

## SPC-Project2-Part 2

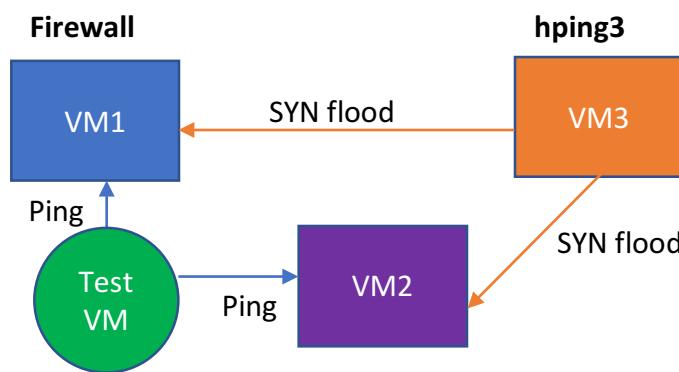
Yue Guan

### 1. Firewall

For the firewall section, because we need to launch SYN flood DoS attack as three different types: basic SYN flood, SYN flood with spoofed IP and SYN flood with SYN rule on Firewall. So, we set up firewall and architecture as this:

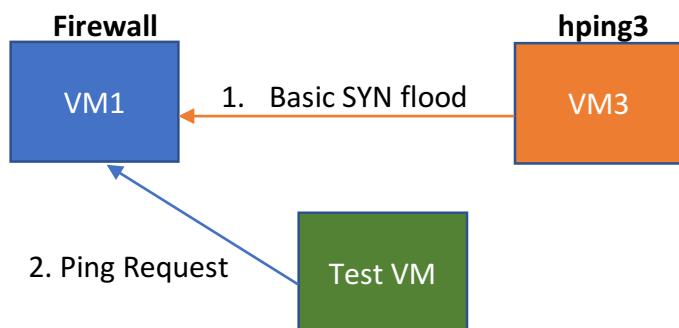
#### 1. Basic SYN flood

##### a. The very first basic test:



For the basic SYN flood attack test on the firewall, I first activated firewall on VM1 with specific rules that block the SYN request from VM3 and disabled firewall on VM2. Then I launched basic SYN flood on VM3 to attack VM1 and VM2 using hping3. With another test VM sending ping request to both, VM1 responds much faster than VM2. So, the firewall is fully configured.

##### b. Basic SYN flood as required:

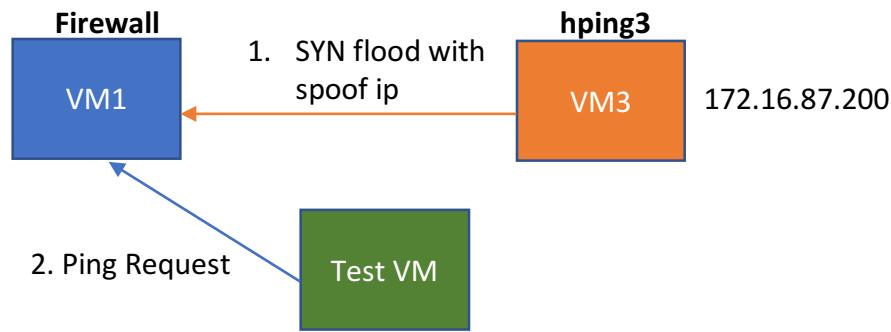


For the basic SYN flood, I use nmap to scan the open port and use one of the open port 22 to do SYN flood attack. We set up vm3 with hping3, and

configure ufw on VM1 with the rule that deny all SYN request from vm3 IP: 172.16.87.137.

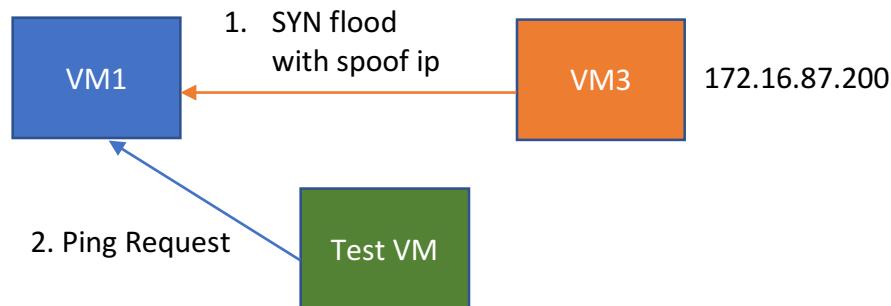
After we exploit basic attack from vm3 to vm1, we choose Test VM(vm2, IP: 172.16.87.136) to ping vm1 to test vm1's responsive capability. VM1 is still working. The pcap file captured on vm1 is *syn1basicvim.pcap*. The pcap file captured on vm3 is *syn1basicattack.pcap*.

## 2. SYN flood with spoofed IP



For the spoofing IP one, I didn't change the ufw configure on VM1. So, the ufw on VM1 is just deny on SYN from ip 172.16.87.137. Then we use a spoof ip address: 172.16.87.200 on hping3 on vm3 to do SYN flood attack on VM1. And use test VM to do ping request on vm1 to test the responsive capability here. It shows that the ping request didn't work so well now. The related pcap file *syn2attack.pcap* on VM3 and *syn2vim.pcap* on VM1.

## 3. SYN flood with SYN rule on Firewall



To fully detect SYN flood using ufw, I modify the firewall before.rules by blocking all SYN request when it receives 5 per second or more. Then we still test VM1 from VM3 using spoofing IP and hping3. With the ping test from Test VM(IP: 172.16.87.136), it shows VM1 is work great now. The related pcap files are *syn3attack.pcap*, *syn3vim.pcap*.

## IDS/IPS

Run 2 tests for each category of Pytbull (one that fails and one that passes) for each of the IDSs. Mainly we require you to run tests for the below categories:

### a. Denial of Service

#### 1. The failed DoS Screenshots:

This screenshot shows a Mozilla Firefox window titled "pytbull report - Mozilla Firefox" displaying a Pytbull report. The URL in the address bar is "127.0.0.1:8080/details/". The main content area shows two entries in a table:

#	Description	Module	Port	Payload fmt	Result
1	Dos against MSSQL	denialOfService		scapy	

Below the table, there is a "Payload:" section containing the command: `sr1(IP(dst="172.16.87.136")/TCP(dport=1433)/*"1000, verbose=0)`. There is also an "Alerts:" section which is currently empty.

This screenshot shows a Mozilla Firefox window titled "pytbull report - Mozilla Firefox" displaying a Pytbull report. The URL in the address bar is "127.0.0.1:8080/details/". The main content area shows two entries in a table:

#	Description	Module	Port	Payload fmt	Result
1	ApacheBench DoS	denialOfService		command	

Below the table, there is a "Payload:" section containing the command: `/usr/bin/ab -k -c 25 -n 10000 http://%target%`. There is also an "Alerts:" section which is currently empty.

The reason why I failed this time: I didn't set up correct snort rules to detect those DoS attack. At the meantime, there is some configuration mistakes I have on snort too.

## 2. The Success One:

The corresponding alert logs file looks like this:

pytbull report - Mozilla Firefox

127.0.0.1:8080/details/ 67% Search

**putbull**

Stats Details Search

#	Description	Module	Port	Payload fmt	Result
1	Dos against MSSQL	denialOfService		scapy	

**Start:** 2017-11-03 03:36:51.608941  
**End:** 2017-11-03 03:36:53.345513  
**Sig match:** None

**Payload:**  
`sr1(IP(dst="172.16.87.136")/TCP(dport=1433)/*"+1000, verbose=0)`

**Alerts:**

```
11/03/17 04:01:889796 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.137:55864 -> 172.16.87.136:50882
11/03/00:40:01:890198 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.137:55864 -> 172.16.87.136:50882
11/03/08:49:91:890233 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.137:53516 -> 172.16.87.136:21
11/03/08:49:91:891500 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.137:55864 -> 172.16.87.136:50882
11/03/08:49:91:892433 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.137:53516 -> 172.16.87.136:21
11/03/08:49:91:892450 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.137:53516 -> 172.16.87.136:21
11/03/08:49:91:893986 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.137:53516 -> 172.16.87.136:21
```

#	Description	Module	Port	Payload fmt	Result
2	ApacheBench DoS	denialOfService		command	
3	hping SYN flood	denialOfService		command	

pytbull report - Mozilla Firefox

pytbull report

127.0.0.1:8080/details/

67% Search

Alerts:

11/03-08:48:05.635615 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.137:53520 ->
11/03-08:48:05.636223 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.137:53520 ->
11/03-08:48:05.636901 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.137:53520 ->
11/03-08:48:05.639921 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.137:53520 ->
11/03-08:48:05.640561 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.137:53520 ->
11/03-08:48:05.668501 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.137:53520 ->
11/03-08:48:05.669284 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.137:53520 ->
11/03-08:48:05.675551 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.137:53520 ->

Start: 2017-11-03 03:37:16.497230  
End: 2017-11-03 03:37:18.194623  
Sig match: None

Payload:  
/usr/bin/sudo /usr/sbin/hping3 172.16.87.136 -S --faster -p 80 -I ens33 -c 50000 -a 1.2.3.4

Alerts:

11/03-08:40:24.770713 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.137:35510 ->
11/03-08:40:24.770808 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.136:35510 ->
11/03-08:40:24.771012 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.136:35510 ->
11/03-08:40:24.771315 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.136:35510 ->
11/03-08:40:24.771412 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.136:35510 ->
11/03-08:40:24.771621 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.136:35510 ->
11/03-08:40:24.771737 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.136:35510 ->
11/03-08:40:24.771759 [**] [1:10001:1] DoS attack [**] [Priority: 0] {TCP} 172.16.87.137:35510 ->

## b. Brute Force

### 1. The failed one:

The screenshot shows a Firefox browser window titled "pytbull report - Mozilla Firefox". The address bar shows the URL "127.0.0.1:8080/details/". The main content area displays the "pytbull" interface. At the top, there are three tabs: "Stats", "Details", and "Search". Below them is a table with the following data:

#	Description	Module	Port	Payload fmt	Result
1	Bruteforce against FTP with ncrack	bruteForce		command	

Below the table, there is a section labeled "Payload:" containing the command: "/usr/local/bin/ncrack -f -U data/ncrack-users.txt -P data/ncrack-passwords.txt 172.16.87.136:21". There is also a section labeled "Alerts:" which is currently empty.

At the bottom of the interface, it says "pytbull is developed and maintained by Sébastien Damayet" and provides links to "pytbull.sf.net" and "aldeid.com".

The reason why I failed: I set up some snort rules that just detect the content where has ftp, however, it should be has some content with 530 (not found).

The Successful one with correspond alert log files:

pytbull report - Mozilla Firefox

SNORT RULE TO DETECT [SOLVED] snort rule for [REDACTED]

pytbull report

127.0.0.1:8080/details/ 67% 4:19 AM

**pytbull**

Stats Details Search

	#	Description	Module	Port	Payload fmt	Result
<input type="checkbox"/>	1	Bruteforce against FTP with ncrack	bruteForce		command	<input type="button"/> <input type="button"/>

**Start:** 2017-11-03 04:18:11.139926  
**End:** 2017-11-03 04:18:32.178376  
**Sig match:** (?i)brute

**Payload:**  
/usr/local/bin/ncrack -f -U data/ncrack-users.txt -P data/ncrack-passwords.txt 172.16.87.136:21

**Alerts:**

```
11/03-01:21:21.235733 [**] [1:1000001:1] FTP login incorrect [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 172.16.87.137:54804 -> 172.16.87.136:21
11/03-01:21:21.236225 [**] [1:1000001:1] FTP login incorrect [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 172.16.87.137:54804 -> 172.16.87.136:21
11/03-01:21:21.236267 [**] [1:1000001:1] FTP login incorrect [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 172.16.87.137:54804 -> 172.16.87.136:21
11/03-01:21:21.266129 [**] [1:1000001:1] FTP login incorrect [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 172.16.87.137:54804 -> 172.16.87.136:21
11/03-01:21:21.266176 [**] [1:1000001:1] FTP login incorrect [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 172.16.87.137:54804 -> 172.16.87.136:21
11/03-01:21:21.305460 [**] [1:1000001:1] FTP login incorrect [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 172.16.87.137:54810 -> 172.16.87.136:21
11/03-01:21:21.305841 [**] [1:1000001:1] FTP login incorrect [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 172.16.87.137:54810 -> 172.16.87.136:21
```

pytbull is developed and maintained by Sébastien Damayé  
pytbull.st.net | aldeid.com

### c. Evasion Techniques:

#### 1. The failed one:

The screenshot shows a Mozilla Firefox window titled "pytbull report - Mozilla Firefox". The address bar displays "127.0.0.1:8080/details/". The main content area is a table titled "Details" showing three rows of data:

#	Description	Module	Port	Payload fmt	Result
1	Nmap decoy test (6th position)	evasionTechniques		command	
2	Nmap decoy test (7th position)	evasionTechniques		command	
3	Hex encoding	evasionTechniques	80/tcp	socket	

Below the table, there is a "Payload:" section containing the following text:

```
• Start: 2017-11-03 04:46:53.547674
• End: 2017-11-03 04:46:53.549220
• Sig match: 1:1122:8

Payload:
GET /index.php?page=%2e%2f%2e%2e%2f%2e%2f%65%74%63%2f%70%61%73%73%77%64 HTTP/1.1
Host: 127.0.0.1
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.1; en-US; rv:1.7.5) Gecko/20041202 Firefox/1.0
```

Under the "Payload:" section, there is an "Alerts:" section which is currently empty.

The reason why I failed: The reason why I failed for Hex encoding here is because I didn't detect the specific HTTP things in my snort rule. I didn't set the rule properly.

## 2. The successful one with alert files:

Two screenshots of a Firefox browser window showing Pytbull report details for two different SNORT rules.

**Screenshot 1 (Top):**

The browser title is "pytbull report - Mozilla Firefox". The address bar shows "127.0.0.1:8080/details/". The page displays a table of 10 evasion techniques:

#	Description	Module	Port	Payload fmt	Result
1	Nmap decoy test (6th position)	evasionTechniques		command	[green circle]
2	Nmap decoy test (7th position)	evasionTechniques		command	[green circle]
3	Hex encoding	evasionTechniques	80/tcp	socket	[yellow circle]
4	Nmap scan with fragmentation	evasionTechniques		command	[yellow circle]
5	Nikto Random URI encoding	evasionTechniques		command	
6	Nikto Directory self reference	evasionTechniques		command	
7	Nikto Premature URL ending	evasionTechniques		command	
8	Nikto Prepend long random string	evasionTechniques		command	
9	Nikto Fake parameter	evasionTechniques		command	
10	Nikto TAB as request spacer	evasionTechniques		command	

**Screenshot 2 (Bottom):**

The browser title is "pytbull report - Mozilla Firefox". The address bar shows "127.0.0.1:8080/details/". The page displays two rows of details for SNORT rules:

- Row 1 (Top):**
  - Start:** 2017-11-03 04:42:46.383961
  - End:** 2017-11-03 04:43.09.162286
  - Sig match:** 172.16.87.136
  - Payload:**

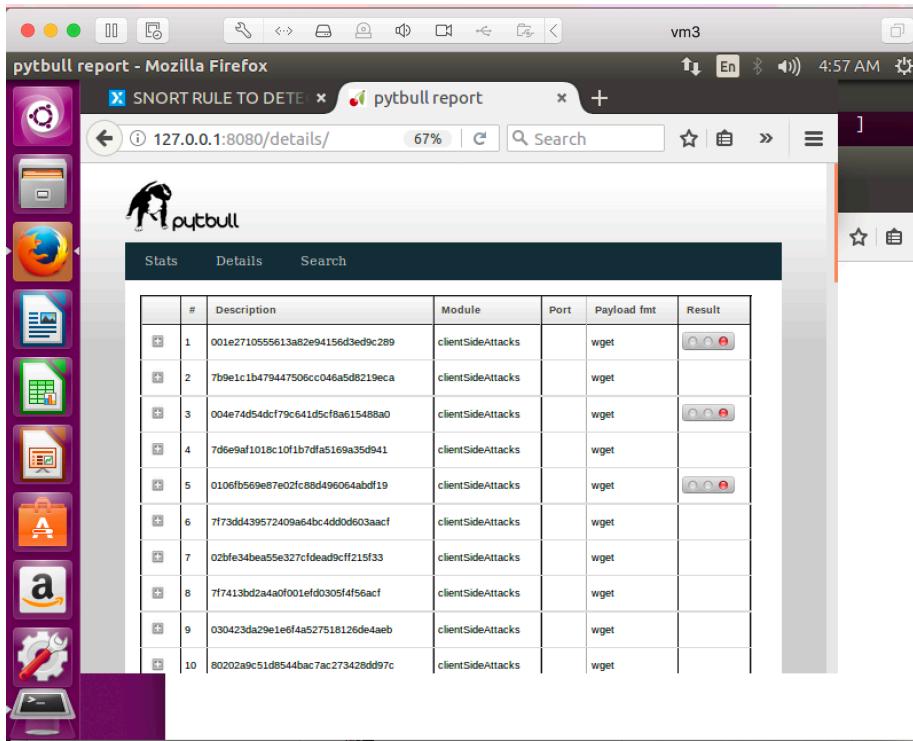
```
/usr/bin/sudo /usr/bin/nmap -s5 -A -D
192.168.100.1,192.168.100.2,192.168.100.3,192.168.100.4,192.168.100.5,ME 172.16.87.136
```
  - Alerts:**

```
11/03/03:49:52.314838 [**] [1:108001:1] Evasion detected [**] [Priority: 0] {TCP}
172.16.87.137:48342 -> 172.16.87.136:22
11/03/03:49:52.314838 [**] [1:108001:1] Evasion detected [**] [Priority: 0] {TCP}
172.16.87.137:48342 -> 172.16.87.136:22
11/03/03:49:52.314488 [**] [1:108001:1] Evasion detected [**] [Priority: 0] {TCP}
172.16.87.137:426880 -> 172.16.87.136:88
11/03/03:49:52.315145 [**] [1:108001:1] Evasion detected [**] [Priority: 0] {TCP}
172.16.87.137:426880 -> 172.16.87.136:88
11/03/03:49:52.315158 [**] [1:108001:1] Evasion detected [**] [Priority: 0] {TCP}
172.16.87.137:426880 -> 172.16.87.136:88
11/03/03:49:52.333948 [**] [1:108001:1] Evasion detected [**] [Priority: 0] {TCP}
172.16.87.137:469666 -> 172.16.87.136:21
11/03/03:49:52.334179 [**] [1:108001:1] Evasion detected [**] [Priority: 0] {TCP}
172.16.87.137:469666 -> 172.16.87.136:21
```
- Row 2 (Bottom):**
  - Start:** 2017-11-03 04:43:13.256061
  - End:** 2017-11-03 04:43:28.503333
  - Sig match:** 172.16.87.136
  - Payload:**

```
/usr/bin/sudo /usr/bin/nmap -s5 -A -D
192.168.100.1,192.168.100.2,192.168.100.3,192.168.100.4,192.168.100.5,192.168.100.6,ME 172.16.87.136
```

## d. Client Side Attacks

### 1. The failed one:



A screenshot of a Mozilla Firefox browser window titled "pytbull report - Mozilla Firefox". The address bar shows "127.0.0.1:8080/details/". The main content area displays a table titled "pytbull report" with 10 rows of data. The columns are labeled "#", "Description", "Module", "Port", "Payload fmt", and "Result". All rows show "clientSideAttacks" in the "Module" column and "wget" in the "Payload fmt" column. The "Result" column contains icons for each row, all of which are red with a diagonal slash, indicating failure. The table has a dark header and light gray rows.

#	Description	Module	Port	Payload fmt	Result
1	001e2710555613a82e94156d3ed9c289	clientSideAttacks		wget	
2	7b9e1c1b479447506cc046a5db219eca	clientSideAttacks		wget	
3	004e74d54dcf79c641d5cf8a615488a0	clientSideAttacks		wget	
4	7d6e9af1018c10fb7dfa5169a35df941	clientSideAttacks		wget	
5	0106fb569e87e02fc88d496064abdf19	clientSideAttacks		wget	
6	7f73dd439572409a64bc4dd0d603aacf	clientSideAttacks		wget	
7	02bfe34bea55e327cf0ead9cff215f53	clientSideAttacks		wget	
8	7f7413bd2a4a0f001efd0305f4f5faeb	clientSideAttacks		wget	
9	030423da29e1e6f4a527518126de4aeb	clientSideAttacks		wget	
10	80202a9c51d854bac7ac273428dd97c	clientSideAttacks		wget	

The reason why I failed: I didn't set up a snort rule related to the client side attack. It all relates to something else.

## 2. The successful one with alert logs:

The screenshot shows a Firefox browser window titled "pytbull report - Mozilla Firefox". The address bar displays "127.0.0.1:8080/details/". The main content area is a table with the following columns: #, Description, Module, Port, Payload fmt, and Result.

#	Description	Module	Port	Payload fmt	Result
1	001e2710555613a82e94156d3ed9c289	clientSideAttacks		wget	[button]
2	7b9e1c1b479447506cc046a5d8219eca	clientSideAttacks		wget	[button]
3	004e74d54dcf79c641d5cf8a615488a0	clientSideAttacks		wget	[button]
4	7d6e9af1018c10f1b7dfa5169a35d941	clientSideAttacks		wget	[button]

Below the table, there is a summary section with the following information:

- Start: 2017-11-03 04:58:49.266701
- End: 2017-11-03 04:58:51.655972
- Sig match: None

The Alerts section contains a list of log entries:

```
11/03-04:05:55.191964 [**] [1:100001:1] Client-side Attack [**] [Priority: 0] {TCP}
172.16.87.137:47860 -> 172.16.87.136:21
11/03-04:05:55.192604 [**] [1:100001:1] Client-side Attack [**] [Priority: 0] {TCP}
172.16.87.137:47860 -> 172.16.87.136:21
11/03-04:05:55.196173 [**] [1:100001:1] Client-side Attack [**] [Priority: 0] {TCP}
172.16.87.137:47860 -> 172.16.87.136:21
11/03-04:05:55.196459 [**] [1:100001:1] Client-side Attack [**] [Priority: 0] {TCP}
172.16.87.137:47860 -> 172.16.87.136:21
11/03-04:05:55.197124 [**] [1:100001:1] Client-side Attack [**] [Priority: 0] {TCP}
172.16.87.137:47860 -> 172.16.87.136:21
11/03-04:05:55.225333 [**] [1:100001:1] Client-side Attack [**] [Priority: 0] {TCP}
172.16.87.137:47860 -> 172.16.87.136:21
11/03-04:05:55.225777 [**] [1:100001:1] Client-side Attack [**] [Priority: 0] {TCP}
```

At the bottom of the table, there is another summary section:

- Start: 2017-11-03 04:58:51.658266

## e. SIEM

### 1. Some Screenshot on SIEM:

This screenshot shows the 'Scan & Add Assets' page of the AlienVault OSSIM web interface. On the left, there's a sidebar with icons for Network Interfaces, Asset Discovery, Deploy HIDS, Log Management, and Join OTX. The main area has a heading 'Scan & Add Assets' with instructions on how to add assets. It includes a 'Add Asset Manually' form with fields for Hostname and Asset Type, and buttons for 'SCAN NETWORKS', 'IMPORT FROM CS', and '+ ADD'. Below this is a table listing assets with columns for HOSTNAME, IP, and TYPE. The table contains entries for 'AttackMachine', 'Firewall', and several hosts with IP addresses 172.16.87.1 through 172.16.87.139, all categorized as Linux.

This screenshot shows the main dashboard of the AlienVault OSSIM system. The top navigation bar includes links for DASHBOARDS, ANALYSIS, ENVIRONMENT, REPORTS, and CONFIGURATION. Below the navigation is a menu with EXECUTIVE, TICKETS, SECURITY, TAXONOMY, and VULNERABILITIES. The dashboard features several widgets: 'TOP 10 PROMISCUOUS HOSTS' (No data available yet), 'SECURITY EVENTS: TOP 5 ALARMS' (No data available yet), 'SECURITY EVENTS: TOP 5 EVENTS' (A bar chart showing event counts: 184, 61, 61, 51, 51), 'TOP 10 HOSTS WITH MULTIPLE EVENTS' (No data available yet), 'SECURITY EVENTS TREND: LAST DAY' (A line graph showing a single sharp peak around 26 Oct at 11h 12m), and 'SECURITY EVENTS TREND: LAST WEEK' (A line graph showing two peaks, one on 26 Oct and another on 31 Oct).



vm1

Ubuntu 64-bit

Add a note here

## 2. The reason why I failed on OSSIM:

However, the OSSIM part didn't succeed this time. Fortunately, professor Ayo understand us about the situation on OSSIM that we faced this time and let us to write the reason why we failed this time on OSSIM and will still give us scores on that. It's great.

So, the reason why I failed OSSIM here is I met with some trouble to integrate snort into my OSSIM. Also, there is some issue with my own computer here because of the large plenty amount of VMs I need to open at the same time. My laptop failed for hundreds of time on opening 4 or more VMs at the same time. In addition, failed thousands of time to integrate snort into OSSIM. The screenshot shows where I was going through, and I look forward to seeing where it should be. It would be great if you could show us on class about how OSSIM should work and integrate all the security devices together in one platform.