Network Security Review

Cybersecurity

Review Week Day 2

Times Up! Let’s Review.

Class Objectives

In today's class, we will review Linux by:

- Design and implement firewall policies with UFW
- Use Wireshark to analyze malicious traffic
- Schedule backups with `cron` and `tar` to protect availability

Class Overview

Today's class revolves around four exercises covering fundamental network security topics:

01 Exercise 1: **Firewall Policies**

02 Exercise 2: **Sniffing Web Attacks**

03 Exercise 3: **Interpreting Snort Rules**

04 Exercise 4: **Scheduling Back-Ups**

Firewall Policies

Exercise 1: Firewall Policies

For this activity you will work with a partner to design and implement a firewall policy by:

- Determining which services each machine provides
- Determining which ports those services require
- Designing and documenting a firewall policy for each machine
- Implementing the policy on each machine

Your Turn: Firewall Policies

In this activity, you will play the role of a **network administrator** responsible for securing a network. You will determine which services should be available, determine which ports they require, and then configure firewalls to expose only those ports..

Activities/Stu_Firewall_Policies

Suggested Time:
20 Minutes

Firewall Policies Review: Auditing

1. Developer Level Firewall Change Request

User Data API

- Service(s): _____
- Port(s): _____

User Database

- Service(s): _____
- Port(s): _____

Command Server

- Service(s): _____
- Port(s): _____

Description: Web server that provides access to organization's IAM database.

This machine runs an HTTP and HTTPS server, which should be available to all machines on the subnet.

There is also an SSH listener, which should only allow access from the command server.

Disable connections to all other running services.

Firewall Policies Review: Auditing

1. Developer Level Firewall Change Request

User Data API

- Service(s): `HTTP, SSH, HTTPS`
- Port(s): `22, 80, 443`

User Database

- Service(s): _____
- Port(s): _____

Command Server

- Service(s): _____
- Port(s): _____

Description: Web server that provides access to organization's IAM database.

This machine runs an HTTP and HTTPS server, which should be available to all machines on the subnet.

There is also an SSH listener, which should only allow access from the command server.

Disable connections to all other running services.

Firewall Policies Review: Auditing

1. Developer Level Firewall Change Request

User Data API

- Service(s): **`HTTP, HTTPS`**
- Port(s): **`80, 433`**

User Database

- Service(s): _____
- Port(s): _____

Command Server

- Service(s): _____
- Port(s): _____

Description:

MySQL server containing user data for IAM policies.

Should *only* be accessible by the User Data API server.

Firewall Policies Review: Auditing

1. Developer Level Firewall Change Request

User Data API

- Service(s): **`HTTP, HTTPS`**
- Port(s): **`80, 433`**

User Database

- Service(s): `MySQL`
- Port(s): `3306`

Command Server

- Service(s): _____
- Port(s): _____

Description:

MySQL server containing user data for IAM policies.

Should *only* be accessible by the User Data API server

Firewall Policies Review: Auditing

1. Developer Level Firewall Change Request

User Data API

- Service(s): **HTTP, HTTPS**
- Port(s): **80, 433**

User Database

- Service(s): **MySQL**
- Port(s): **3306**

Command Server

- Service(s): _____
- Port(s): _____

Description:
Server responsible for controlling whether other servers on the network are up or down.

Allows administrators to login remotely to issue commands.

Block all requests to the telnet server that *do not* come from the local subnet.

Allow outgoing SSH and telnet connections to other machines within the subnet.

Firewall Policies Review: Auditing

1. Developer Level Firewall Change Request

User Data API

- Service(s): **`HTTP, HTTPS`**
- Port(s): **`80, 433`**

User Database

- Service(s): **`MySQL`**
- Port(s): **`3306`**

Command Server

- Service(s): `SSH, Telnet`
- Port(s): `22, 23`

Description:
Server responsible for controlling whether other servers on the network are up or down.

Allows administrators to login remotely to issue commands.

Block all requests to the telnet server that *do not* come from the local subnet.

Allow outgoing SSH and telnet connections to other machines within the subnet.

Firewall Policies Review: Generating Documentation

Generate documentation by **implementing the firewall** and **exporting you rules**.

1. Create a directory to save firewall information in: `~/security/firewall_policies`. Move into this directory.
2. Create a new file for each machine—e.g., `command_server.ufw.rules`.
3. Write the UFW commands you would run to implement the correct firewall rules in each file.

Firewall Policies Review: Generating Documentation

Complete the following support documentation:

1. Create a directory to save firewall information in: `~/security/firewall_policies`. Move into this directory.
2. Create a new file for each machine—e.g., `command_server.ufw.rules`.
3. Write the UFW commands you would run to implement the correct firewall rules in each file.

Solution:

```
$ mkdir -p ~/security/firewall_policies
$ cd ~/security/firewall_policies
$ touch command_server.ufw.rule api_server.ufw.rules
userdb.ufw.rules
```

Firewall Policies Review: Generating Documentation

Complete the following support documentation:

1. Create a directory to save firewall information in: `~/security/firewall_policies`. Move into this directory.
2. Create a new file for each machine—e.g., `command_server.ufw.rules`.
3. Write the UFW commands you would run to implement the correct firewall rules in each file.

Solution:

```
$ mkdir -p ~/security/firewall_policies
$ cd ~/security/firewall_policies
$ cat command_server.ufw.rules
  sudo ufw allow from 192.168.12.0/24 to any port 22
  sudo ufw allow from 192.168.12.0/24 to any port 23
```

Firewall Policies Review: Generating Documentation

Complete the following support documentation:

1. Create a directory to save firewall information in: `~/security/firewall_policies`. Move into this directory.
2. Create a new file for each machine—e.g., `command_server.ufw.rules`.
3. Write the UFW commands you would run to implement the correct firewall rules in each file.

Solution:

```
$ mkdir -p ~/security/firewall_policies
$ cd ~/security/firewall_policies
$ cat userdb.ufw.rules
  sudo ufw allow from 192.168.12.50 to any port 3306
```

Firewall Policies Review: Generating Documentation

Complete the following support documentation:

1. Create a directory to save firewall information in: `~/security/firewall_policies`. Move into this directory.
2. Create a new file for each machine—e.g., `command_server.ufw.rules`.
3. Write the UFW commands you would run to implement the correct firewall rules in each file.

Solution:

```
$ mkdir -p ~/security/firewall_policies
$ cd ~/security/firewall_policies
$ cat api_server.ufw.rules
  sudo ufw allow from 192.168.12.100 to any port 22
  sudo ufw allow from 192.168.12.0/24 to any port 80
  sudo ufw allow from 192.168.12.0/24 to any port 443
```

Sniffing Web Attacks

Sniffing Web Attacks

In the next activity, you will act as a Network Engineer.

- Recently, one of your techs noticed what seemed to be an **SQL injection attack** on the company website.
- They have saved the **PCAP file** and handed it to you for **further analysis**.
- **Using Wireshark**, you will need to identify what payloads were used, and if the attack was successful.

Your Turn: Sniffing Web Attacks

- Open the Pcap Activities/Stu_Sniffing_Web_Attacks: http_dvwa_sqlinjection.pcapng
- Find the packets containing the SQLi payloads and record the three payloads you found.

  Hint: Use a tool like URL decoder: https://www.urldecoder.org/

- Determine if the attacks were successful and support your conclusion with evidence.

Suggested Time:
20 Minutes

Sniffing Web Attacks

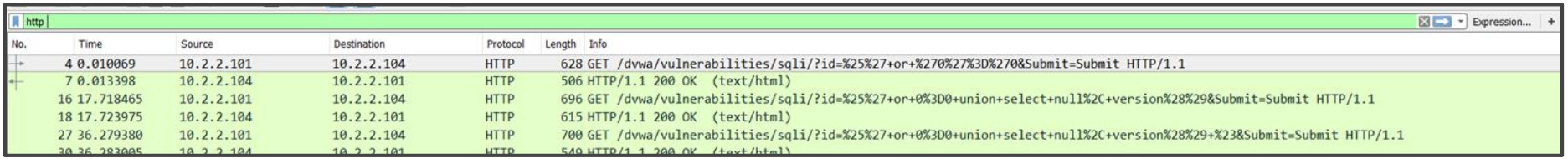
Step 1: Open the pcap in http_dvwa_sqlinjection.pcapng, find the packets containing the SQLi payloads, and record the three payloads you found.

Solution:

Sniffing Web Attacks

Step 1: Open the pcap in [http_dvwa_sqlinjection.pcapng](http://dvwa_sqlinjection.pcapng),find the packets containing the SQLi payloads, and record the three payloads you found.

Solution:



http | Expression... +

No.	Time	Source	Destination	Protocol	Length	Info
4	0.010069	10.2.2.101	10.2.2.104	HTTP	628	GET /dvwa/vulnerabilities/sqli/?id=%25%27+or+%270%27%3D%270&Submit=Submit HTTP/1.1
7	0.013398	10.2.2.104	10.2.2.101	HTTP	506	HTTP/1.1 200 OK (text/html)
16	17.718465	10.2.2.101	10.2.2.104	HTTP	696	GET /dvwa/vulnerabilities/sqli/?id=%25%27+or+0%3D0+union+select+null%2C+version%28%29&Submit=Submit HTTP/1.1
18	17.723975	10.2.2.104	10.2.2.101	HTTP	615	HTTP/1.1 200 OK (text/html)
27	36.279380	10.2.2.101	10.2.2.104	HTTP	700	GET /dvwa/vulnerabilities/sqli/?id=%25%27+or+0%3D0+union+select+null%2C+version%28%29+%23&Submit=Submit HTTP/1.1
30	36.283005	10.2.2.104	10.2.2.101	HTTP	549	HTTP/1.1 200 OK (text/html)

Filter for **`http`**. You'll find three GET requests, with the following payloads.

- `%' or '0'='0`
- `%' or 0=0 union select null, version()`
- `%' or '0'='0' union select null, version()`

Sniffing Web Attacks

Step 2: Were the attacks successful? Support your conclusion with evidence.

`The first payload: %' or '0'='0`

Solution:

Sniffing Web Attacks

Step 2: Were the attacks successful? Support your conclusion with evidence.

The first payload: %' or '0'='0

Solution: Yes, this payload was successful.

```
<!-- Below is an excerpt of the server's response -->
<pre>ID: %' or '0'='0<br />First name: admin<br />Surname: admin</pre>
<pre>ID: %' or '0'='0<br />First name: Gordon<br />Surname: Brown</pre>
<pre>ID: %' or '0'='0<br />First name: Hack<br />Surname: Me</pre>
<pre>ID: %' or '0'='0<br />First name: Pablo<br />Surname: Picasso</pre>
<pre>ID: %' or '0'='0<br />First name: Bob<br />Surname: Smith</pre>
```

Sniffing Web Attacks

Step 2: Were the attacks successful? Support your conclusion with evidence.

The second payload: `%' or 0=0 union select null, version()`

Solution:

Sniffing Web Attacks

Step 2: Were the attacks successful? Support your conclusion with evidence.

The second payload: `%' or 0=0 union select null, version()`

Solution:

No, this response contains an error message, indicating that the SQL injection was unsuccessful.

```
<pre>You have an error in your SQL syntax; check the manual that
corresponds to your MySQL server version for the right syntax to
use near ''' at line 1</pre>>
```

Sniffing Web Attacks

Step 2: Were the attacks successful? Support your conclusion with evidence.

The third payload: `%' or '0'='0' union select null, version()`

Solution:

Sniffing Web Attacks

Step 2: Were the attacks successful? Support your conclusion with evidence.

The third payload: `%' or '0'='0' union select null, version()`

Solution: Yes, this payload was successful.

This response also contains HTML. If you scroll through it, you'll find that it reports the server's OS version: `5.5.54-0ubuntu0.14.04.1`.

```
<!-- Below is an excerpt of the server's response -->
<pre>ID: %' or 0=0 union select null, version()<br />First name: admin<br />Surname: admin</pre>
<pre>ID: %' or 0=0 union select null, version()<br />First name: Gordon<br />Surname: Brown</pre>
<pre>ID: %' or 0=0 union select null, version()<br />First name: Hack<br />Surname: Me</pre>
<pre>ID: %' or 0=0 union select null, version()<br />First name: Pablo<br />Surname:
Picasso</pre>
<pre>ID: %' or 0=0 union select null, version()<br />First name: Bob<br />Surname: Smith</pre>
<pre>ID: %' or 0=0 union select null, version()<br />First name: <br />Surname: 5.5.54-
0ubuntu0.14.04.1</pre>
```

Interpreting Snort Rules

Your Turn: Interpreting Snort Rules

In this exercise, you'll **interpret rules** that fire in response to suspicious ping probes and then use the Snort documentation to **research** additional rule options.

Activites/Stu_Interpreting_Snort_Rules

Suggested Time:
20 Minutes

Interpreting Snort Rules

Explain the below **ICMP Echo Reply** rules. Be sure to identify the:

- Action
- Protocol
- Source/destination addresses
- Meaning of each new rule option

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (
    msg:"PROTOCOL-ICMP Echo Reply";
    icode:0; itype:0;
    metadata:ruleset community;
    classtype:misc-activity;
    sid:408;
    rev:8;)

alert icmp $EXTERNAL_NET any -> $HOME_NET any (
    msg:"PROTOCOL-ICMP Echo Reply undefined code";
    icode:>0;
    itype:0;
    metadata:ruleset community;
    classtype:misc-activity;
    sid:409;
    rev:10;)
```

Interpreting Snort Rules

Explain the below **ICMP Echo Reply** rules. Be sure to identify the:

- Action: alert
- Protocol: icmp
- Source/destination addresses: `$EXTERNAL_NET any -> $HOME_NET any`
- Meaning of each new rule option: `icode`: Checks for a specific ICMP code value.

```
alert icmp $EXTERNAL_NET any -> $HOME_NET any (
    msg:"PROTOCOL-ICMP Echo Reply";
    icode:0; itype:0;
    metadata:ruleset community;
    classtype:misc-activity;
    sid:408;
    rev:8;)

alert icmp $EXTERNAL_NET any -> $HOME_NET any (
    msg:"PROTOCOL-ICMP Echo Reply undefined code";
    icode:>0;
    itype:0;
    metadata:ruleset community;
    classtype:misc-activity;
    sid:409;
    rev:10;)
```

Interpreting Snort Rules

Explain the below **ICMP Unusual Ping** rule. Be sure to identify the:

- Action
- Protocol
- Source/destination addresses
- Meaning of each new rule option

```
alert icmp $HOME_NET any -> $EXTERNAL_NET any (
    msg:"PROTOCOL-ICMP Unusual PING detected";
    icode:0;
    itype:8;
    fragbits:!M;
    content:!"ABCDEFGHIJKLMNOPQRSTUVWABCDEFGHI"; depth:32;
    content:!"0123456789abcdefghijklmnopqrstuv"; depth:32;
    content:!"EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE"; depth:36;
    content:!"WANG2";
    content:!"cacti-monitoring-system"; depth:65;
    content:!"SolarWinds"; depth:72;
    metadata:ruleset community;
    reference:url,krebsonsecurity.com/2014/01/a-closer-look-at-the-target-malware-part-ii/;
    reference:url,krebsonsecurity.com/2014/01/a-first-look-at-the-target-intrusion-malware/;
    classtype:successful-recon-limited;
    sid:29456;
    rev:2;)
```

Interpreting Snort Rules

Explain the below **ICMP Unusual Ping** rule. Be sure to identify the:

- Action: alert
- Protocol: icmp
- Source/destination addresses: `$HOME_NET any -> $EXTERNAL_NET any`
- Meaning of each new rule option:
  - `fragbits: M`: Checks if the "More Fragments" fragmentation and reserved headers are set in the IP packet. This flag indicates that this packet is just one in a stream of packets.
  - reference:url,`: Include a link to relevant documentation.

```
alert icmp $HOME_NET any -> $EXTERNAL_NET any (
    msg:"PROTOCOL-ICMP Unusual PING detected";
    icode:0;
    itype:8;
    fragbits:!M;
    content:!"ABCDEFGHIJKLMNOPQRSTUVWABCDEFGHI"; depth:32;
    content:!"0123456789abcdefghijklmnopqrstuv"; depth:32;
    content:!"EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE"; depth:36;
    content:!"WANG2";
    content:!"cacti-monitoring-system"; depth:65;
    content:!"SolarWinds"; depth:72;
    metadata:ruleset community;
    reference:url,krebsonsecurity.com/2014/01/a-closer-look-at-the-target-malware-part-ii/;
    reference:url,krebsonsecurity.com/2014/01/a-first-look-at-the-target-intrusion-malware/;
    classtype:successful-recon-limited;
    sid:29456;
    rev:2;)
```

Interpreting Snort Rules

1. There's an exclamation mark in front of each content string. What does the ! signify?

2. Why do you suppose this rule includes so many?

```
alert icmp $HOME_NET any -> $EXTERNAL_NET any (
    msg:"PROTOCOL-ICMP Unusual PING detected";
    icode:0;
    itype:8;
    fragbits:!M;
    content:!"ABCDEFGHIJKLMNOPQRSTUVWABCDEFGHI"; depth:32;
    content:!"0123456789abcdefghijklmnopqrstuv"; depth:32;
    content:!"EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE"; depth:36;
    content:!"WANG2";
    content:!"cacti-monitoring-system"; depth:65;
    content:!"SolarWinds"; depth:72;
    metadata:ruleset community;
    reference:url,krebsonsecurity.com/2014/01/a-closer-look-at-the-target-malware-part-ii/;
    reference:url,krebsonsecurity.com/2014/01/a-first-look-at-the-target-intrusion-malware/;
    classtype:successful-recon-limited;
    sid:29456;
    rev:2;)
```

Interpreting Snort Rules

1. There's an exclamation mark in front of each content string. What does the ! signify?

The ! means to match packets that do *not* have these contents.

2. Why do you suppose this rule includes so many?

This rule monitors for unusual ping activity. Each content block includes a different *trusted* string. This rule is a "blanket check" for packets that do *not* contain these trusted strings.

```
alert icmp $HOME_NET any -> $EXTERNAL_NET any (
    msg:"PROTOCOL-ICMP Unusual PING detected";
    icode:0;
    itype:8;
    fragbits:!M;
    content:!"ABCDEFGHIJKLMNOPQRSTUVWABCDEFGHI"; depth:32;
    content:!"0123456789abcdefghijklmnopqrstuv"; depth:32;
    content:!"EEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEE"; depth:36;
    content:!"WANG2";
    content:!"cacti-monitoring-system"; depth:65;
    content:!"SolarWinds"; depth:72;
    metadata:ruleset community;
    reference:url,krebsonsecurity.com/2014/01/a-closer-look-at-the-target-malware-part-ii/;
    reference:url,krebsonsecurity.com/2014/01/a-first-look-at-the-target-intrusion-malware/;
    classtype:successful-recon-limited;
    sid:29456;
    rev:2;)
```

Interpreting Snort Rules

Explain the following rule. Be sure to identify:

- Action
- Protocol
- Source/destination addresses
- Meaning of each new rule option

```
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (
    msg:"SQL PK-CMS SQL injection attempt";
    flow:to_server,established;
    content:"/default.asp?"; fast_pattern;
    nocase; http_uri;
    content:"pagina="; distance:0; http_uri; pcre:"/pagina=[^&]*\x27/Ui";
    metadata:service http;
    reference:url,github.com/BuddhaLabs/PacketStorm-Exploits/blob/master/1309-exploits/pkcms-sql.txt;
    classtype:web-application-attack;
    sid:32768;
    rev:1;)
```

Interpreting Snort Rules

Explain the following rule. Be sure to identify:

- Action: alert
- Protocol: tcp
- Source/destination addresses: `$EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS`
    - This watches traffic from the public internet to local https servers.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (
    msg:"SQL PK-CMS SQL injection attempt";
    flow:to_server,established;
    content:"/default.asp?"; fast_pattern;
    nocase; http_uri;
    content:"pagina="; distance:0; http_uri; pcre:"/pagina=[^&]*\x27/Ui";
    metadata:service http;
    reference:url,github.com/BuddhaLabs/PacketStorm-Exploits/blob/master/1309-exploits/pkcms-sql.txt;
    classtype:web-application-attack;
    sid:32768;
    rev:1;)
```

Interpreting Snort Rules

Meaning of each new rule:

- **`flow:to_server,established`**: Watches traffic flowing from the public Internet *into* the server on an *established* connection. This packet is *not* a request to initiate a connection, rather it's being transferred over an existing one.
- **`fast_pattern`**: Enables fast pattern matching.
- **`nocase`**: Make the search for the content string case-insensitive.
- **distance:0**: Ignore 0 bytes before looking for the content string.
- **http_uri**: Restricts the content search to the HTTP URI field, *not* the rest of the packet.
- **pcre`**: Allows the rule writer to specify a regular expression, so they can look for multiple patterns in a single rule.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (
    msg:"SQL PK-CMS SQL injection attempt";
    flow:to_server,established;
    content:"/default.asp?"; fast_pattern;
    nocase; http_uri;
    content:"pagina="; distance:0; http_uri; pcre:"/pagina=[^&]*\x27/Ui";
    metadata:service http;
    reference:url,github.com/BuddhaLabs/PacketStorm-Exploits/blob/master/1309-exploits/pkcms-sql.txt;
    classtype:web-application-attack;
    sid:32768;
    rev:1;)
```

Interpreting Snort Rules

Based on the reference URL, which exploit does this rule monitor for?

Based on your knowledge of SQL injection and the rule above, which of the following carries the SQLi payload? What is the name of the parameter or header that is being attacked?

```
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (
    msg:"SQL PK-CMS SQL injection attempt";
    flow:to_server,established;
    content:"/default.asp?"; fast_pattern;
    nocase; http_uri;
    content:"pagina="; distance:0; http_uri; pcre:"/pagina=[^&]*\x27/Ui";
    metadata:service http;
    reference:url,github.com/BuddhaLabs/PacketStorm-Exploits/blob/master/1309-exploits/pkcms-sql.txt;
    classtype:web-application-attack;
    sid:32768;
    rev:1;)
```

Interpreting Snort Rules

Based on the reference URL, which exploit does this rule monitor for?

This exploit watches for the `pkcms-sql` exploit.

Based on your knowledge of SQL injection and the rule above, which of the following carries the SQLi payload? What is the name of the parameter or header that is being attacked?

This exploit is delivered via the GET query string, through the parameter `pagina`.

```
alert tcp $EXTERNAL_NET any -> $HOME_NET $HTTP_PORTS (
    msg:"SQL PK-CMS SQL injection attempt";
    flow:to_server,established;
    content:"/default.asp?"; fast_pattern;
    nocase; http_uri;
    content:"pagina="; distance:0; http_uri; pcre:"/pagina=[^&]*\x27/Ui";
    metadata:service http;
    reference:url,github.com/BuddhaLabs/PacketStorm-Exploits/blob/master/1309-exploits/pkcms-sql.txt;
    classtype:web-application-attack;
    sid:32768;
    rev:1;)
```

Take a Break!

Malicious Activity

Malicious Activity

When it comes to computer viruses, the best medicine is **prevention**. But it's hard to keep malware from getting in without knowing what it looks like.

In the next activity, you'll watch a computer get hit with a **sophisticated backdoor** that gave hackers access to machines at **Google**, **Adobe**, and other workplaces back in 2010.

Your Turn: Malicious Activity

In the next activity, you will observer malicious activity and answer the provided questions.

Activities/Stu_Malicious_Activity

Suggested Time:
30 Minutes

Malicious Activity: Operation Aurora

What is the name of the first resource the client requests from the server?

What is the server's response code for this request?

What is the value of the Location header in that response?

Malicious Activity: Operation Aurora

What is the name of the first resource the client requests from the server?

Filter for `http`.

Note that the first packet is a **`GET`** request for **`/info`**.

What is the server's response code for this request?

What is the value of the Location header in that response?

Malicious Activity: Operation Aurora

What is the name of the first resource the client requests from the server?

Filter for `http`.

Note that the first packet is a **`GET`** request for **`/info`**.

What is the server's response code for this request?

The server responds with **`302 Moved`**.

What is the value of the Location header in that response?

Malicious Activity: Operation Aurora

What is the name of the first resource the client requests from the server?

Filter for `http`.

Note that the first packet is a **`GET`** request for **`/info`**.

What is the server's response code for this request?

The server responds with **`302 Moved`**.

What is the value of the Location header in that response?

The Location header is **`/info?rFfWELUjLJHpP`**

Malicious Activity: Operation Aurora

Do you notice anything suspicious about the server's response?

Does it look anything like the HTML you've seen so far? Can you figure out anything about what the code means?

Malicious Activity: Operation Aurora

Do you notice anything suspicious about the server's response?

The query string is suspicious because it strangely resembles a command and control message.

Does it look anything like the HTML you've seen so far? Can you figure out anything about what the code means?

Malicious Activity: Operation Aurora

Do you notice anything suspicious about the server's response?

The query string is suspicious because it strangely resembles a command and control message.

Does it look anything like the HTML you've seen so far? Can you figure out anything about what the code means?

This is JavaScript code, because it's code contained in a `<script>` tag.

It's obfuscated, meaning the code is purposefully hard to read. You can tell because the first variable name is **`IwpVuiFqihVySoJStwXmT`**.
Its value looks like a hex-encoded string, but hex decoding doesn't reveal much.

Towards the very bottom, there are some regular expression replacements and a call to eval, likely intended to de-obfuscate the JavaScript and execute the result.

Malicious Activity: Operation Aurora

Immediately after receiving the HTML you just read, the browser sends a request for another file. What is the name of this file?

Malicious Activity: Operation Aurora

Immediately after receiving the HTML you just read, the browser sends a request for another file. What is the name of this file?

The GIF is named **infowTVeeGDYJWNfsrdrvXiYApnuPoCMjRrSZuKtbVgwuZCXwxKjtEclbPuJPPctcflhsttMRrSyxl.gif**.

Malicious Activity: Operation Aurora

Look at the source/destination addresses for packet 23 and packet 24. What do you notice? Explain what happened with packet 24.

Look at packet 25. What kind of data got sent?

Malicious Activity: Operation Aurora

Look at the source/destination addresses for packet 23 and packet 24. What do you notice? Explain what happened with packet 24.

Packet 23 has **`src`** and **`dst`** addresses of **`192.168.100.202`** and **`192.168.0.206`**, respectively. Packet 24 has these addresses reversed. This indicates that **`192.168.0.206`** is opening a connection to **`192.168.0.202`**.

Look at packet 25. What kind of data got sent?

Malicious Activity: Operation Aurora

Look at the source/destination addresses for packet 23 and packet 24. What do you notice? Explain what happened with packet 24.

Packet 23 has **`src`** and **`dst`** addresses of **`192.168.100.202`** and **`192.168.0.206`**, respectively. Packet 24 has these addresses reversed. This indicates that **`192.168.0.206`** is opening a connection to **`192.168.0.202`**.

Look at packet 25. What kind of data got sent?

These packets follow another TCP handshake between the client and server. If you right-click and select **`Follow->Stream`**, you'll see that a payload got sent over the wire—followed by a Windows Shell opened by the Aurora exploit!

Scheduling Backups

Your Turn: Scheduling Backups

In this exercise, you'll create **cron jobs** to create and update backups.

Activities/Stu_Scheduling_Backups

Suggested Time:
20 minutes

Scheduling Backups

Write a cron job that creates a bzipped backup of your root directory every Sunday at midnight. It should always save to `/var/backups/backup.tar.`

Write a cron job to *update* the backup of the /home directory in the `backup every day at midnight.

What *is* the correct way to create frequently updated backups?

Scheduling Backups

Write a cron job that creates a bzipped backup of your root directory every sunday at midnight. It should always save to `/var/backups/backup.tar`.

```
0 0 * * 0 cvf /var/backups/backup.tar /
```

or

```
@weekly tar cvf var/backups/backup.tar /
```

Write a cron job to *update* the backup of the /home directory every day at midnight.

What *is* the correct way to create frequently updated backups?

Scheduling Backups

Write a cron job that creates a bzipped backup of your root directory every sunday at midnight. It should always save to `/var/backups/backup.tar`.

```
0 0 * * 0 cvf /var/backups/backup.tar /
```

or

```
@weekly tar cvf var/backups/backup.tar /
```

Write a cron job to *update* the backup of the /home directory every day at midnight.

```
@daily tar --update /var/backups/backup.tar /home
```

What *is* the correct way to create frequently updated backups?

Scheduling Backups

Write a cron job that creates a bzipped backup of your root directory every sunday at midnight. It should always save to `/var/backups/backup.tar`.

```
0 0 * * 0 cvf /var/backups/backup.tar /
```

or

```
@weekly tar cvf var/backups/backup.tar /
```

Write a cron job to *update* the backup of the /home directory every day at midnight.

```
@daily tar --update /var/backups/backup.tar /home
```

What *is* the correct way to create frequently updated backups?

`--update` isn't used to create backups because it doesn't update directory content entries, and because it increases the size of the archive each time it's run. Instead, incremental backups should be used.

Any Questions?