

# **Fyshwick Medical Centre SIEM**

## **Demonstration**

### System requirements

#### Hardware Requirements

- 7<sup>th</sup> gen Intel i5 or greater ( supports Intel VT-x)
- 16gb of DDR Ram or greater
- 500GB Hard drive or greater
- A motherboard that supports Intel VT-x or equivalent

#### Software Requirements

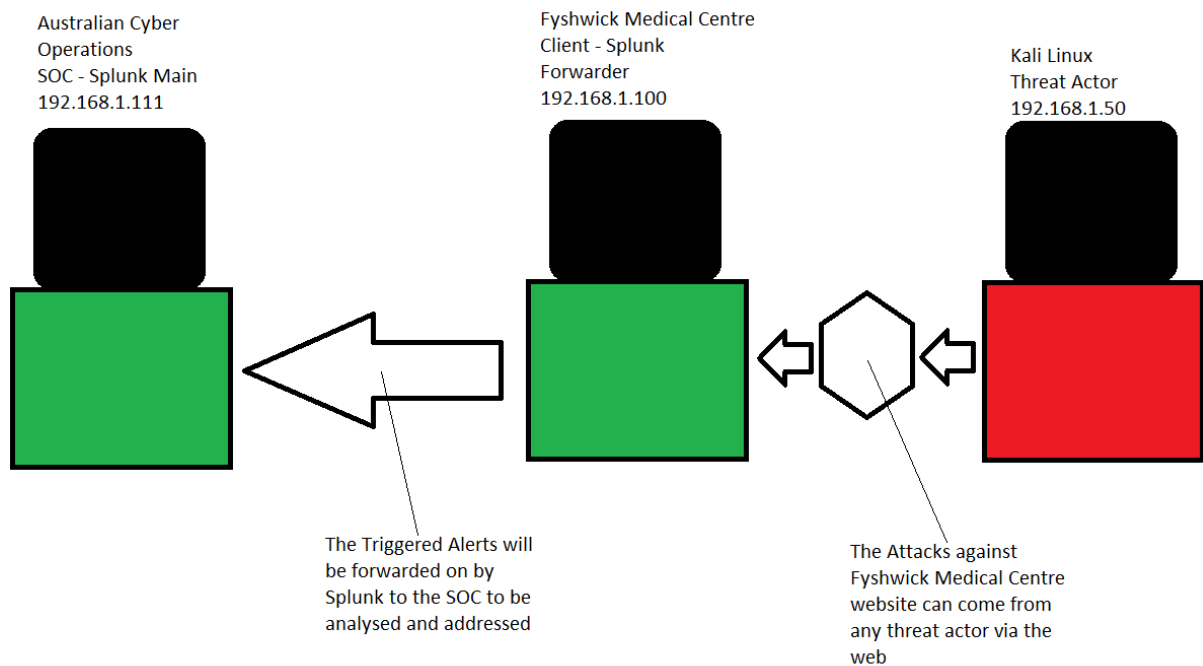
- Windows 7 or greater
- Oracle's VirtualBox
- OffSec's Kali Linux
- Splunk and relevant licensing

#### Staff Requirements

We only require 1 staff member from Cyber Operations Australia in order to demonstrate the use case scenario of the SIEM model. As previously stated requirements for real implementation can include: Australian Cyber Operation staff to install the initial project model, After this we will need at least 2 member in the SOC monitoring traffic on the network.

### Instructions/User Training

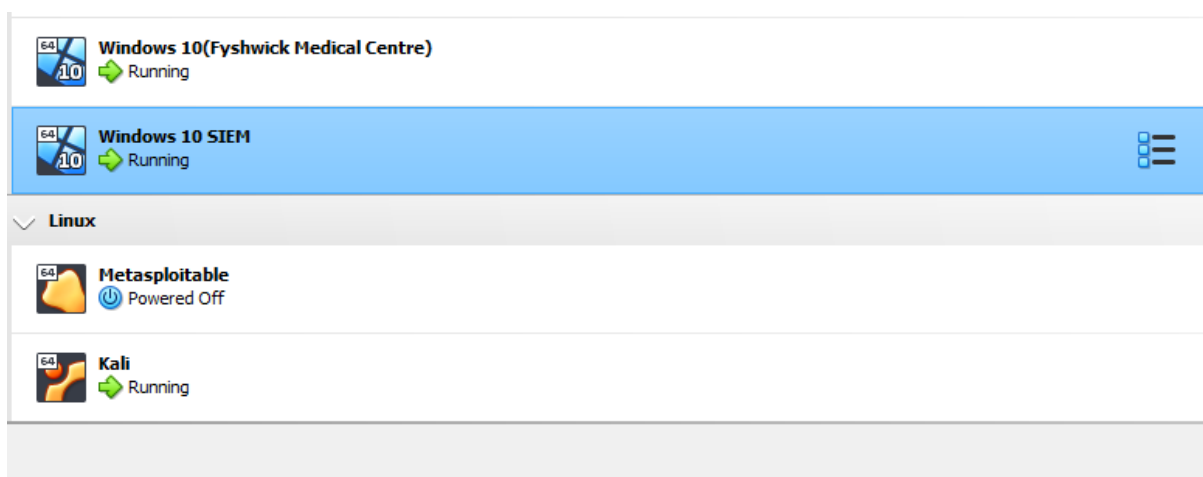
Below will contain the demonstration with information on what has been performed along with important information about that procedure intended to also be used as a training document. The following model has been designed to be educational and as user friendly to all stakeholders as possible. The Report submitted prior to this demonstration and training document contains the rationale for this use case scenario.



## Create Virtual Machines for the client, for the SIEM and the penetration testing

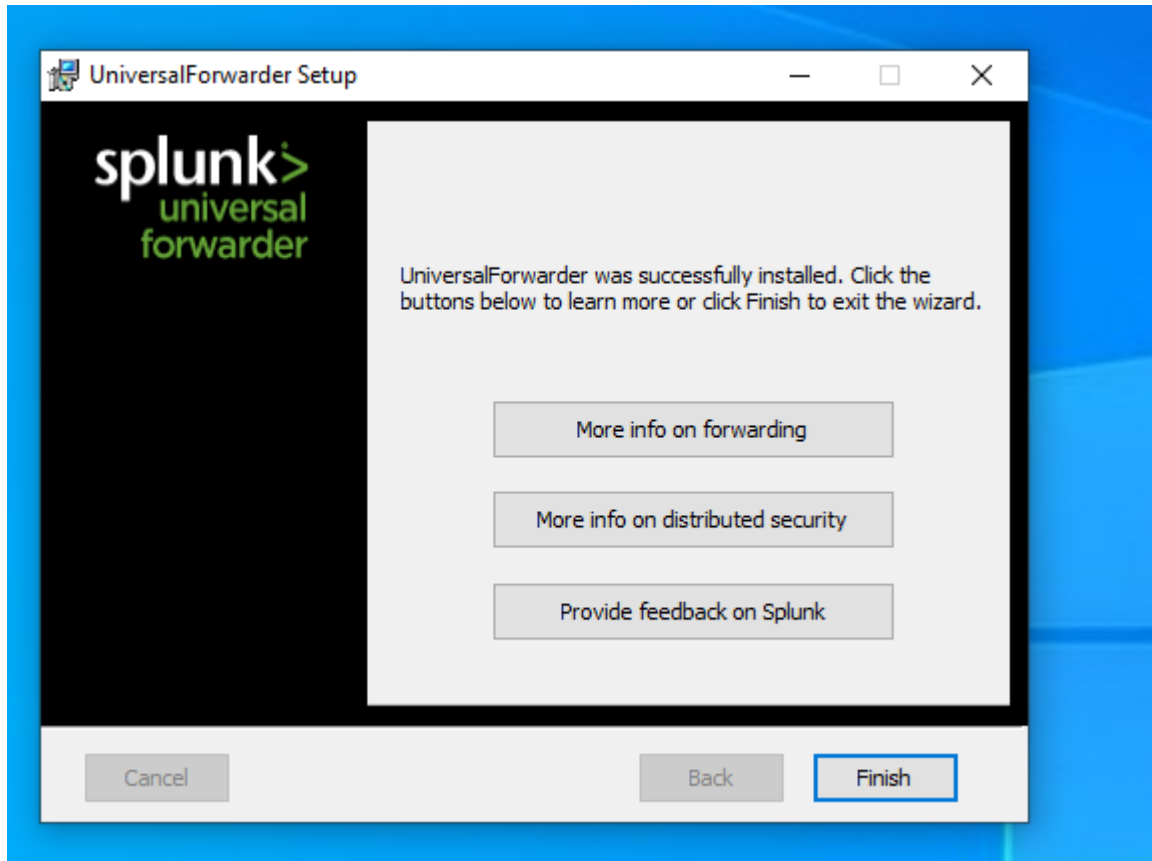
Here we have created our 3 virtual machines to be used in the scenario, two Windows 10 machines and a Kali Linux machine. Ensure all of these machines are configured with at-least 25gb hard disk space, ensure the Windows machines receive minimum 2 CPU's and 4096gb RAM, The kali machine can function with 1 CPU and 2046gb RAM

Ensure the VM's Network settings are set to "Internal" in order to minimize risk for this demonstration and enable ease of functionality. Give the IP addresses in accordance with the picture above.



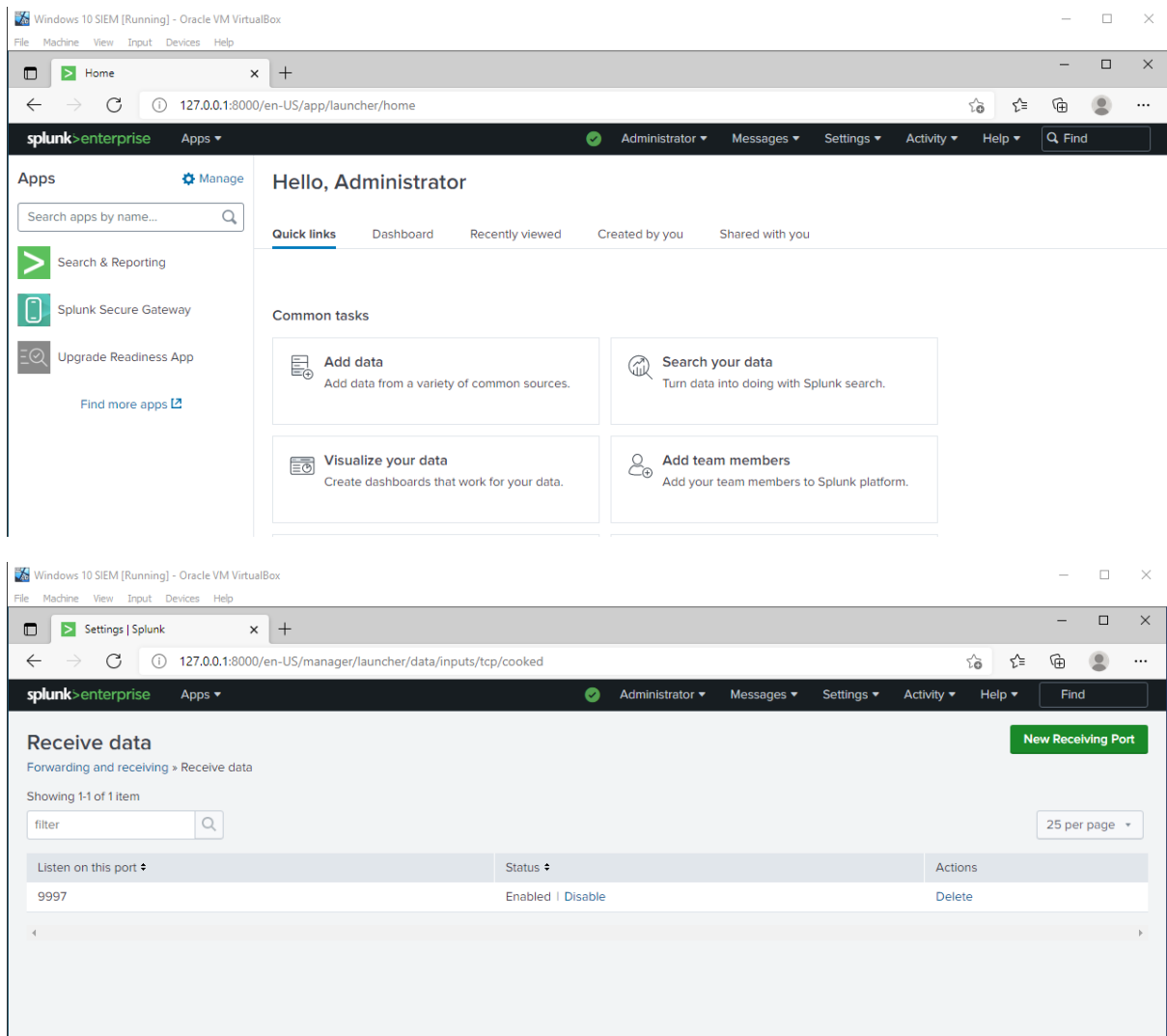
## Splunk Forwarder installation complete on Medical Centre

Ensure the Port selected for forwarding is memorized for input into the SIEM, we are using 9997 in this use case.



## Install Splunk on SIEM and ensure the forwarder port is configured

Follow the standard Splunk installation, Ensure the forwarder port in the Receive data column is configured correctly as displayed below.



## Creating Log File for Splunk

Capturing required web logs to detect SQLi and XSS we need to monitor that what is going to POST in to the webserver or backend database (db2 or user\_new). For this I have added the couple of lines of code in C:\xampp\htdocs\test1\Submit.php as follows:

```
$file = "./postlog.txt";
$msg1 = $_POST['firstname'];
file_put_contents($file, $msg1, FILE_APPEND);
$msg2 = $_POST['lastname'];
file_put_contents($file, $msg2, FILE_APPEND);
```

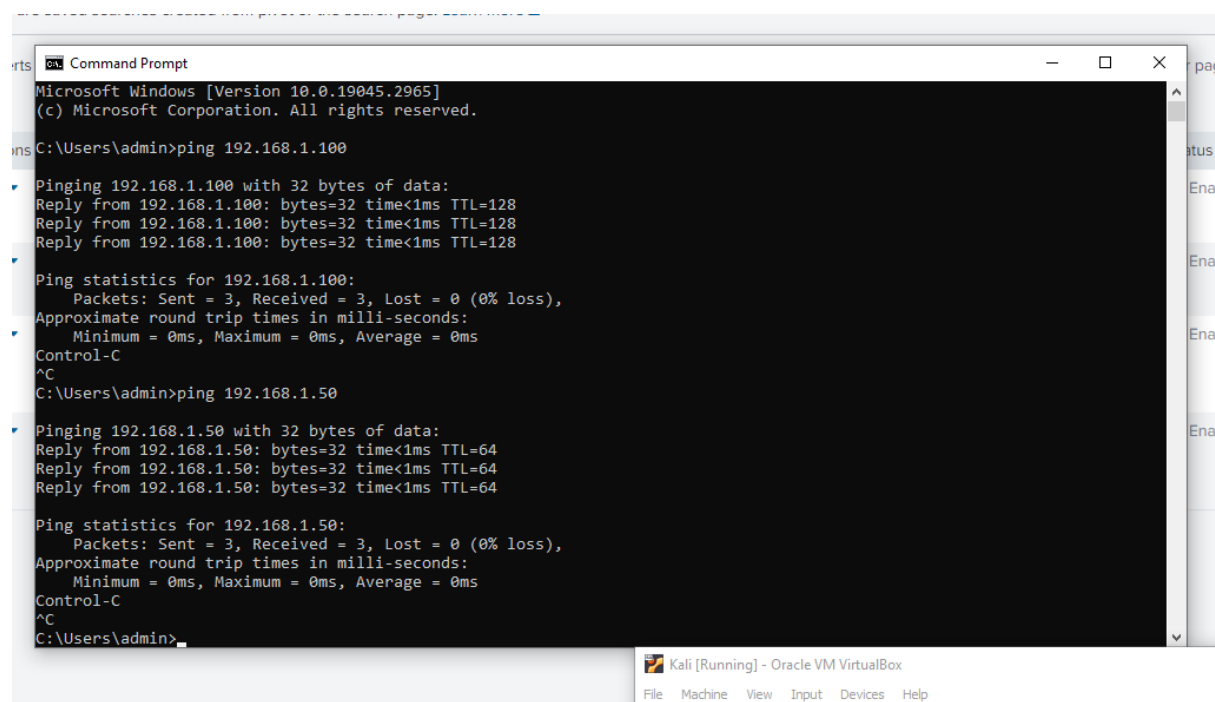
The above lines of code will create a file as C:\xampp\htdocs\test1\postlogs.txt and this file will capture the POST request to the webserver (website server/ apache web server/ backend database / db2 or user\_new). This means that whatever a user will put in the

following fields and when pressing submit, the inputted data will be captured/logged/noted in postlogs.txt.

Note that the file C:\xampp\htdocs\test1\postlogs.txt will be monitored in splunk with rules to detect SQLi or XSS attacks for our dummy website i.e., <http://localhost/test1>. Because whenever a hacker wants to inject XSS or SQLi attacks it will be first captured in postlogs.txt which will be under surveillance by splunk.

## Ensure functionality of the network

We will send ICMP packets to ensure the end points are talking to each other within the network



```
Microsoft Windows [Version 10.0.19045.2965]
(c) Microsoft Corporation. All rights reserved.

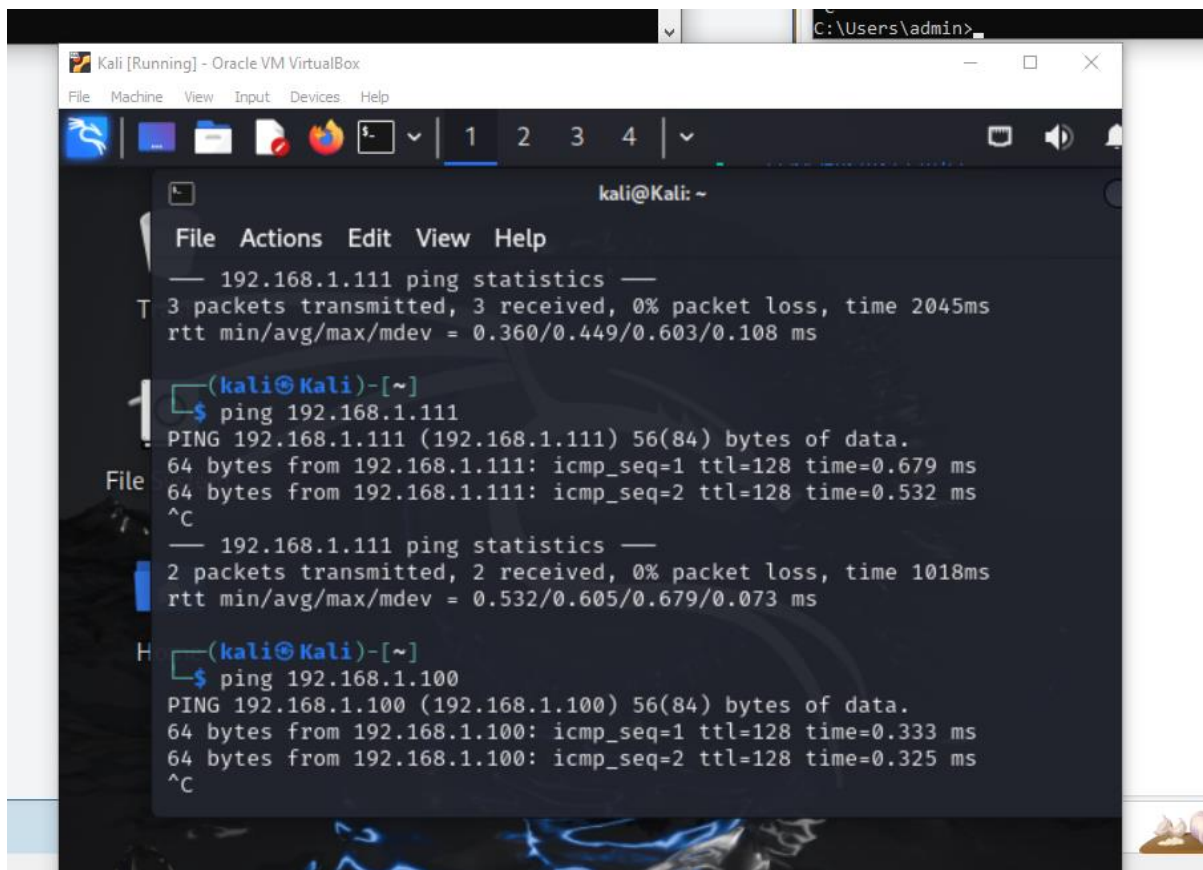
C:\Users\admin>ping 192.168.1.100

Pinging 192.168.1.100 with 32 bytes of data:
Reply from 192.168.1.100: bytes=32 time<1ms TTL=128
Reply from 192.168.1.100: bytes=32 time<1ms TTL=128
Reply from 192.168.1.100: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.100:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
^C
C:\Users\admin>ping 192.168.1.50

Pinging 192.168.1.50 with 32 bytes of data:
Reply from 192.168.1.50: bytes=32 time<1ms TTL=64
Reply from 192.168.1.50: bytes=32 time<1ms TTL=64
Reply from 192.168.1.50: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.50:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
^C
C:\Users\admin>
```



```
Submit
Command Prompt
Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
^C
C:\Users\admin>ping 192.168.1.111

Pinging 192.168.1.111 with 32 bytes of data:
Reply from 192.168.1.111: bytes=32 time<1ms TTL=128
Reply from 192.168.1.111: bytes=32 time<1ms TTL=128
Reply from 192.168.1.111: bytes=32 time<1ms TTL=128

Ping statistics for 192.168.1.111:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
^C
C:\Users\admin>ping 192.168.1.50

Pinging 192.168.1.50 with 32 bytes of data:
Reply from 192.168.1.50: bytes=32 time<1ms TTL=64
Reply from 192.168.1.50: bytes=32 time<1ms TTL=64
Reply from 192.168.1.50: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.50:
    Packets: Sent = 3, Received = 3, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms
Control-C
^C
C:\Users\admin>
```

## Create Trigger Alerts for XSS and SQLI

Here we will send dummy attacks and record the results to be configured as triggered alerts. In the real scenario we must try and anticipate the possible attacks to be used against the website/database and ensure they are set up as alerts as this is how Splunk will function effectively.

Windows 10 SIEM [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Search | Splunk 9.1.1

127.0.0.1:8000/en-US/app/search/search?q=search%20\*script\*&display.page.search.mode=smart&dispatch.sample\_ratio=1&workloa...

splunk>enterprise Apps Administrator Messages Settings Activity Help Find

Search Analytics Datasets Reports Alerts Dashboards Search & Reporting

### New Search

Save As Create Table View Close

\*script\* Last 24 hours

✓ 1 event (4/2/24 10:00:00.000 AM to 4/3/24 10:56:24.000 AM) No Event Sampling Job

Events (1) Patterns Statistics Visualization

Format Timeline Zoom Out Zoom to Selection Deselect 1 hour per column

List Format 20 Per Page

Hide Fields	All Fields	i	Time	Event
SELECTED FIELDS		>	4/3/24 10:47:36.000 AM	<script>alert(' Hacked ' )</script> host = WINDOWS-10 source = C:\xampp\htdocs\demo\postlog.txt sourcetype = postlog-too_small

1 a host 1  
a source 1

Windows 10 SIEM [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

Search | Splunk 9.1.1

127.0.0.1:8000/en-US/app/search/search?q=search%20\*\*OR\*\*&display.page.search.mode=smart&dispatch.sample\_ratio=1&workload\_p...

splunk>enterprise Apps Administrator Messages Settings Activity Help Find

Search Analytics Datasets Reports Alerts Dashboards Search & Reporting

### New Search

Save As Create Table View Close

\*OR\* Last 24 hours

✓ 4 events (4/2/24 11:00:00.000 AM to 4/3/24 11:40:05.000 AM) No Event Sampling Job

Events (4) Patterns Statistics Visualization

Format Timeline Zoom Out Zoom to Selection Deselect 1 hour per column

List Format 20 Per Page

Hide Fields	All Fields	i	Time	Event
SELECTED FIELDS		>	4/3/24 10:54:59.000 AM	' OR 1=1/* ' 'jane host = WINDOWS-10 source = C:\xampp\htdocs\demo\postlog.txt sourcetype = postlog-too_small
a host 1 a source 1 a sourcetype 1		>	4/3/24 10:54:44.000 AM	' or 1=1/* ' 'jane host = WINDOWS-10 source = C:\xampp\htdocs\demo\postlog.txt sourcetype = postlog-too_small
INTERESTING FIELDS		>	4/3/24 10:54:37.000 AM	' or 1=1/* ' 'jane host = WINDOWS-10 source = C:\xampp\htdocs\demo\postlog.txt sourcetype = postlog-too_small
a index 1 # linecount 1 a punct 2 a splunk_server 1 a timestamp 1		>	4/3/24 10:53:44.000 AM	% OR * = * host = WINDOWS-10 source = C:\xampp\htdocs\demo\postlog.txt sourcetype = postlog-too_small

+ Extract New Fields



Windows 10 SIEM [Running] - Oracle VM VirtualBox

FileMachineViewInputDevicesHelp

Search | Splunk 9.1.1

127.0.0.1:8000/en-US/app/search/search?q=search%20\*\*or%201\*\*&display.page.search.mode=smart&dispatch.sample\_ratio=1&workdo...

AdministratorMessagesSettingsActivityHelpFind

SearchAnalyticsDatasetsReportsAlertsDashboards

Search & Reporting

SQLI

SaveSave AsViewCreate Table ViewClose

\*\*or 1\*\*All time

3 events (before 4/3/24 11:42:46.000 AM)No Event SamplingJobPauseRefreshDownloadSmart Mode

Events (3)PatternsStatisticsVisualization

Format TimelineZoom OutZoom to SelectionDeselect1 second per column

ListFormat20 Per Page

< Hide Fields

All Fields

SELECTED FIELDS

a host 1

a source 1

a sourcetype 1

INTERESTING FIELDS

a index 1

# linecount 1

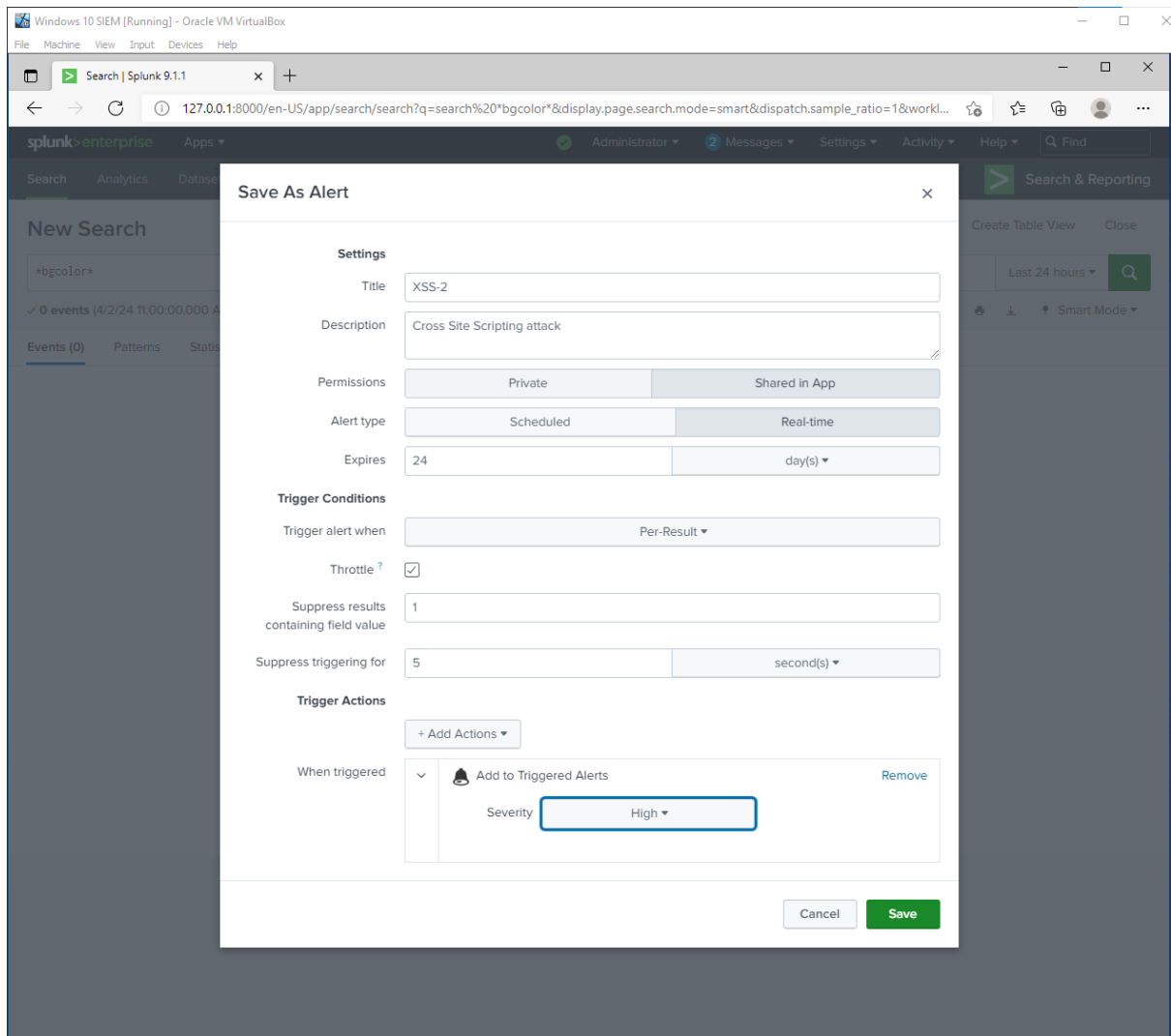
a punct 1

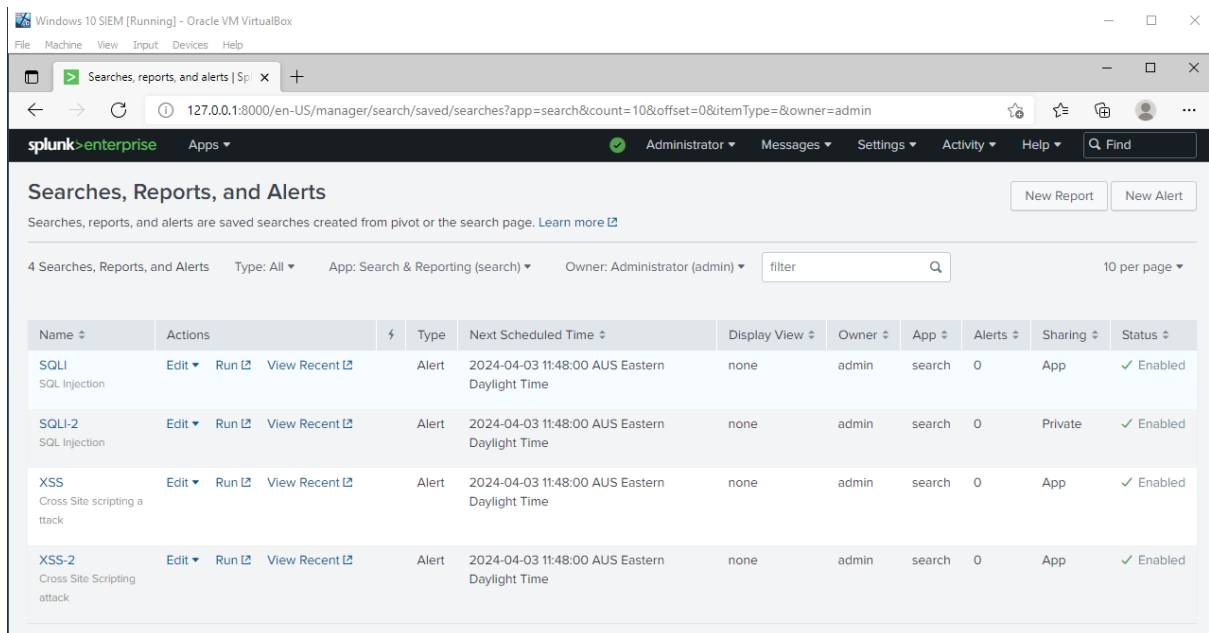
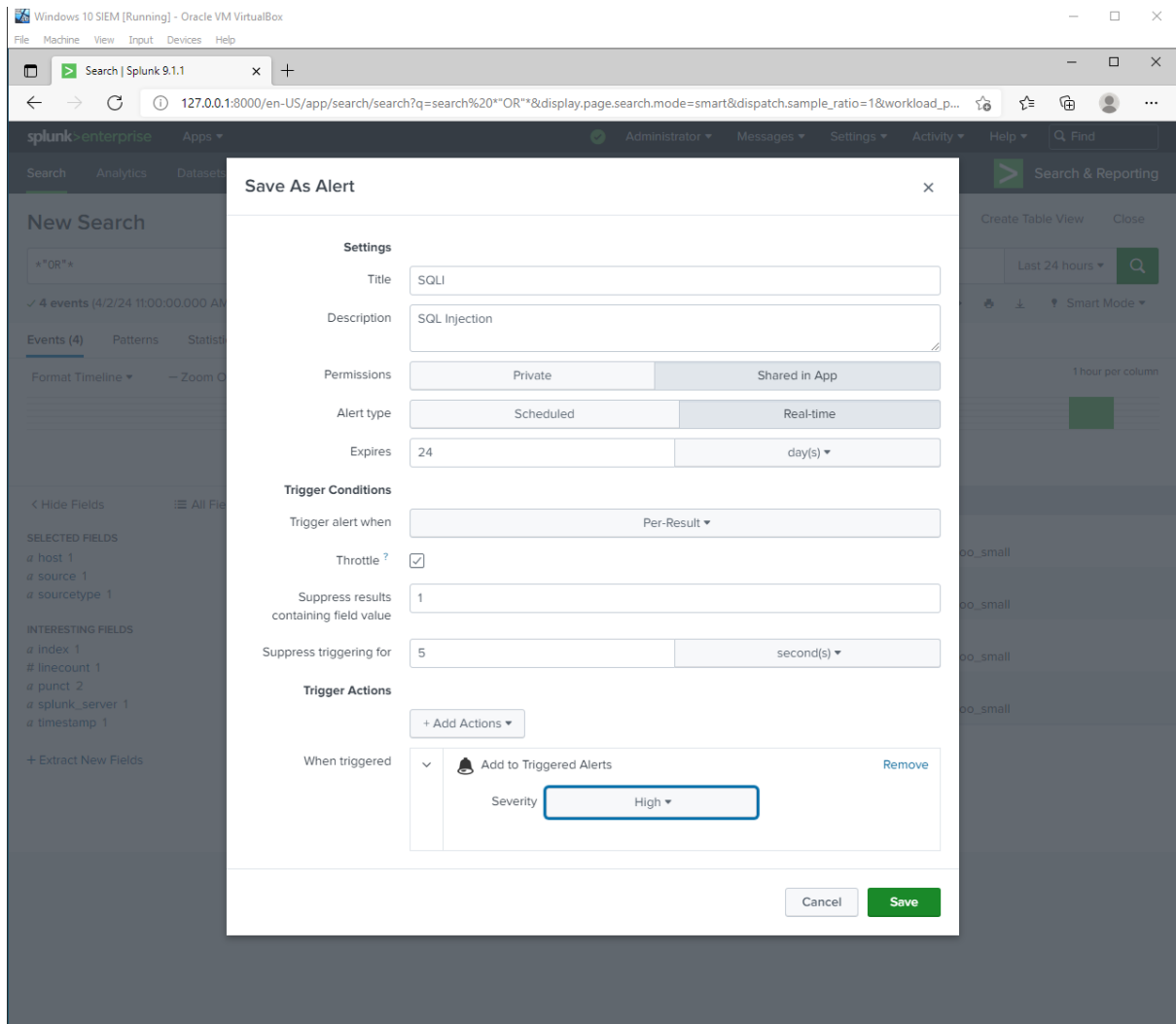
a splunk\_server 1

a timestamp 1

+ Extract New Fields

i	Time	Event
>	4/3/24 10:54:59.000 AM	<code>" OR 1=1/* " 'jane</code> host = WINDOWS-10 : source = C:\xampp\htdocs\demo\postlog.txt : sourcetype = postlog-too_small
>	4/3/24 10:54:44.000 AM	<code>" or 1=1/* " 'jane</code> host = WINDOWS-10 : source = C:\xampp\htdocs\demo\postlog.txt : sourcetype = postlog-too_small
>	4/3/24 10:54:37.000 AM	<code>" or 1=1/* " '</code> host = WINDOWS-10 : source = C:\xampp\htdocs\demo\postlog.txt : sourcetype = postlog-too_small





## Attack commands(SQLI and XSS)

Below are the commands we will run in this demonstration, Code injection attacks are a serious threat to web applications. They occur when an attacker can inject malicious code into a web application through seemingly harmless user input. This code can then be executed by the application, giving the attacker unauthorized access or control.

```
' OR 1=1/*
```

```
*OR 1*
```

```
<script>alert(Hacked )</script>
```

```
<body bgcolor='red'>
```

```
%' OR " = "
```

One common type of code injection attack is SQL injection. In this scenario, the attacker tries to manipulate the database queries used by the web application. For instance, they might inject code like (' OR 1=1 -- )into a login form. This code tricks the database into always returning a result, potentially allowing the attacker to bypass login security and access sensitive data.

Another type of code injection is Cross-Site Scripting (XSS). Here, the attacker injects malicious JavaScript code into the web page itself. If the application doesn't properly sanitize user input, this script can be executed when the page is loaded in a victim's browser. This could be something as simple as displaying an alert box saying "Hacked!", but more dangerous XSS attacks could steal cookies, session IDs, or other sensitive information from the victim.

To prevent these attacks, it's crucial for web developers to validate and sanitize all user input. This means ensuring that the data entered by users conforms to the expected format and doesn't contain any malicious code. By following secure coding practices, developers can help to protect web applications from code injection attacks.

The SIEM we have developed allows us to detect these attack methods and trigger an alert to our SOC.

# Ensure Fyshwick Medical Centre Patient Login is operational

Windows 10(Fyshwick Medical Centre) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

localhost / 127.0.0.1 / db2 / user\_ x localhost/demo/ x +

localhost/demo/

First name:  
Ben

Last name:  
Janssens

submit

Windows 10(Fyshwick Medical Centre) [Running] - Oracle VM VirtualBox

File Machine View Input Devices Help

localhost / 127.0.0.1 / db2 / user\_ x localhost/demo/submit.php x +

localhost/phpmyadmin/sql.php?server=1&db=db2&table=user\_new&pos=0

Server: 127.0.0.1 » Database: db2 » Table: user\_new

Showing rows 0 - 0 (1 total, Query took 0.0008 seconds.)

```
SELECT * FROM `user_new`
```

☐ Show all | Number of rows: 25 | Filter rows: Search this table

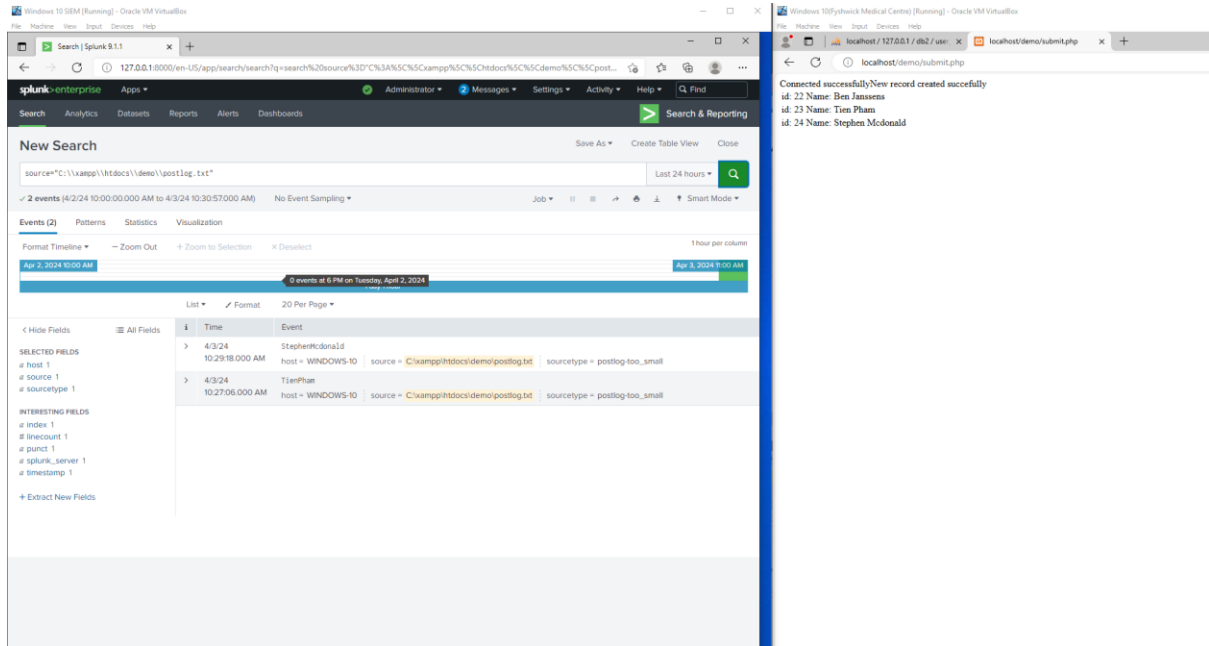
+ Options

	ID	fname	lname
<input type="checkbox"/> Edit Copy Delete	22	Ben	Janssens

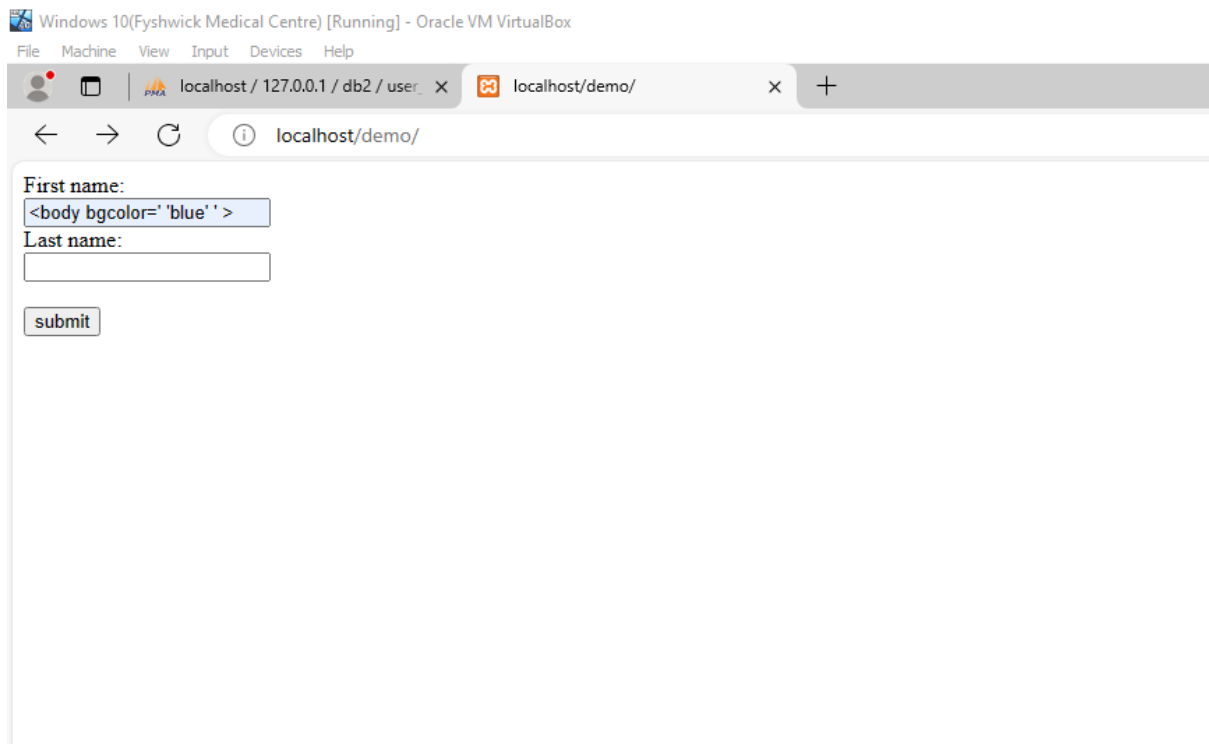
☐ Check all | With selected: Edit Copy Delete Export

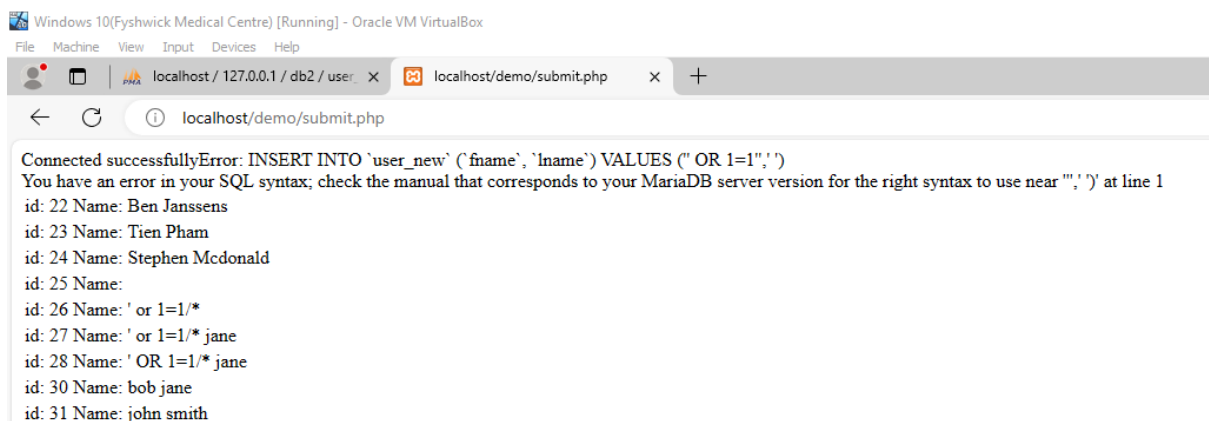
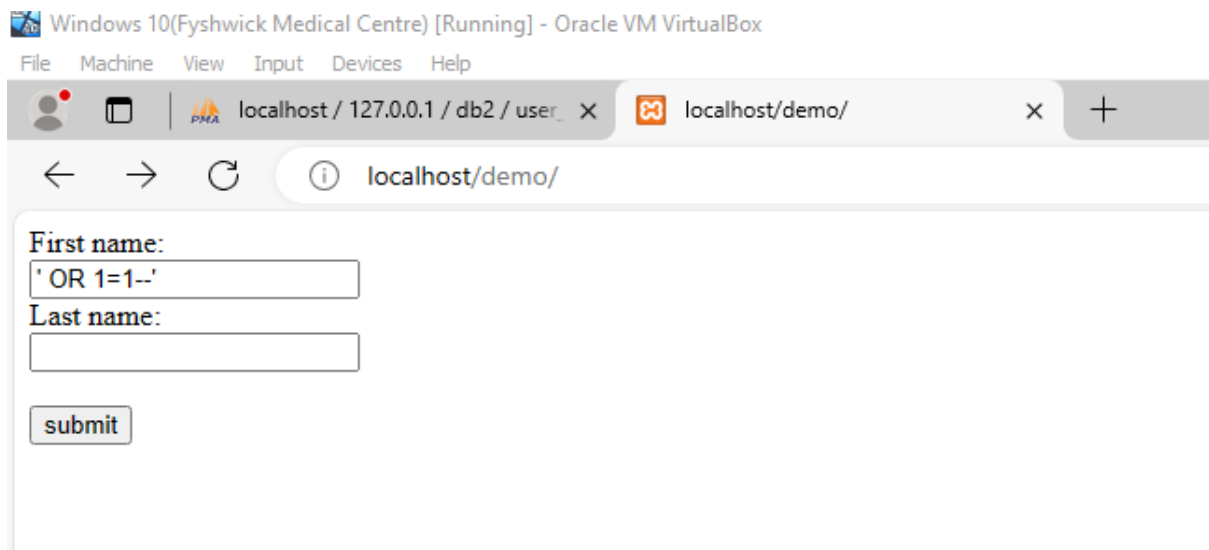
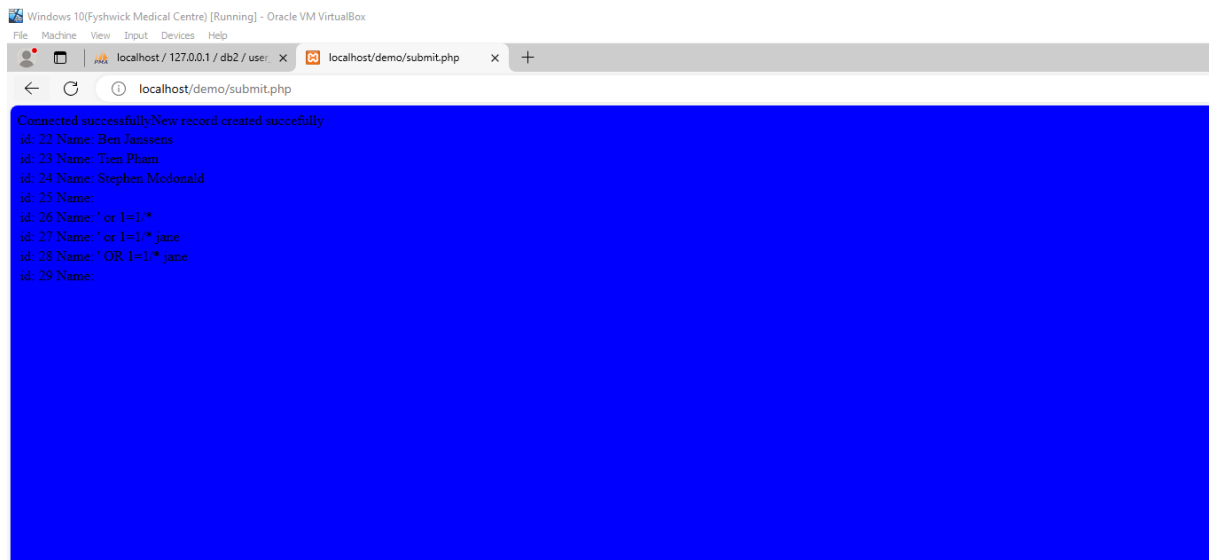
☐ Show all | Number of rows: 25 | Filter rows: Search this table

# Successful Data Capture from Forwarder



## Execute attacks against Fyshwick Medical Centre and capture the alerts





## Takeaway

This project is expected to demonstrate live in front of the stakeholders the possibilities a SIEM brings to monitoring and protecting a network in real time. As this demonstration can be performed from 1 PC with 1 staff member in a timely matter it can be used as a

highly beneficial tool in educating the principal of cyber security and the functionality of the software.

As demonstrated in this model these forms of attacks effect the functionality of the website and can be further exploited to cripple availability of the site or damage the integrity of the data held within. Confidentiality can be diminished by these forms of attacks which is crucial for the health records of the clients and the business itself.

While completely preventing cyberattacks is a difficult task, SIEM systems offer a valuable line of defence against code injection attacks. When the SIEM detects something suspicious it springs into action by generating alerts for security analysts. These alerts are crucial as they provide valuable context about the potential attack. The information might include where the attack originated from, the specific type of injection attempt, and even which systems could be potentially affected. This real-time visibility plays a vital role in incident response, the SIEM essentially becomes a command center offering a comprehensive view of security events. Analysts can use this information to understand the scope of the attack and take quicker steps to contain it, minimizing potential damage.