



10- Implementing Network Security Appliances

Ahmed Sultan

Senior Technical Instructor
ahmedsultan.me/about

Outlines

10.1- Implement Firewalls and Proxy Servers

10.2- Implement Network Security Monitoring

10.3- Summarize the Use of SIEM

Labs

Lab 14: Configuring a Firewall

Lab 15: Configuring an Intrusion Detection System

10.1- Implement Firewalls and Proxy Servers

10.2- Implement Network Security Monitoring

10.3- Summarize the Use of SIEM

PACKET FILTERING FIREWALLS

- Access Control Lists (ACLs)

- ✓ A packet filtering firewall is configured by specifying a group of rules, called an **access control list (ACL)**.
- ✓ Each rule defines a specific type of data packet and the appropriate action to take when a packet matches the rule.
- ✓ An action can be either to **deny** (block or drop the packet, and optionally log an event) or to **accept** (let the packet pass through the firewall).
- ✓ A packet filtering firewall can inspect the headers of IP packets.
- ✓ This means that rules can be based on the information found in those headers:
 - **IP filtering**—accepting or denying traffic on the basis of its source and/or destination IP address.
 - **Protocol ID/type**—(TCP, UDP, ICMP, routing protocols, and so on).
 - **Port filtering/security**—accepting or denying a packet on the basis of source and destination port numbers (TCP or UDP application type).

PACKET FILTERING FIREWALLS (cont.)

- There may be additional functionality in some products, such as the ability to block some types of **ICMP** (ping) traffic but not others, or the ability to filter by hardware (**MAC**) address.
- Another distinction that can be made is whether the firewall can control only inbound traffic or both inbound and outbound traffic.
- This is also often referred to as ingress and egress traffic or filtering.
- Controlling outbound traffic is useful because it can block applications that have not been authorized to run on the network and defeat malware, such as backdoors.
- Ingress and egress traffic is filtered using separate ACLs.

PACKET FILTERING FIREWALLS (cont.)

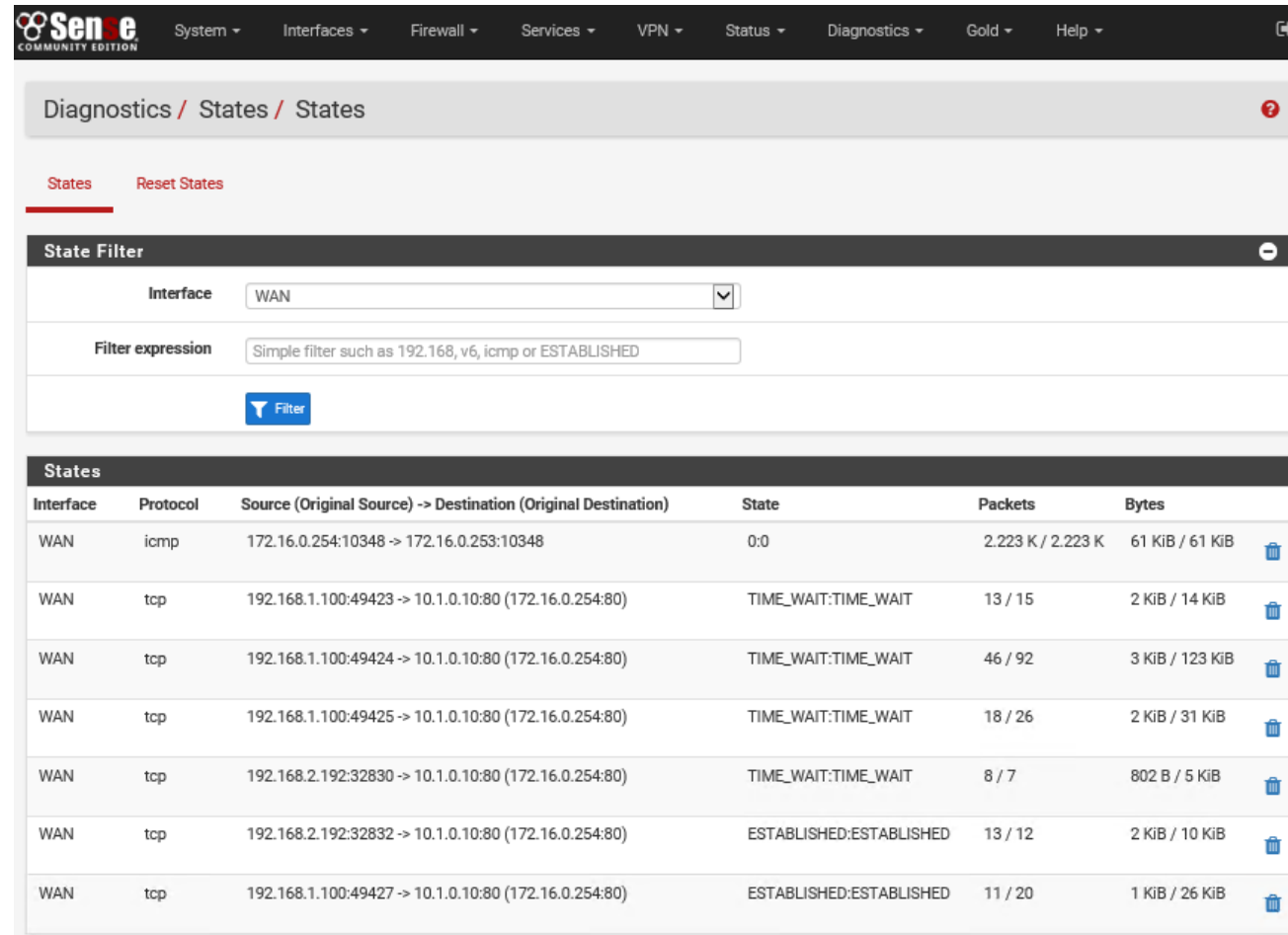
- Stateless Operation

- ✓ A basic packet filtering firewall is **stateless**.
- ✓ This means that it does not preserve information about network sessions.
- ✓ Each packet is analyzed independently, with no record of previously processed packets.
- ✓ This type of filtering requires the least processing effort, but it can be vulnerable to attacks that are spread over a sequence of packets.
- ✓ A stateless firewall can also introduce problems in traffic flow, especially when some sort of load balancing is being used or when clients or servers need to use dynamically assigned ports.

STATEFUL INSPECTION FIREWALLS

- A **stateful inspection firewall** addresses these problems by tracking information about the session established between two hosts, or blocking malicious attempts to start a bogus session.
- The vast majority of firewalls now incorporate some level of stateful inspection capability.
- Session data is stored in a **state table**.
- When a packet arrives, the firewall checks it to confirm whether it belongs to an existing connection.
- If it does not, it applies the ordinary packet filtering rules to determine whether to allow it.

STATEFUL INSPECTION FIREWALLS (cont.)



The screenshot shows the Mikrotik WinBox interface for the Firewall States page. The top navigation bar includes 'System', 'Interfaces', 'Firewall', 'Services', 'VPN', 'Status', 'Diagnostics', 'Gold', and 'Help'. The breadcrumb path is 'Diagnostics / States / States'. Below the breadcrumb, there are tabs for 'States' and 'Reset States'. A 'State Filter' section allows filtering by 'Interface' (set to 'WAN') and 'Filter expression' (with a placeholder 'Simple filter such as 192.168, v6, icmp or ESTABLISHED'). A 'Filter' button is present. The main table, titled 'States', lists active firewall states with columns for Interface, Protocol, Source (Original Source) -> Destination (Original Destination), State, Packets, and Bytes. Each row also has a trash icon for deletion.

Interface	Protocol	Source (Original Source) -> Destination (Original Destination)	State	Packets	Bytes
WAN	icmp	172.16.0.254:10348 -> 172.16.0.253:10348	0:0	2.223 K / 2.223 K	61 KiB / 61 KiB
WAN	tcp	192.168.1.100:49423 -> 10.1.0.10:80 (172.16.0.254:80)	TIME_WAIT:TIME_WAIT	13 / 15	2 KiB / 14 KiB
WAN	tcp	192.168.1.100:49424 -> 10.1.0.10:80 (172.16.0.254:80)	TIME_WAIT:TIME_WAIT	46 / 92	3 KiB / 123 KiB
WAN	tcp	192.168.1.100:49425 -> 10.1.0.10:80 (172.16.0.254:80)	TIME_WAIT:TIME_WAIT	18 / 26	2 KiB / 31 KiB
WAN	tcp	192.168.2.192:32830 -> 10.1.0.10:80 (172.16.0.254:80)	TIME_WAIT:TIME_WAIT	8 / 7	802 B / 5 KiB
WAN	tcp	192.168.2.192:32832 -> 10.1.0.10:80 (172.16.0.254:80)	ESTABLISHED:ESTABLISHED	13 / 12	2 KiB / 10 KiB
WAN	tcp	192.168.1.100:49427 -> 10.1.0.10:80 (172.16.0.254:80)	ESTABLISHED:ESTABLISHED	11 / 20	1 KiB / 26 KiB

STATEFUL INSPECTION FIREWALLS (cont.)

- Stateful inspection can occur at two layers: [transport](#) and [application](#).
- [Transport Layer \(OSI Layer 4\)](#)
 - ✓ At the transport layer, the firewall examines the TCP three-way handshake to distinguish new from established connections.
 - ✓ A legitimate TCP connection should follow a SYN > SYN/ACK > ACK sequence to establish a session, which is then tracked using sequence numbers.
 - ✓ Deviations from this, such as SYN without ACK or sequence number anomalies, can be dropped as malicious flooding or session hijacking attempts.
 - ✓ The firewall can be configured to respond to such attacks by blocking source IP addresses and throttling sessions.
 - ✓ It can also track UDP connections, though this is harder as UDP is a connectionless protocol.
 - ✓ It is also likely to be able to detect IP header and ICMP anomalies.

STATEFUL INSPECTION FIREWALLS (cont.)

Advanced Options	
Source OS	<div>Any</div> <div>Note: this only works for TCP rules. General OS choice matches all subtypes.</div>
Diffserv Code Point	<div></div>
Allow IP options	<input type="checkbox"/> Allow packets with IP options to pass. Otherwise they are blocked by default. This is usually only seen with multicast traffic.
Disable reply-to	<input type="checkbox"/> Disable auto generated reply-to for this rule.
Tag	<div></div> <div>A packet matching this rule can be marked and this mark used to match on other NAT/filter rules. It is called Policy filtering.</div>
Tagged	<div></div> <div>A packet can be matched on a mark placed before on another rule.</div>
Max. states	<div></div> <div>Maximum state entries this rule can create.</div>
Max. src nodes	<div></div> <div>Maximum number of unique source hosts.</div>
Max. connections	<div></div> <div>Maximum number of established connections per host (TCP only).</div>
Max. src. states	<div></div> <div>Maximum state entries per host.</div>
Max. src. conn. Rate	<div></div> <div>Maximum new connections per host (TCP only).</div>
Max. src. conn. Rates	<div></div> <div>/ per how many second(s) (TCP only)</div>
State timeout	<div></div> <div>State Timeout in seconds (TCP only)</div>

STATEFUL INSPECTION FIREWALLS (cont.)

- Application Layer (OSI Layer 7)

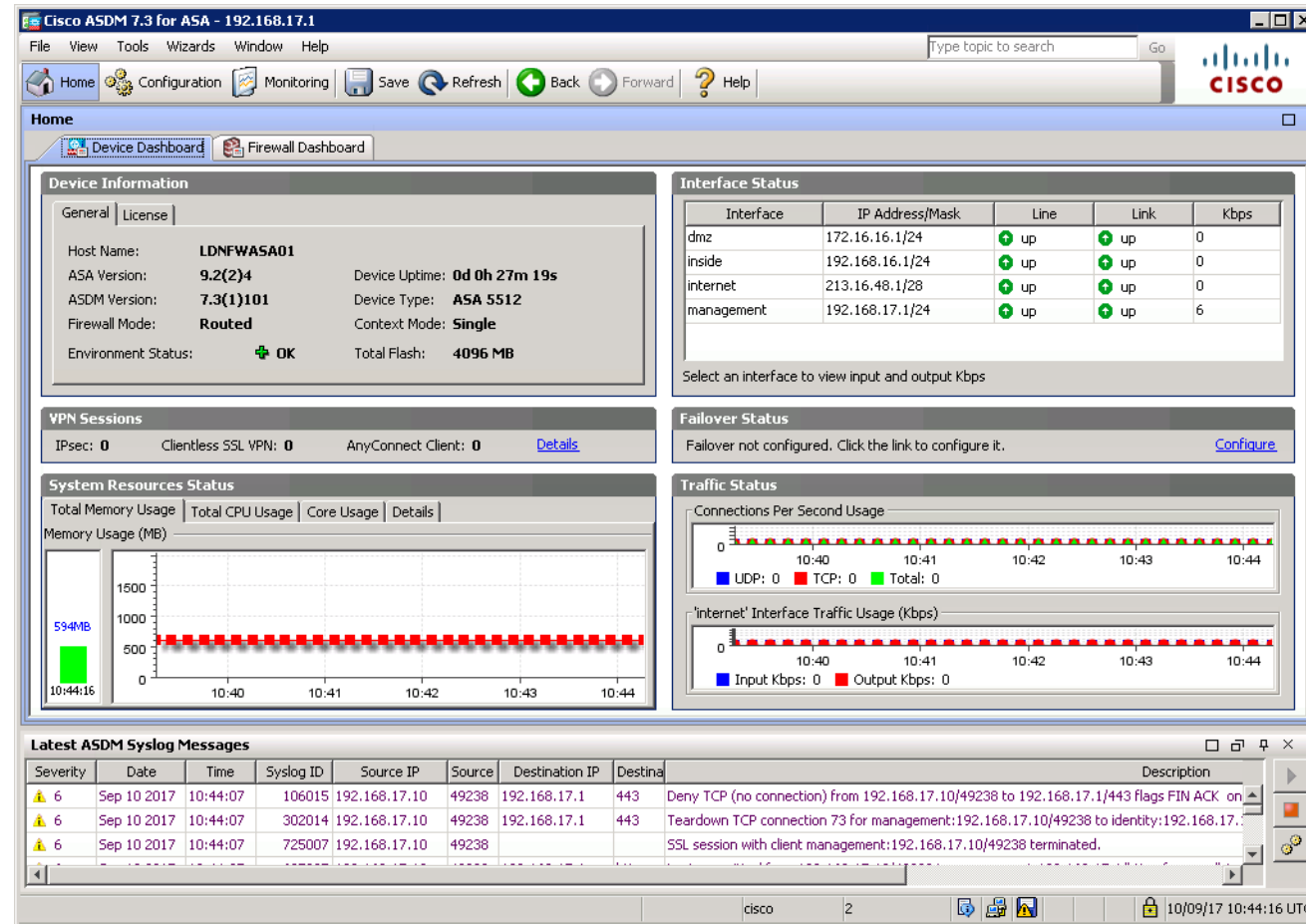
- ✓ An application-aware firewall can inspect the contents of packets at the application layer.
- ✓ One key feature is to verify the application protocol matches the port; to verify that malware isn't sending raw TCP data over port **80** just because port **80** is open, for instance.
- ✓ As another example, a web application firewall could analyze the HTTP headers and the HTML code present in HTTP packets to try to identify code that matches a pattern in its threat database.
- ✓ Application-aware firewalls have many different names, including application layer gateway, stateful multilayer inspection, or deep packet inspection.
- ✓ Application aware devices have to be configured with separate filters for each type of traffic (HTTP and HTTPS, SMTP/POP/IMAP, FTP, and so on).
- ✓ The firewall cannot examine encrypted data packets, unless configured with an **SSL/TLS** inspector.

FIREWALL IMPLEMENTATION

- Firewall Appliances

- ✓ An appliance firewall is a stand-alone hardware firewall deployed to monitor traffic passing into and out of a network zone.
- ✓ A firewall appliance can be deployed in two ways:
 - **Routed (layer 3)**—the firewall performs forwarding between subnets, Each interface on the firewall connects to a different subnet and represents a different security zone.
 - **Bridged (layer 2)**—the firewall inspects traffic passing between two nodes, such as a router and a switch, This is also referred to as transparent mode, The firewall does not have an IP interface (except for configuration management), It bridges the Ethernet interfaces between the two nodes, Despite performing forwarding at layer 2, the firewall can still inspect and filter traffic on the basis of the full range of packet headers, The typical use case for a transparent firewall is to deploy it without having to reconfigure subnets and reassign IP addresses on other devices.

FIREWALL IMPLEMENTATION (cont.)



FIREWALL IMPLEMENTATION (cont.)

- A **router firewall** or firewall router appliance implements filtering functionality as part of the router firmware.
- The difference is that a router appliance is primarily designed for routing, with firewall as a secondary feature.
- SOHO Internet router/modems come with a firewall built-in, for example.

VIRTUAL FIREWALLS

- Virtual firewalls are usually deployed within data centers and cloud services.
- A virtual firewall can be implemented in three different ways:
 - ✓ **Hypervisor-based**—this means that filtering functionality is built into the hypervisor or cloud provisioning tool, You can use the cloud's web app or application programming interface (API) to write access control lists (ACLs) for traffic arriving or leaving a virtual host or virtual network.
 - ✓ **Virtual appliance**—this refers to deploying a vendor firewall appliance instance using virtualization, in the same way you might deploy a Windows or Linux guest OS..
 - ✓ **Multiple context**—this refers to multiple virtual firewall instances running on a hardware firewall appliance, Each context has a separate interface and can perform a distinct filtering role.

Lab

Lab 14: Configuring a Firewall

10.1- Implement Firewalls and Proxy Servers

10.2- Implement Network Security Monitoring

10.3- Summarize the Use of SIEM

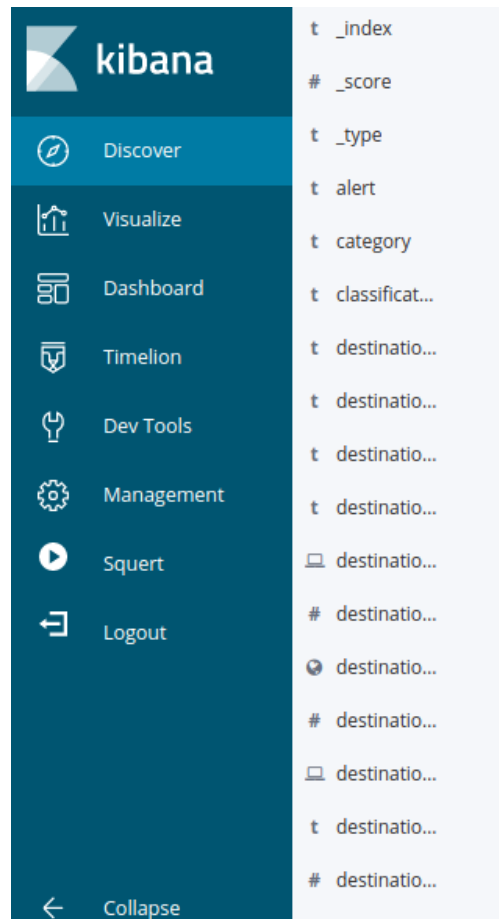
NETWORK-BASED INTRUSION DETECTION SYSTEMS

- An **intrusion detection system (IDS)** is a means of using software tools to provide real-time analysis of either **network traffic** or **system** and **application logs**.
- A **network-based IDS (NIDS)** captures traffic via a **packet sniffer**, referred to as a sensor.
- It analyzes the packets to identify **malicious traffic** and displays alerts to a console or dashboard.
- A NIDS, such as **Snort** (snort.org), **Suricata** (suricata-ids.org), or **Zeek/Bro** (zeek.org) performs passive detection.
- When traffic is matched to a detection signature, it raises an alert or generates a log entry, but does not block the source host.
- This type of passive sensor does not slow down traffic and is undetectable by the attacker.

NETWORK-BASED INTRUSION DETECTION SYSTEMS (cont.)

- A **NIDS** is used to identify and log hosts and applications and to detect attack signatures, password guessing attempts, port scans, worms, backdoor applications, malformed packets or sessions, and policy violations (ports or IP addresses that are not permitted, for instance).
- You can use analysis of the logs to tune firewall rulesets, remove or block suspect hosts and processes from the network, or deploy additional security controls to mitigate any threats you identify.

NETWORK-BASED INTRUSION DETECTION SYSTEMS (cont.)



Time	_source
March 16th 2020, 13:57:40.947	destination_ips: 195.2.253.92 message: [1:2003380:12] ET USER_AGENTS Suspicious User-Agent - Possible Trojan Downloader (ver18/ver19 etc) [Classification: A Network Trojan was detected] [Priority: 1]: <siem-eth1-1> {TCP} 192.168.3.35:1037
<div>TableJSONView surrounding documentsView single document</div>	
@timestamp	March 16th 2020, 13:57:40.947
@version	1
_id	BQyi43ABPEdm6QZiyTol
_index	siem:logstash-ids-2020.03.16
_score	-
_type	doc
alert	ET USER_AGENTS Suspicious User-Agent - Possible Trojan Downloader (ver18/ver19 etc)
category	user_agents
classification	A Network Trojan was detected
destination_geo.continent_code	EU

NETWORK-BASED INTRUSION PREVENTION SYSTEMS

- Compared to the passive function of an IDS, an **intrusion prevention system (IPS)** can provide an **active response** to any network threats that it matches.
- One typical preventive measure is to **end the TCP session, sending a TCP reset packet to the attacking host.**
- *Another option is for the IPS to apply a temporary filter on the firewall to block the attacker's IP address.*
- Other advanced measures include throttling bandwidth to attacking hosts, applying complex firewall filters, and even modifying suspect packets to render them harmless.
- Finally, the appliance may be able to run a script or third-party program to perform some other action not supported by the IPS software itself.

NETWORK-BASED INTRUSION PREVENTION SYSTEMS (cont.)

- Some **IPS** provide inline, wire-speed antivirus scanning.
- Their rulesets can be configured to provide user content filtering, such as blocking URLs, applying keyword-sensitive block lists or allow lists, or applying time-based access restrictions.
- **IPS appliances** are positioned like firewalls **at the border between two network zones**.
- As with proxy servers, the appliances are "inline" with the network, meaning that all traffic passes through them.
- This means that they need to be able to cope with high bandwidths and process each packet very quickly to avoid slowing down the network.

SIGNATURE-BASED DETECTION

- In an **IDS**, the analysis engine is the component that scans and interprets the traffic captured by the sensor with the purpose of identifying suspicious traffic.
- The analysis engine determines how any given event should be classed, with typical options to ignore, log only, alert, and block (IPS).
- The analysis engine is programmed with a set of rules that it uses to drive its decision-making process.
- There are several methods of formulating the ruleset.
 - ✓ Signature-based detection
 - ✓ Behavioral-based detection

SIGNATURE-BASED DETECTION (cont.)

- Signature-based detection

- ✓ (or pattern-matching) means that the engine is loaded with a database of attack patterns or signatures.
- ✓ If traffic matches a pattern, then the engine generates an incident.
- ✓ The signatures and rules (often called **plug-ins** or **feeds**) powering intrusion detection need to be updated regularly to provide protection against the latest threat types.
- ✓ Commercial software requires a paid-for subscription to obtain the updates.
- ✓ It is important to ensure that the software is configured to update only from valid repositories, ideally using a secure connection method, such as HTTPS.

SIGNATURE-BASED DETECTION (cont.)

- Signature-based detection (cont.)

```
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Internet Explorer Plugin.ocx Heap Overfl$
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX winhlp32 ActiveX control attack - phase 1$
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX winhlp32 ActiveX control attack - phase 2$
alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX winhlp32 ActiveX control attack - phase 3$
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX MciWndx ActiveX Control"; flow:from_serv$
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX COM Object Instantiation Memory Corrupti$
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Danim.dll and Dxtmsft.dll COM Objects"; $
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX JuniperSetup Control Buffer Overflow"; f$
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Wmm2fxa.dll COM Object Instantiation Mem$
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Microsoft Multimedia Controls - ActiveX $
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Microsoft Multimedia Controls - ActiveX $
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Microsoft Multimedia Controls - ActiveX $
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Microsoft WMIScriptUtils.WMIOObjectBroker$
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Microsoft VsmIDE.DTE object call CSLID";$
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Microsoft DExplore.AppObj.8.0 object cal$
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Microsoft VisualStudio.DTE.8.0 object ca$
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Microsoft Microsoft.DbgClr.DTE.8.0 objec$
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Microsoft VsaIDE.DTE object call CSLID";$
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Microsoft Business Object Factory object$
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Microsoft Outlook Data Object object cal$
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Microsoft Outlook.Application object cal$
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX ACTIVEX Possible Microsoft IE Install En$
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Possible Microsoft IE Install Engine Ins$
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Possible Microsoft IE Shell.Application $
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX ACTIVEX Possible Microsoft IE Shell.Appl$
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX NCTAudioFile2 ActiveX SetFormatLikeSampl$
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Possible Microsoft Internet Explorer ADO$
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Sony ImageStation (SonyISUpload.cab 1.0.$
#alert tcp $EXTERNAL_NET $HTTP_PORTS -> $HOME_NET any (msg:"ET ACTIVEX Citrix Presentation Server Client WFICA.$
```

```
[ Read 27185 lines (Warning: No write permission) ]
```

BEHAVIOR AND ANOMALY-BASED DETECTION

- Behavioral-based detection

- ✓ Means that the engine is trained to recognize baseline "normal" traffic or events.
- ✓ Anything that deviates from this baseline (outside a defined level of tolerance) generates an incident.
- ✓ The idea is that the software will be able to identify zero day attacks, insider threats, and other malicious activity for which there is no signature.
- ✓ This type of detection was provided by **network behavior and anomaly detection (NBAD)** products.
- ✓ An **NBAD** engine uses heuristics to generate a statistical model of what baseline normal traffic looks like.
- ✓ It may develop several profiles to model network use at different times of the day.
- ✓ This means that the system generates **false positive** and **false negatives** until it has had time to improve its statistical model of what is "normal."
- ✓ A **false positive** is where legitimate behavior generates an alert, while a **false negative** is where malicious activity is not alerted.

NEXT-GENERATION FIREWALLS

- While intrusion detection was originally produced as standalone software or appliances, its functionality very quickly became incorporated into a new generation of firewalls.
- The original [next-generation firewall \(NGFW\)](#) was released as far back as **2010** by **Palo Alto**.
- This product combined application-aware filtering with user account-based filtering and the ability to act as an intrusion prevention system (IPS).
- This approach was quickly adopted by competitor products.
- Subsequent firewall generations have added capabilities such as cloud inspection and combined features of different security technologies.

Unified Threat Management (UTM)

- **Unified threat management (UTM)** refers to a security product that centralizes many types of security controls—**firewall, anti-malware, network intrusion prevention, spam filtering, content filtering, data loss prevention, VPN, cloud access gateway**—into a single appliance.
- This means that you can monitor and manage the controls from a single console.
- UTM has some downsides, When defense is unified under a single system, this creates the potential for a single point of failure that could affect an entire network.
- Additionally, UTM systems can struggle with latency issues if they are subject to too much network activity.
- Also, a UTM might not perform as well as software or a device with a single dedicated security function.

Unified Threat Management (UTM) (cont.)

- ✓ ***NGFW** and **UTM** are just marketing terms.*
- ✓ *A **UTM** is seen as a "**do everything**" solution, while a **NGFW** is an enterprise product with fewer features, or more modularization, and greater configuration complexity, but better performance.*
- ✓ *It can be more helpful to focus on the specific product features, rather than trying to present an implementation decision as a choice of either a **NGFW** or a **UTM**.*

WEB APPLICATION FIREWALLS

- A **web application firewall (WAF)** is designed specifically to protect software running on web servers and their back-end databases from code injection and DoS attacks.
- **WAFs** use application-aware processing rules to filter traffic and perform application-specific intrusion detection.
- The **WAF** can be programmed with signatures of known attacks and use pattern matching to block requests containing suspect code.
- The output from a **WAF** will be written to a log, which you can inspect to determine what threats the web application might be subject to.

WEB APPLICATION FIREWALLS (cont.)

- A **WAF** may be deployed as an appliance or as plug-in software for a web server platform.
- Some examples of WAF products include:
 - ✓ **ModSecurity** (modsecurity.org) is an open source (sponsored by Trustwave) WAF for Apache, nginx, and IIS.
 - ✓ **NAXSI** (github.com/nbs-system/naxsi) is an open source module for the nginx web server software.
 - ✓ **Imperva** (imperva.com) is a commercial web security offering with a particular focus on data centers, Imperva markets WAF, DDoS, and database security through its **SecureSphere** appliance.

Lab

Lab 15: Configuring an Intrusion Detection System

10.1- Implement Firewalls and Proxy Servers

10.2- Implement Network Security Monitoring

10.3- Summarize the Use of SIEM

MONITORING SERVICES

- Security assessments and incident response both require real-time monitoring of host and network status indicators plus audit information.
- Packet Capture
 - ✓ Data captured from network sensors/sniffers plus netflow sources provides both summary statistics about bandwidth and protocol usage and the opportunity for detailed frame analysis.

MONITORING SERVICES (cont.)

- Network Monitors

- ✓ A network monitor collects data about network appliances, such as **switches, access points, routers, firewalls, and servers.**
- ✓ This is used to monitor load status for CPU/memory, state tables, disk capacity, fan speeds/temperature, network link utilization/error statistics, and so on.
- ✓ Another important function is a **heartbeat message** to indicate availability.
- ✓ This data might be collected using the **Simple Network Management Protocol (SNMP)** or a proprietary management system.
- ✓ As well as supporting availability, network monitoring might reveal unusual conditions that could point to some kind of attack.

MONITORING SERVICES (cont.)

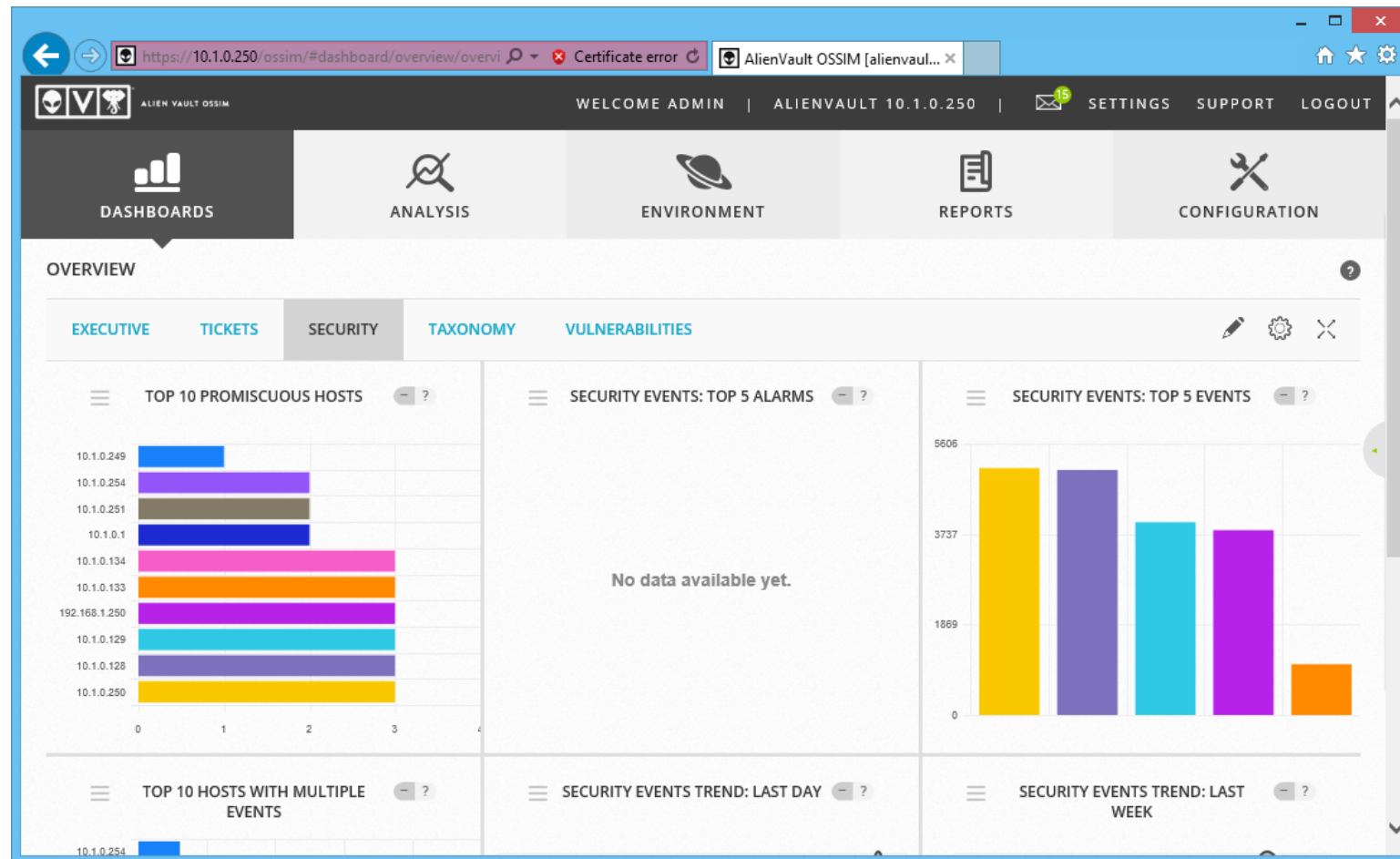
- Logs

- ✓ Logs are one of the most valuable sources of security information.
- ✓ A system log can be used to diagnose availability issues.
- ✓ A security log can record both authorized and unauthorized uses of a resource or privilege.
- ✓ Logs function both as an audit trail of actions and provide a warning of intrusion attempts.
- ✓ Log review is a critical part of security assurance.
- ✓ Only referring to the logs following a major incident is missing the opportunity to identify threats and vulnerabilities early and to respond proactively.

SECURITY INFORMATION AND EVENT MANAGEMENT (SIEM)

- Software designed to assist with managing security data inputs and provide reporting and alerting is often described as **security information and event management (SIEM)**.
- The core function of a SIEM tool is to aggregate traffic data and logs.
- In addition to logs from **Windows** and **Linux-based hosts**, this could include **switches, routers, firewalls, IDS sensors, vulnerability scanners, malware scanners, data loss prevention (DLP) systems, and databases**.

SECURITY INFORMATION AND EVENT MANAGEMENT (SIEM) (cont.)



Security Orchestration, Automation, and Response (SOAR)

- Security orchestration, automation, and response (SOAR) is designed as a solution to the problem of the volume of alerts overwhelming analysts' ability to respond.
- A SOAR may be implemented as a standalone technology or integrated with a SIEM—often referred to as a next-gen SIEM.
- The basis of SOAR is to scan the organization's store of security and threat intelligence, analyze it using machine/deep learning techniques, and then use that data to automate and provide data enrichment for the workflows that drive incident response and threat hunting.