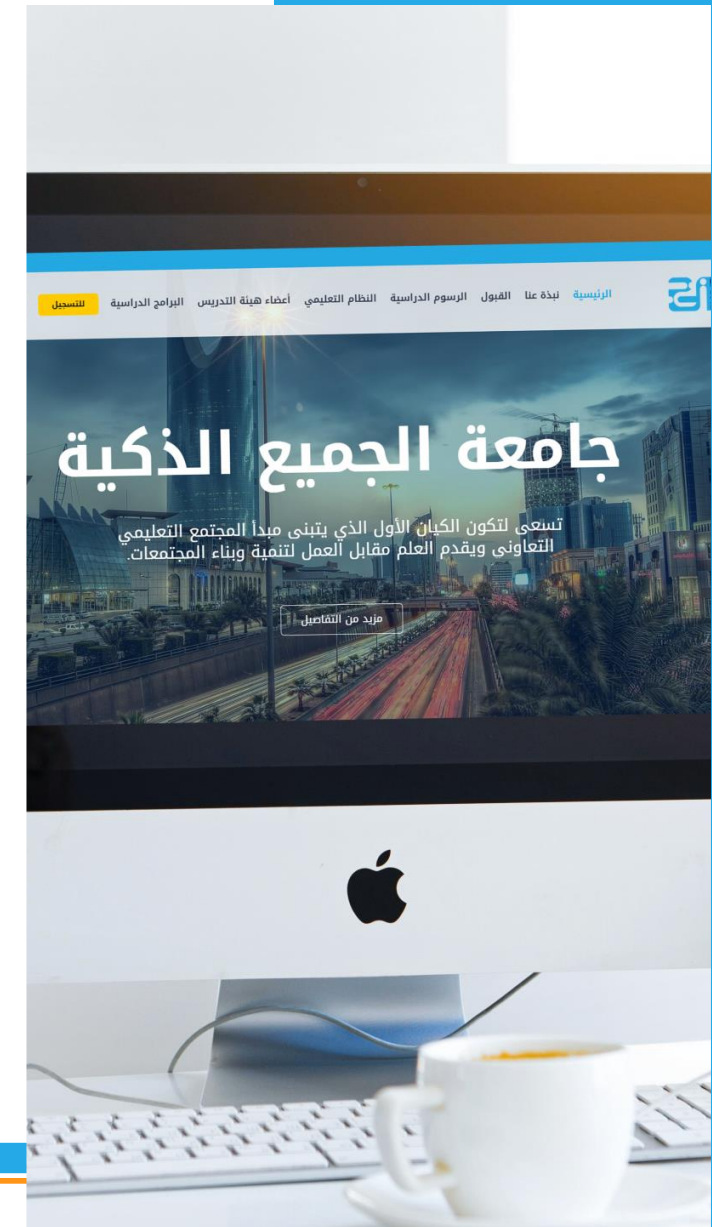


جامعة الجميع الذكية
EVERYONE'S SMART UNIVERSITY

Topic Title	Topic Objective
Types of Security Data	Describe the types of data used in security monitoring.
Source of Alerts	Identify the structure of alerts.
Overview of Alert Evaluation	Explain how alerts are classified.



Types of Security Data



- **Alert data** consists of messages generated by intrusion prevention systems (IPSs) or intrusion detection systems (IDSs) in response to traffic that violates a rule or matches the signature of a known exploit.
- A network IDS (NIDS), such as Snort, comes configured with rules for known exploits.
- Alerts are generated by Snort and are made readable and searchable by the Sguil and Squert applications, which are part of the Security Onion suite of NSM tools.

SGUIL-0.9.0 - Connected To localhost

File Query Reports Sound: Off ServerName: localhost Username: analyst UserID: 2 2020-06-03 14:58:25 GMT

RealTime Events Escalated Events

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	1	seconion...	5.1482	2020-05-10 21:20:55	209.165.201.17	52332	209.165.200.235	80	6	ET WEB_SPECIFIC_APPS phpSkelSite theme parameter remote file inclusi
RT	1	seconion...	7.1795	2020-05-10 21:20:55	209.165.201.17	52332	209.165.200.235	80	6	ET WEB_SPECIFIC_APPS phpSkelSite theme parameter remote file inclusi
RT	1	seconion...	7.1688	2020-05-10 21:20:52	209.165.201.17	52298	209.165.200.235	80	6	ET WEB_SPECIFIC_APPS phptraverse mp3_id.php GLOBALS Parameter F
RT	1	seconion...	5.1375	2020-05-10 21:20:52	209.165.201.17	52298	209.165.200.235	80	6	ET WEB_SPECIFIC_APPS phptraverse mp3_id.php GLOBALS Parameter F
RT	1	seconion...	5.1580	2020-05-10 21:21:17	209.165.201.17	52414	209.165.200.235	80	6	ET WORM TheMoon.linksys.router 1
RT	1	seconion...	7.1893	2020-05-10 21:21:17	209.165.201.17	52414	209.165.200.235	80	6	ET WORM TheMoon.linksys.router 1
RT	4	seconion...	5.362	2020-05-10 20:58:01	209.165.200.235	6200	209.165.201.17	37071	6	GPL ATTACK_RESPONSE id check returned root
RT	4	seconion...	7.675	2020-05-10 20:58:01	209.165.200.235	6200	209.165.201.17	37071	6	GPL ATTACK_RESPONSE id check returned root
RT	12	seconion...	7.690	2020-05-10 21:20:25	209.165.201.17	52158	209.165.200.235	80	6	GPL EXPLOIT .crf access
RT	12	seconion...	5.377	2020-05-10 21:20:25	209.165.201.17	52158	209.165.200.235	80	6	GPL EXPLOIT .crf access
RT	8	seconion...	7.683	2020-05-10 21:20:25	209.165.201.17	52156	209.165.200.235	80	6	GPL EXPLOIT .htr access
RT	8	seconion...	5.370	2020-05-10 21:20:25	209.165.201.17	52156	209.165.200.235	80	6	GPL EXPLOIT .htr access
RT	1	seconion...	5.1055	2020-05-10 21:20:49	209.165.201.17	52238	209.165.200.235	80	6	GPL EXPLOIT /isadmpwd/aexp2.htr access
RT	1	seconion...	7.1368	2020-05-10 21:20:49	209.165.201.17	52238	209.165.200.235	80	6	GPL EXPLOIT /isadmpwd/aexp2.htr access

IP Resolution Agent Status Snort Statistics System Msgs User Msgs

☐ Reverse DNS ☒ Enable External DNS

Src IP:
 Src Name:
 Dst IP:
 Dst Name:

Whois Query: ☐ None ☐ Src IP ☐ Dst IP

☒ Show Packet Data ☒ Show Rule

alert ip any any -> any any (msg:"GPL ATTACK_RESPONSE id check returned root"; content:"uid=0(root)[29]"; fast_pattern_only; classtype:bad-unknown; sid:2100498; rev:8; metadata:created_at 2010_09_23, updated_at 2010_09_23;) /usr/share/snort/rules/seconion-ens192-1/downloaded.rules: Line 700

IP	Source IP	Dest IP	Ver	HL	TOS	len	ID	Flags	Offset	TTL	ChkSum
TCP	209.165.200.235	209.165.201.17	4	5	0	76	13818	2	0	64	53097
	Source Port	Dest Port	U	A	P	R	S	F			
	6200	37071	1	0	G	K	H	T	N	N	
			Seq#	Ack#	Offset	Res	Window	Up	ChkSum		
			2269574098	3537747796	8	0	181	0	10442		
DATA	75 69 64 3D 30 28 72 6F 6F 74 29 20 67 69 64 3D	uid=0(root) gid=0(root).									
	30 28 72 6F 6F 74 29 6A										

Search Packet Payload ☐ Hex ☒ Text ☐ NoCase

Sguil Console Showing Test Alert from Snort IDS

- **Session data** is a record of a conversation between two network endpoints.
- It includes **the five tuples** of source and destination IP addresses, source and destination port numbers, and the IP code for the protocol in use.
- Data about the session includes a session ID, the amount of data transferred by source and destination and information related to the duration of the session.
- The figure shows a partial output for three HTTP sessions from a Zeek connection log.

1	2	3	4	5	6	7	8	9	10	11	12	13
ts	uid	id.orig_h	id.orig_p	id.resp_h	id.resp_p	proto	service	duration	orig_bytes	resp_bytes	orig_pkts	resp_pkts
1320279567	CEv1254N5gT3PwJLog	192.168.2.76	52034	174.129.249.33	80	tcp	http	0.082899	389	1495	5	4
1320279567	Ci8Ueb3SkSJHwASNW4	192.168.2.76	52035	184.72.234.3	80	tcp	http	2.56194	905	731	9	8
1320279567	CaTMSv1Sb8HIFunqj	192.168.2.76	52033	184.72.234.3	80	tcp	http	3.345539	1856	1445	15	13

1. **ts**: session start timestamp
2. **uid**: unique session ID
3. **id.orig_h**: IP address of host that originated the session (source address)
4. **id.orig_p**: protocol port for the originating host (source port)
5. **id.resp_h**: IP address of host responding to the originating host (destination address)
6. **id.resp_p**: protocol of responding host (destination port)
7. **proto**: transport layer protocol for session
8. **service**: application layer protocol
9. **duration**: duration of the session
10. **orig_bytes**: bytes from originating host
11. **resp_bytes**: bytes from responding host
12. **orig_packets**: packets from the originating host
13. **resp_packets**: packets from responding host

- **Transaction data** consists of the messages that are exchanged during network sessions.
- These transactions can be viewed in packet capture transcripts.
- The transactions that represent the requests and replies would be logged in an access log on a server or by a NIDS like Zeek.
- A session might include the downloading of content from a webserver, as shown in the figure.



Full Packet Captures

- Full packet captures are the most detailed network data that is generally collected.
- It contains the actual content of the conversations such as text of email messages, the HTML in web pages, and the files that enter or leave the network.
- Extracted content can be recovered from full packet captures and analyzed for malware or user behavior that violates business and security policies.
- The figure here shows the interface for the **Network Analysis Monitor** component of **Cisco Prime** Infrastructure system, which can display full packet captures.

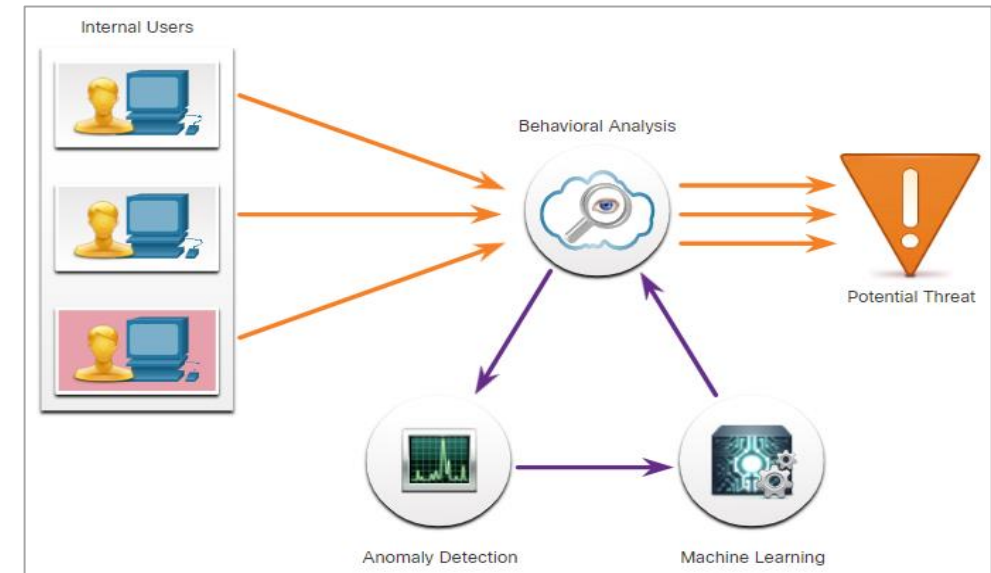
No.	Time	Source	Destination	Protocol	Length	Info
38333	2.691104	1.3.2.178	1.2.0.2	TCP	70	[TCP Dup ACK 34839#1] [TCP ACKed unseen segment] 54735 > http [ACK]
38334	2.691167	1.4.0.1	1.1.2.27	HTTP	1504	[TCP Previous segment not captured] Continuation or non-HTTP traffic[Packe...
38335	2.691175	1.4.0.1	1.1.2.27	HTTP	1504	[TCP Previous segment not captured] Continuation or non-HTTP traffic[Packe...
38336	2.691189	1.4.0.1	1.1.2.27	HTTP	1504	[TCP Previous segment not captured] Continuation or non-HTTP traffic[Packe...
38337	2.691193	1.4.0.1	1.1.2.27	HTTP	1504	[TCP Previous segment not captured] Continuation or non-HTTP traffic[Packe...
38338	2.691214	1.4.0.1	1.1.2.27	HTTP	1504	[TCP Previous segment not captured] Continuation or non-HTTP traffic[Packe...
38339	2.691221	1.4.0.1	1.1.2.27	HTTP	1504	[TCP Previous segment not captured] Continuation or non-HTTP traffic[Packe...

Frame 1: 1504 bytes on wire (12032 bits), 1500 bytes captured (12000 bits)		
Ethernet II, Src: 02:1a:c5:01:00:00 (02:1a:c5:01:00:00), Dst: 02:1a:c5:02:00:00 (02:1a:c5:02:00:00)		
Internet Protocol Version 4, Src: 1.2.0.2 (1.2.0.2), Dst: 1.3.1.229 (1.3.1.229)		
Transmission Control Protocol, Src Port: http (80), Dst Port: 55998 (55998), Seq: 1, Ack: 1, Len: 1438		
Hypertext Transfer Protocol		
0000	02 1A C5 02 00 00 02 1A C5 01 00 00 08 00 45 00	E.
0010	05 DC 87 D5 40 00 20 06 C9 5B 01 02 00 02 01 03	...@. ..{.....
0020	01 E5 00 50 DA BE CF FD 2D 19 4F DA E7 D9 80 18	...P.....O.....
0030	1C 48 BE E1 00 00 01 01 08 0A AC 19 04 03 AB C7	..H.....
0040	79 16 37 BE 45 A5 2F B6 30 9C 7E 72 D7 50 D1 17	y.7.E./..~r.P..
0050	38 71 79 A9 68 D0 D0 B8 17 58 97 B8 42 C7 9E 55	;qy.k....X..B..U
0060	FF 2F 83 02 04 72 00 26 16 89 3C 21 68 B8 04 E0	./...r.&...!h...
0070	DD D4 DE 59 AB 69 AA A3 A0 BC D8 C9 61 B8 C4 CB	...Y.i.....a...
0080	FF 1E 7F 8B 5A DC B3 FB DC 55 93 D0 A9 79 83 35	...Z.....U...y.5

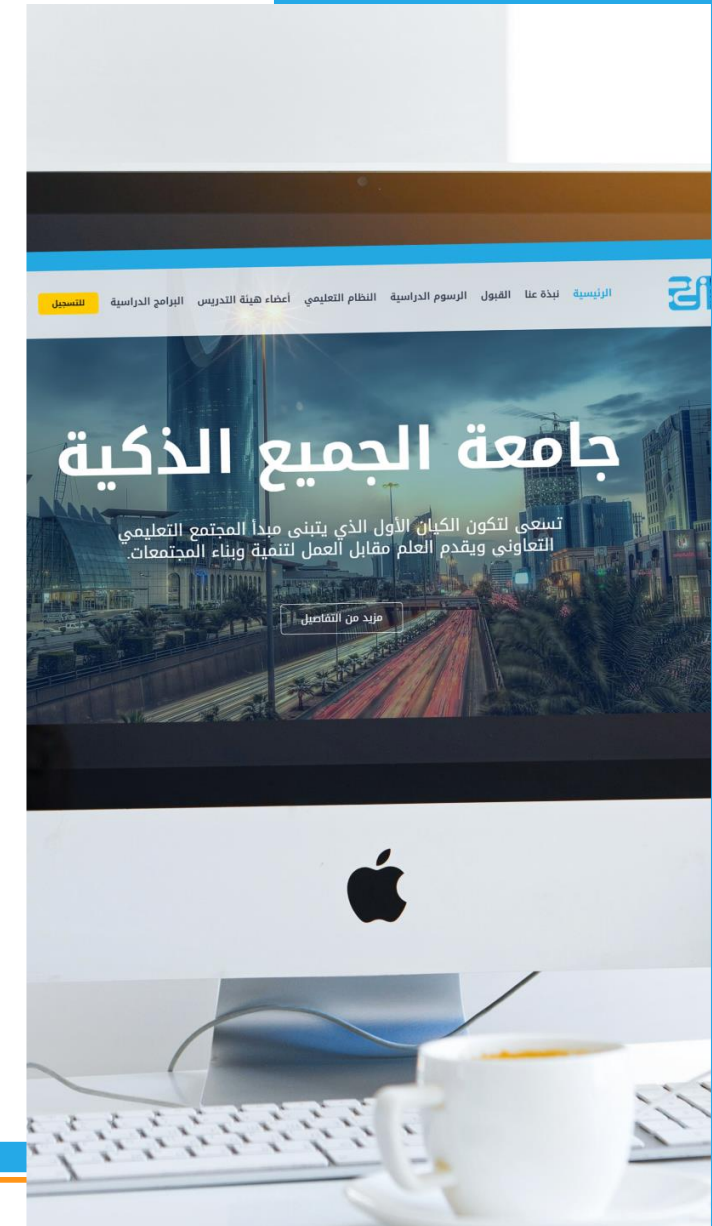


Full Packet Captures

- **Statistical data** is about network traffic which is created through the analysis of other forms of network data.
- Statistics can be used to characterize normal amounts of variation in network traffic patterns in order to identify network conditions that are significantly outside of those ranges.
- An example of an NSM tool that utilizes statistical analysis is **Cisco Cognitive Threat Analytics**.
- It can find malicious activity that has bypassed security controls or entered the network through unmonitored channels (including removable media) and is operating inside an organization's environment.



Sources of Alerts

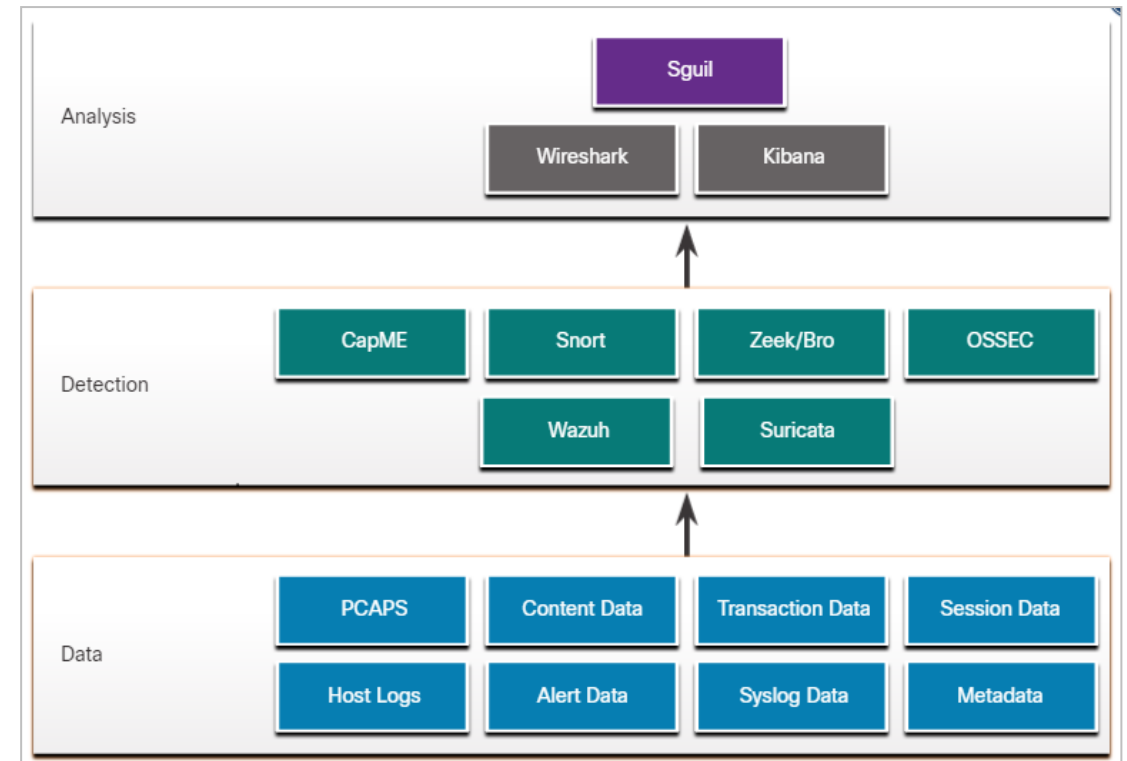


Security Onion

- **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 **Riverbed 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

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.

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	3	seconion...	7.2089	2020-05-10 23:16:38	209.165.201.17	60574	209.165.200.235	111	6	GPL RPC portmap request TCP 111
RT	3	seconion...	7.2090	2020-05-10 23:16:38	209.165.201.17	44811	209.165.200.235	111	17	GPL RPC portmap mountd request UDP
RT	3	seconion...	5.1796	2020-05-10 23:16:38	209.165.201.17	60574	209.165.200.235	111	6	GPL RPC portmap NFS request TCP
RT	3	seconion...	5.1797	2020-05-10 23:16:38	209.165.201.17	44811	209.165.200.235	111	17	GPL RPC portmap mountd request UDP
RT	1	seconion...	5.1814	2020-06-15 18:40:03	209.165.201.17	37517	209.165.200.235	1433	6	ET SCAN Suspicious Inbound to MSSQL port 1433
RT	1	seconion...	5.1815	2020-06-15 18:40:03	209.165.201.17	37517	209.165.200.235	5432	6	ET SCAN Suspicious Inbound to PostgreSQL port 5432
RT	1	seconion...	5.1816	2020-06-15 18:40:03	209.165.201.17	37517	209.165.200.235	1521	6	ET SCAN Suspicious Inbound to Oracle SQL port 1521
RT	1	seconion...	5.1817	2020-06-15 18:40:03	209.165.201.17	37517	209.165.200.235	5810	6	ET SCAN Potential VNC Scan 5800-5820
RT	4	seconion...	3.301	2020-06-15 19:04:14	192.168.0.1		192.168.0.10		1	GPL ICMP_INFO PING *NIX
RT	6	seconion...	7.2138	2020-06-17 15:58:17	209.165.201.17	58016	209.165.200.235	80	6	ET CURRENT_EVENTS QNAP Shellshock CVE-2014-6271
RT	6	seconion...	5.1849	2020-06-17 15:58:17	209.165.201.17	58016	209.165.200.235	80	6	ET CURRENT_EVENTS QNAP Shellshock CVE-2014-6271
RT	1	seconion...	1.2330	2020-06-17 16:42:09	0.0.0.0		0.0.0.0		0	[OSSEC] unix_chkpwd: Password check failed.
RT	1	seconion...	7.4281	2020-06-17 16:45:23	209.165.201.17	58524	209.165.200.235	80	6	ET TROJAN CozyDuke APT HTTP Checkin

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. This 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-0.9.0 - Connected To localhost

File Query Reports Sound: Off ServerName: localhost UserName: analyst UserID: 2 2020-07-17 15:55:09 GMT

RealTime Events Escalated Events

ST	CNT	Sensor	Alert ID	Date/Time	Src IP	SPort	Dst IP	DPort	Pr	Event Message
RT	1	seconion...	7.2089	2020-05-10 23:16:38	209.165.201.17	60574	209.165.200.235	111	6	GPL RPC portmap NFS request TCP 111
RT	3	seconion...	7.2089	2020-05-10 23:16:38	209.165.201.17	60574	209.165.200.235	111	6	GPL RPC portmap NFS request TCP
RT	3	seconion...	7.2090	2020-05-10 23:16:38	209.165.201.17	44811	209.165.200.235	111	17	GPL RPC portmap mountd request UDP
RT	3	seconion...	5.1796	2020-05-10 23:16:38	209.165.201.17	60574	209.165.200.235	111	6	GPL RPC portmap NFS request TCP
RT	3	seconion...	5.1797	2020-05-10 23:16:38	209.165.201.17	44811	209.165.200.235	111	17	GPL RPC portmap mountd request UDP
RT	1	seconion...	5.1814	2020-06-15 18:40:03	209.165.201.17	37517	209.165.200.235	1433	6	ET SCAN Suspicious inbound to MSSQL port 1433
RT	1	seconion...	5.1815	2020-06-15 18:40:03	209.165.201.17	37517	209.165.200.235	5432	6	ET SCAN Suspicious inbound to PostgreSQL port 5432
RT	1	seconion...	5.1816	2020-06-15 18:40:03	209.165.201.17	37517	209.165.200.235	1521	6	ET SCAN Suspicious inbound to Oracle SQL port 1521
RT	1	seconion...	5.1817	2020-06-15 18:40:03	209.165.201.17	37517	209.165.200.235	5810	6	ET SCAN Potential VNC Scan 5800-5820
RT	4	seconion...	3.301	2020-06-15 19:04:14	192.168.0.1		192.168.0.10		1	GPL ICMP_INFO PING *NIX
RT	6	seconion...	7.2138	2020-06-17 15:58:17	209.165.201.17	58016	209.165.200.235	80	6	ET CURRENT_EVENTS QNAP Shellshock CVE-2014-6271
RT	6	seconion...	5.1849	2020-06-17 15:58:17	209.165.201.17	58016	209.165.200.235	80	6	ET CURRENT_EVENTS QNAP Shellshock CVE-2014-6271
RT	1	seconion...	1.2330	2020-06-17 16:42:09	0.0.0.0		0.0.0.0		0	[OSSEC] unix_chkpwd: Password check failed.
RT	1	seconion...	7.4281	2020-06-17 16:45:23	209.165.201.17	58524	209.165.200.235	80	6	ET TROJAN CozyDuke APT HTTP Checkin

IP Resolution Agent Status Snort Statistics System Msgs User Msgs

Reverse DNS Enable External DNS

Src IP:
Src Name:
Dst IP:
Dst Name:
Whois Query: ☐ None ☐ Src IP ☐ Dst IP

Show Packet Data Show Rule

IP	Source IP	Dest IP	Ver	HL	TOS	len	ID	Flags	Offset	TTL	ChkSum	
TCP	Source Port	Dest Port	U R R C S S Y I	0	G K H T N N	Seq #	Ack #	Offset	Res	Window	Urp	ChkSum
DATA												

Search Packet Payload ☐ Hex ☐ Text ☐ NoCase

Sguil Window

Rules and Alerts

- Alerts can come from several sources:
 - NIDS** - Snort, Zeek, and Suricata
 - HIDS** - OSSEC, Wazuh
 - HTTP, DNS, and TCP transactions** - Recorded by Zeek and pcaps
 - Syslog messages** - Multiple sources

The screenshot shows a Snort rule configuration at the top, labeled 'Rule'. The rule is: `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`. Below the rule, an orange arrow points down to an alert entry in a table, labeled 'Alert'. The table has columns for ID, Rule, Action, Time, Source IP, Destination IP, Source Port, Destination Port, and Message. The alert entry shows: ID 1, Rule seconion-eth1-1, Action 5.23, Time 2017-06-19 23:51:12, Source IP 209.165.201.17, Destination IP 209.165.200.235, Source Port 40599, Destination Port 21, and Message ET EXPLOIT VSFTPD Backdoor User Login Smiley.

ID	Rule	Action	Time	Source IP	Destination IP	Source Port	Destination Port	Message
1	seconion-eth1-1	5.23	2017-06-19 23:51:12	209.165.201.17	209.165.200.235	40599	21	ET EXPLOIT VSFTPD Backdoor User Login Smiley

- 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

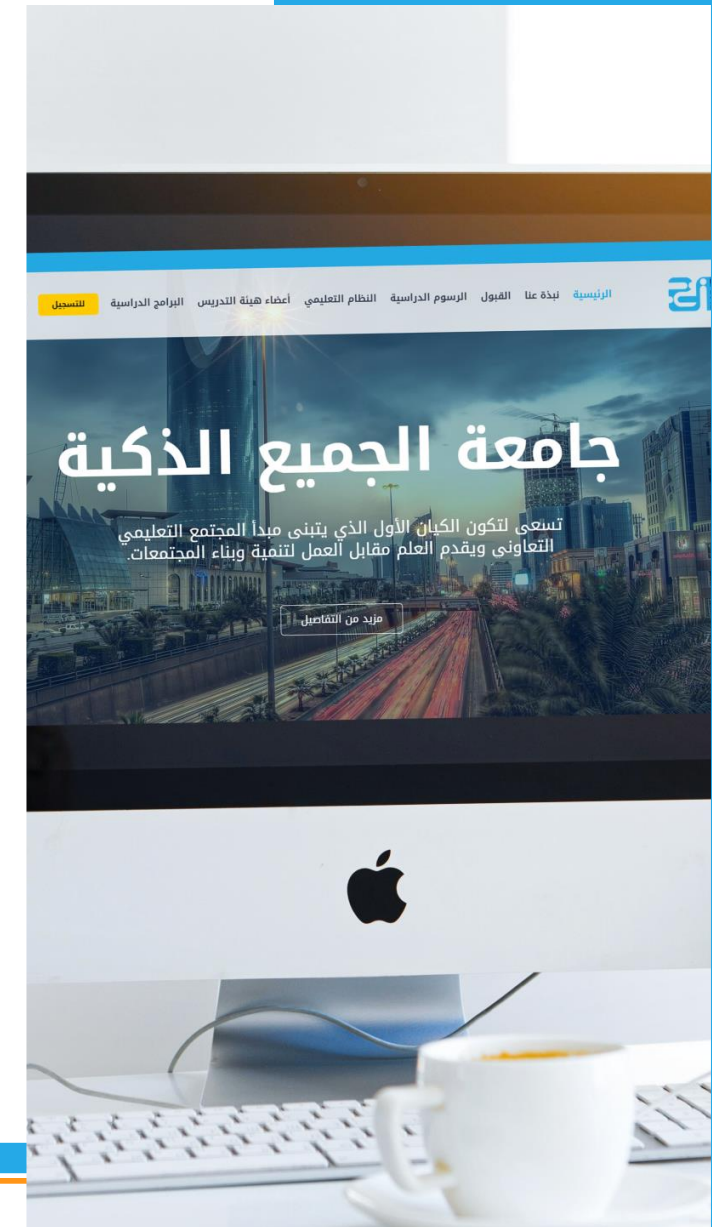
- The structure of the options section of the rule is variable. It is the portion of the rule that is enclosed in parenthesis, as shown in the figure. It contains the text message that identifies the alert. It also contains metadata about the alert, such as a URL.
- Snort rule messages may include the 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
```

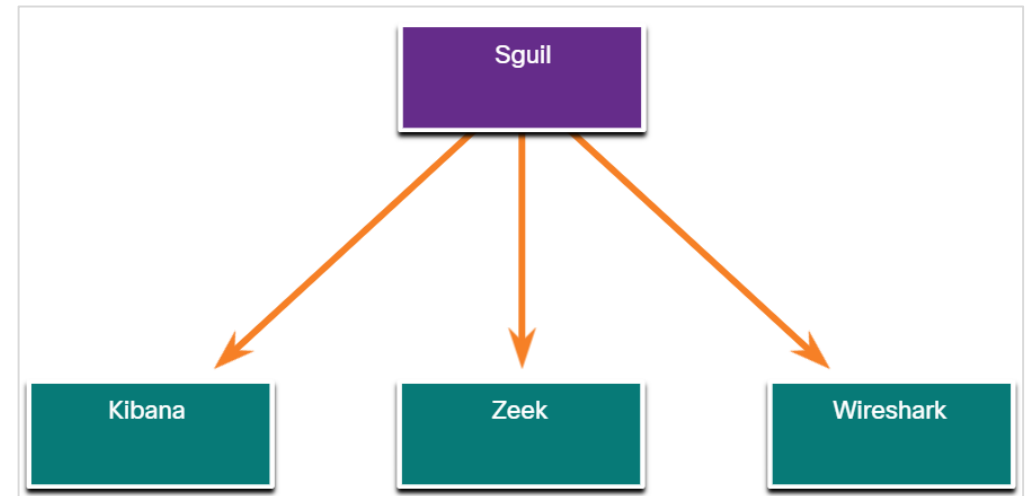
Component	Explanation
msg:	Text that describes the alert.
content:	Refers to content of the packet. In this case, an alert will be sent if the literal text “uid=0(root)” appears anywhere in the packet data. Values specifying the location of the text can be provided.
reference:	This is not shown in the 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.

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.

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.

Extract an Executable from a PCAP

Looking at logs is very important, but it is also important to understand how network transactions happen at the packet level.

In this lab, you will complete the following objective:

- Analyze the traffic in a previously captured pcap file and extract an executable file from the traffic.

Lecture – 8

Activity 2

Interpret HTTP and DNS Data to Isolate Threat Actor

In this lab, you will complete the following objective:

- Investigate SQL injection and DNS exfiltration exploits using Security Onion tools.

Thanks

