Singapore Institute of Technology
Bachelor of Information and Communications Technology
(Honours)

Team Project Report - Additional Security Features

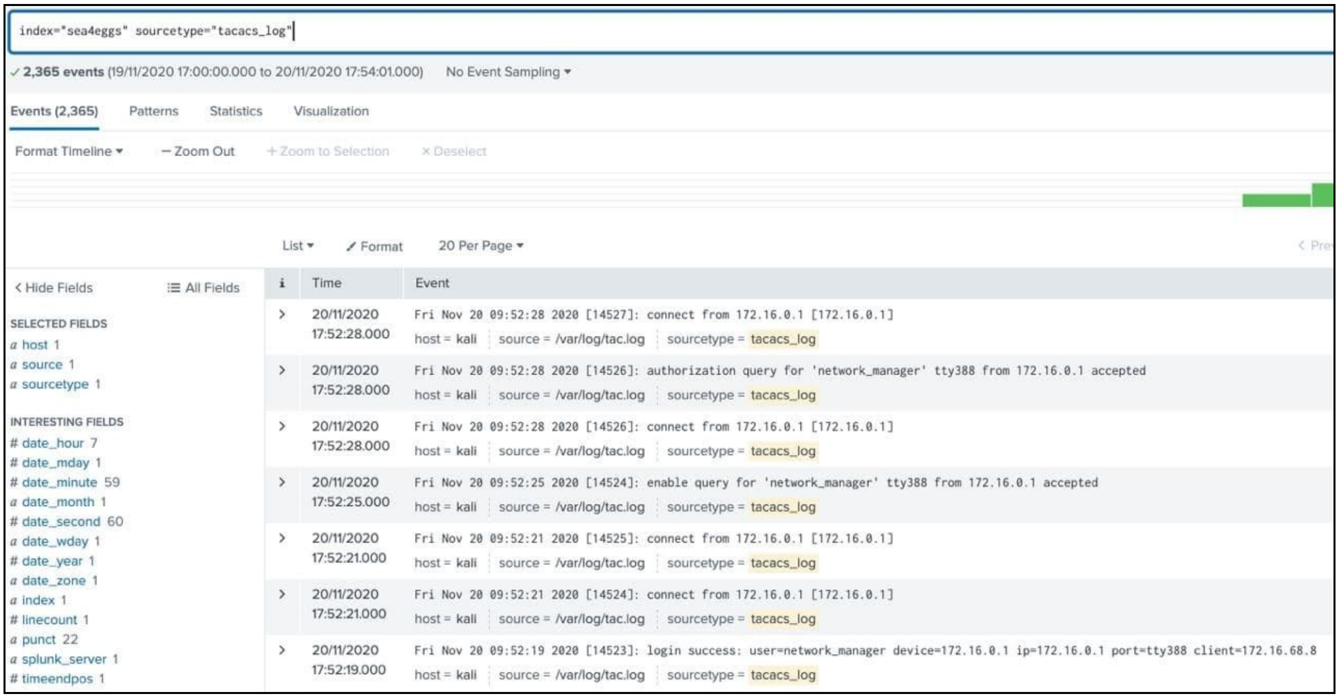
Module Code:	ICT2203
Module Title:	Network Security
Submitted to:	Dr. Purnima Murali Mohan
Date of submission:	22 November 2020

Group Members & Student ID:

Gerald Peh Wei Xiang	1802959
Lim Zhao Xiang	1802976
Ho Xiu Qi	1802962
Tan Zhao Yea	1802992
Chin Clement	1802951

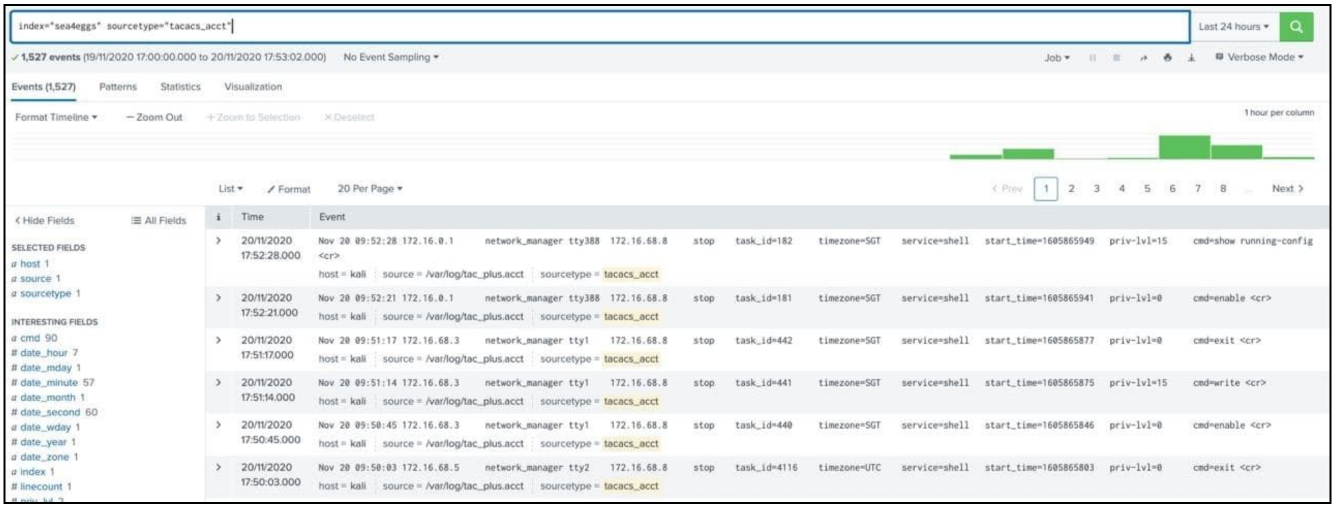
Security Information and Event Management (SIEM)

A SIEM will complement the team's efforts to monitor logs from a central location and track any potential security incidents. Early detection of attacks is crucial to mitigate the resulting impact, and the SIEM allows us to easily review and correlate events and logs from different sources to form an incident timeline. The team has implemented **Splunk** as the SIEM solution, as well as the network's Syslog server and NetFlow collector.



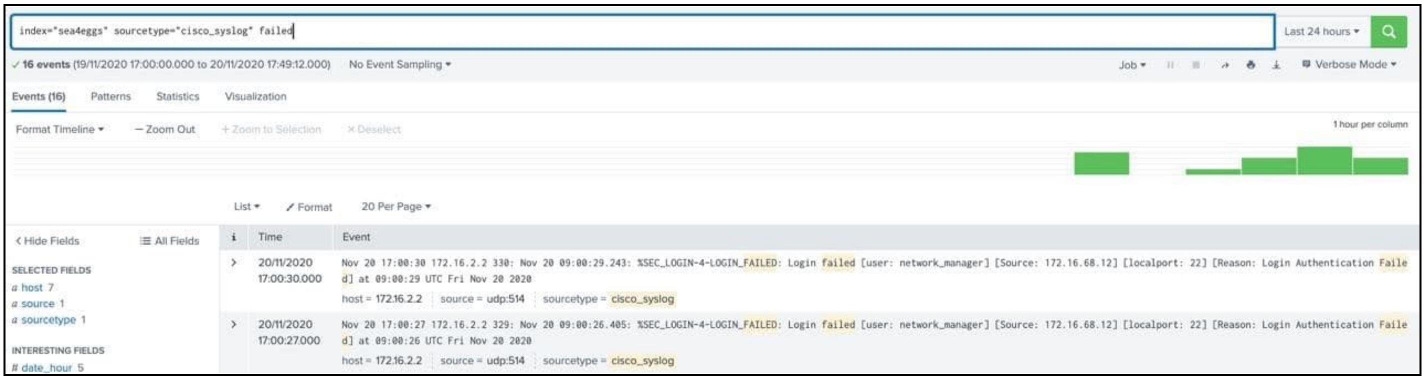
```
index="sea4eggs" sourcetype="tacacs_log"
✓ 2,365 events (19/11/2020 17:00:00.000 to 20/11/2020 17:54:01.000)   No Event Sampling

i  Time                    Event
>  20/11/2020 17:52:28.000 Fri Nov 20 09:52:28 2020 [14527]: connect from 172.16.0.1 [172.16.0.1]
                           host = kali   source = /var/log/tac.log   sourcetype = tacacs_log
>  20/11/2020 17:52:28.000 Fri Nov 20 09:52:28 2020 [14526]: authorization query for 'network_manager' tty388 from 172.16.0.1 accepted
                           host = kali   source = /var/log/tac.log   sourcetype = tacacs_log
>  20/11/2020 17:52:28.000 Fri Nov 20 09:52:28 2020 [14526]: connect from 172.16.0.1 [172.16.0.1]
                           host = kali   source = /var/log/tac.log   sourcetype = tacacs_log
>  20/11/2020 17:52:25.000 Fri Nov 20 09:52:25 2020 [14524]: enable query for 'network_manager' tty388 from 172.16.0.1 accepted
                           host = kali   source = /var/log/tac.log   sourcetype = tacacs_log
>  20/11/2020 17:52:21.000 Fri Nov 20 09:52:21 2020 [14525]: connect from 172.16.0.1 [172.16.0.1]
                           host = kali   source = /var/log/tac.log   sourcetype = tacacs_log
>  20/11/2020 17:52:21.000 Fri Nov 20 09:52:21 2020 [14524]: connect from 172.16.0.1 [172.16.0.1]
                           host = kali   source = /var/log/tac.log   sourcetype = tacacs_log
>  20/11/2020 17:52:19.000 Fri Nov 20 09:52:19 2020 [14523]: login success: user=network_manager device=172.16.0.1 ip=172.16.0.1 port=tty388 client=172.16.68.8
                           host = kali   source = /var/log/tac.log   sourcetype = tacacs_log
```

The filter <u>sourcetype="tacacs_log"</u> will show us information about the authentication phase of network devices configured to use AAA (All network devices use AAA in our setup).



```
index="sea4eggs" sourcetype="tacacs_acct"
✓ 1,527 events (19/11/2020 17:00:00.000 to 20/11/2020 17:53:02.000)   No Event Sampling

i  Time                    Event
>  20/11/2020 17:52:28.000 Nov 20 09:52:28 172.16.0.1    network_manager tty388  172.16.68.8  stop  task_id=182   timezone=SGT  service=shell  start_time=1605865949  priv-lvl=15  cmd=show running-config <cr>
                           host = kali   source = /var/log/tac_plus.acct   sourcetype = tacacs_acct
>  20/11/2020 17:52:21.000 Nov 20 09:52:21 172.16.0.1    network_manager tty388  172.16.68.8  stop  task_id=181   timezone=SGT  service=shell  start_time=1605865941  priv-lvl=0   cmd=enable <cr>
                           host = kali   source = /var/log/tac_plus.acct   sourcetype = tacacs_acct
>  20/11/2020 17:51:17.000 Nov 20 09:51:17 172.16.68.3   network_manager tty1    172.16.68.8  stop  task_id=442   timezone=SGT  service=shell  start_time=1605865877  priv-lvl=0   cmd=exit <cr>
                           host = kali   source = /var/log/tac_plus.acct   sourcetype = tacacs_acct
>  20/11/2020 17:51:14.000 Nov 20 09:51:14 172.16.68.3   network_manager tty1    172.16.68.8  stop  task_id=441   timezone=SGT  service=shell  start_time=1605865875  priv-lvl=15  cmd=write <cr>
                           host = kali   source = /var/log/tac_plus.acct   sourcetype = tacacs_acct
>  20/11/2020 17:50:45.000 Nov 20 09:50:45 172.16.68.3   network_manager tty1    172.16.68.8  stop  task_id=440   timezone=SGT  service=shell  start_time=1605865846  priv-lvl=0   cmd=enable <cr>
                           host = kali   source = /var/log/tac_plus.acct   sourcetype = tacacs_acct
>  20/11/2020 17:50:03.000 Nov 20 09:50:03 172.16.68.5   network_manager tty2    172.16.68.8  stop  task_id=4116  timezone=UTC  service=shell  start_time=1605865803  priv-lvl=0   cmd=exit <cr>
                           host = kali   source = /var/log/tac_plus.acct   sourcetype = tacacs_acct
```

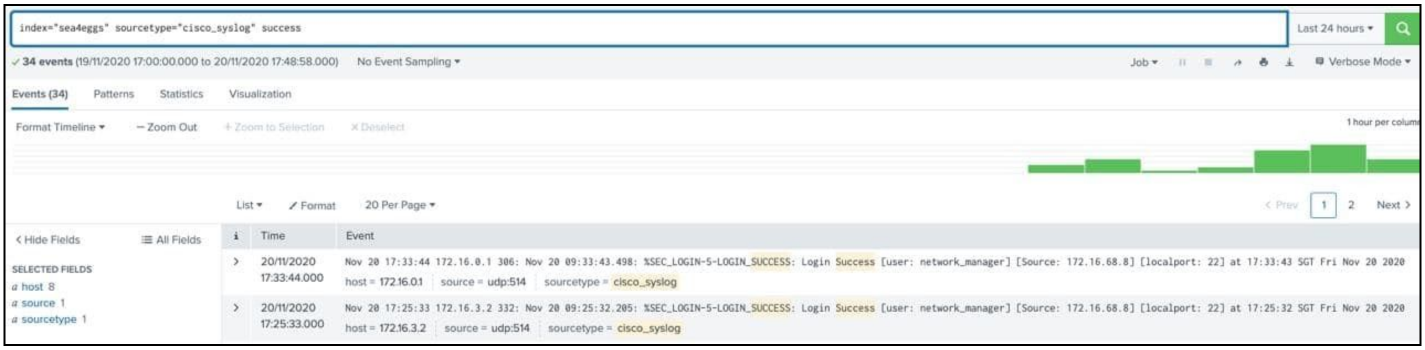
The filter <u>sourcetype="tacacs_acct"</u> will show details about commands executed by a user when they are connected to network devices configured to use AAA (All network devices use AAA in our setup).



```
index="sea4eggs" sourcetype="cisco_syslog" failed
```

✓ 16 events (19/11/2020 17:00:00.000 to 20/11/2020 17:49:12.000) No Event Sampling

Time	Event
20/11/2020 17:00:30.000	Nov 20 17:00:30 172.16.2.2 330: Nov 20 09:00:29.243: %SEC_LOGIN-4-LOGIN_FAILED: Login failed [user: network_manager] [Source: 172.16.68.12] [localport: 22] [Reason: Login Authentication Failed] at 09:00:29 UTC Fri Nov 20 2020 host = 172.16.2.2 source = udp:514 sourcetype = cisco_syslog
20/11/2020 17:00:27.000	Nov 20 17:00:27 172.16.2.2 329: Nov 20 09:00:26.405: %SEC_LOGIN-4-LOGIN_FAILED: Login failed [user: network_manager] [Source: 172.16.68.12] [localport: 22] [Reason: Login Authentication Failed] at 09:00:26 UTC Fri Nov 20 2020 host = 172.16.2.2 source = udp:514 sourcetype = cisco_syslog



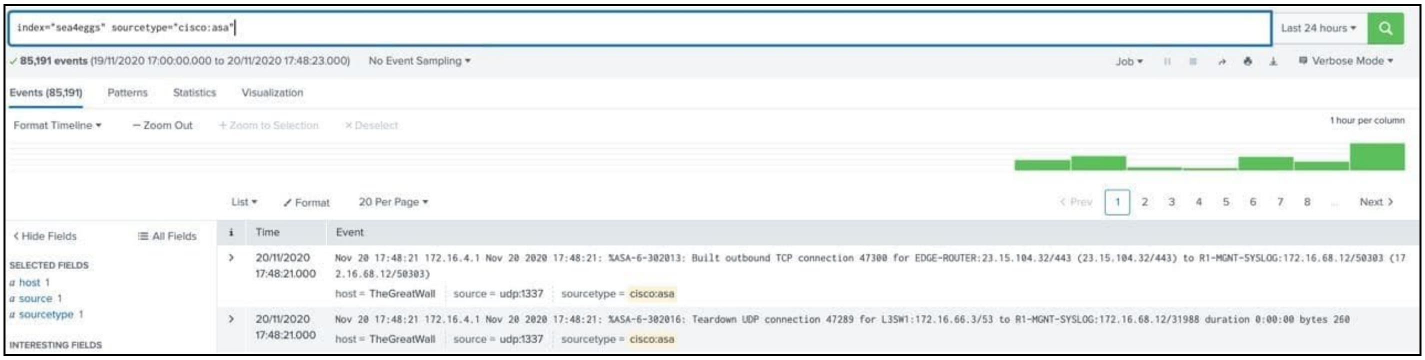
```
index="sea4eggs" sourcetype="cisco_syslog" success
```

✓ 34 events (19/11/2020 17:00:00.000 to 20/11/2020 17:48:58.000) No Event Sampling

Time	Event
20/11/2020 17:33:44.000	Nov 20 17:33:44 172.16.0.1 306: Nov 20 09:33:43.498: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: network_manager] [Source: 172.16.68.8] [localport: 22] at 17:33:43 SGT Fri Nov 20 2020 host = 172.16.0.1 source = udp:514 sourcetype = cisco_syslog
20/11/2020 17:25:33.000	Nov 20 17:25:33 172.16.3.2 332: Nov 20 09:25:32.205: %SEC_LOGIN-5-LOGIN_SUCCESS: Login Success [user: network_manager] [Source: 172.16.68.8] [localport: 22] at 17:25:32 SGT Fri Nov 20 2020 host = 172.16.3.2 source = udp:514 sourcetype = cisco_syslog

The filter sourcetype="cisco_syslog" will display all syslog messages from all Cisco network devices, which includes all successful and failed login attempts made on the different network devices. This can be used to keep track of all login attempts, even if the AAA server is unavailable, as well as any system messages generated by the network device (E.g. When a switch port or uplink is down on a network device).

In addition to security monitoring, this can also be used to enable the team to respond to and troubleshoot potential network issues by analysing the syslog messages.

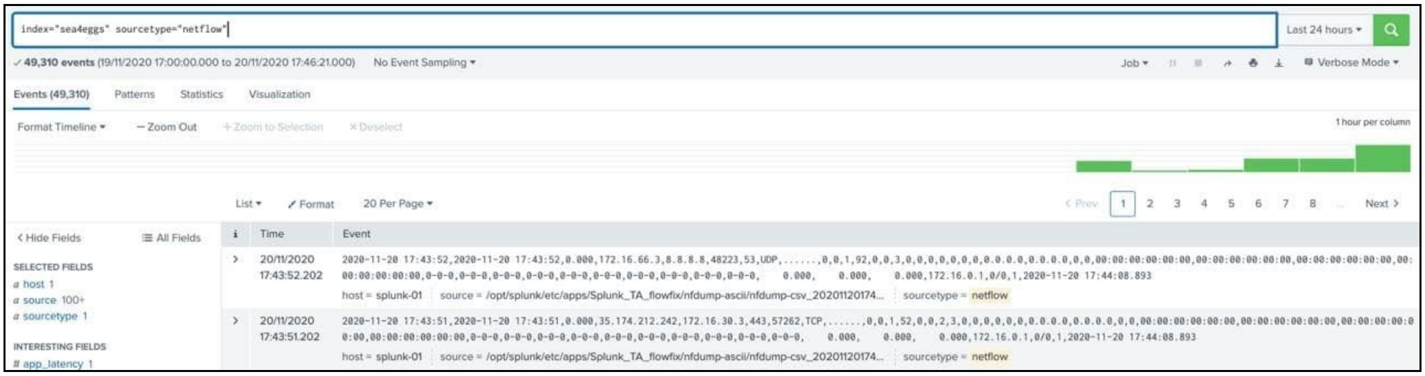


```
index="sea4eggs" sourcetype="cisco:asa"
```

✓ 85,191 events (19/11/2020 17:00:00.000 to 20/11/2020 17:48:23.000) No Event Sampling

Time	Event
20/11/2020 17:48:21.000	Nov 20 17:48:21 172.16.4.1 Nov 20 2020 17:48:21: %ASA-6-302013: Built outbound TCP connection 47300 for EDGE-ROUTER:23.15.104.32/443 (23.15.104.32/443) to R1-MGNT-SYSLOG:172.16.68.12/50303 (172.16.68.12/50303) host = TheGreatWall source = udp:1337 sourcetype = cisco:asa
20/11/2020 17:48:21.000	Nov 20 17:48:21 172.16.4.1 Nov 20 2020 17:48:21: %ASA-6-302016: Teardown UDP connection 47289 for L3SW1:172.16.66.3/53 to R1-MGNT-SYSLOG:172.16.68.12/31988 duration 0:00:00 bytes 260 host = TheGreatWall source = udp:1337 sourcetype = cisco:asa

The filter sourcetype="cisco:asa" will show syslog messages from the Cisco ASA firewall, including any established connections as well as denied traffic based on configured Access Control Lists (ACLs). This information can be used to alert us to any abnormal traffic that is denied by the firewall.



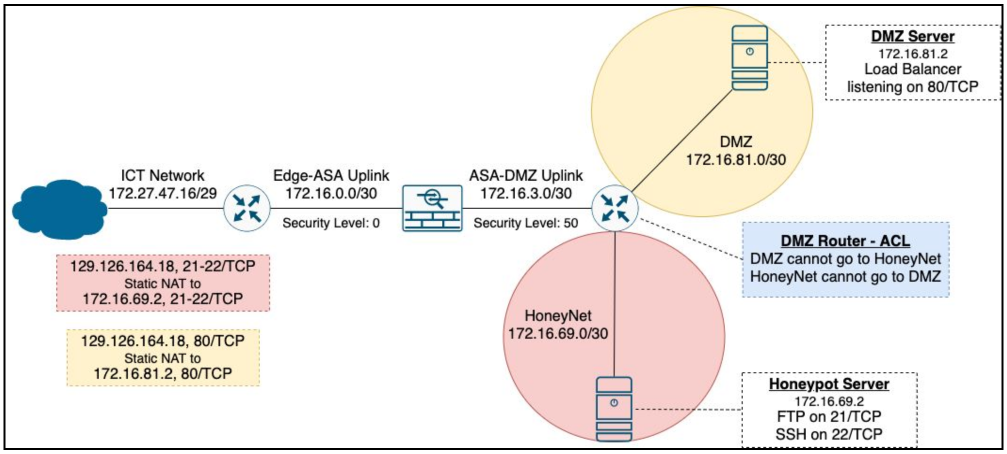
```
index="sea4eggs" sourcetype="netflow"
```

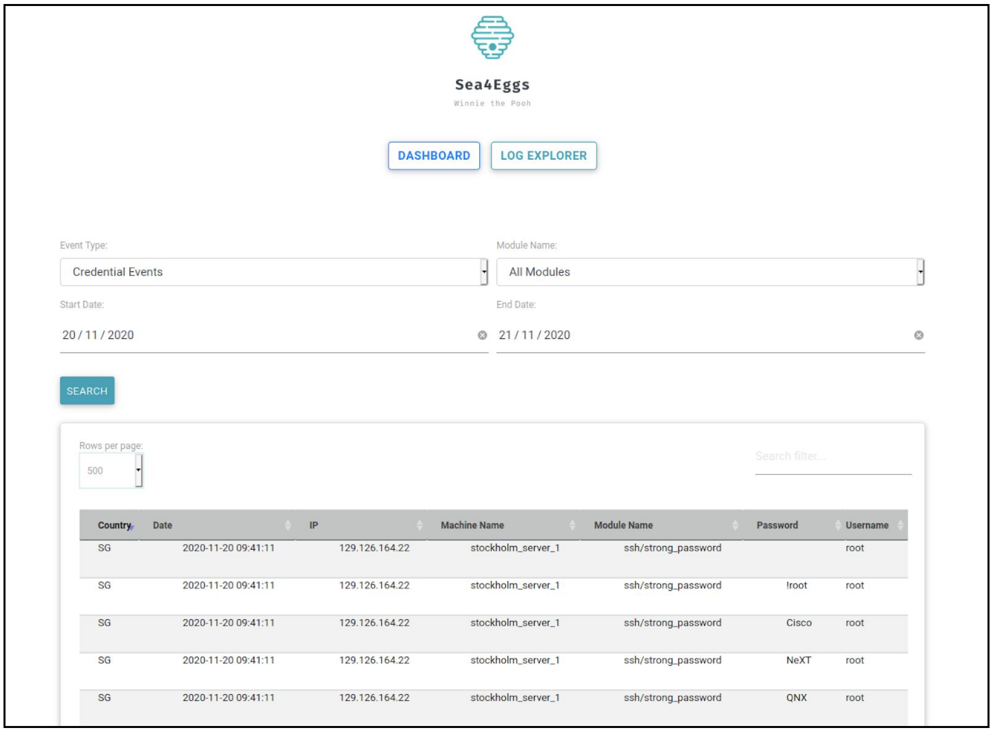
```
>  20/11/2020     2020-11-20 17:43:52,2020-11-20 17:43:52,0.000,172.16.66.3,8.8.8.8,48223,53,UDP,......,0,0,1,92,0,0,3,0,0,0,0,0,0,0,0.0.0.0,0.0.0.0,0,0,00:00:00:00:00:00,00:00:00:00:00:00,00:00:00:00:00:00,00:
   17:43:52.202   00:00:00:00:00,0-0-0,0-0-0,0-0-0,0-0-0,0-0-0,0-0-0,0-0-0,0-0-0,0-0-0,0-0-0,    0.000,    0.000,    0.000,172.16.0.1,0/0,1,2020-11-20 17:44:08.893
                  host = splunk-01   source = /opt/splunk/etc/apps/Splunk_TA_flowfix/nfdump-ascii/nfdump-csv_20201120174...   sourcetype = netflow

>  20/11/2020     2020-11-20 17:43:51,2020-11-20 17:43:51,0.000,35.174.212.242,172.16.30.3,443,57262,TCP,......,0,0,1,52,0,0,2,3,0,0,0,0,0,0,0.0.0.0,0.0.0.0,0,0,00:00:00:00:00:00,00:00:00:00:00:00,00:00:00:00:00:0
   17:43:51.202   0:00,00:00:00:00:00:00,0-0-0,0-0-0,0-0-0,0-0-0,0-0-0,0-0-0,0-0-0,0-0-0,0-0-0,0-0-0,    0.000,    0.000,    0.000,172.16.0.1,0/0,1,2020-11-20 17:44:08.893
                  host = splunk-01   source = /opt/splunk/etc/apps/Splunk_TA_flowfix/nfdump-ascii/nfdump-csv_20201120174...   sourcetype = netflow
```

The filter sourcetype="netflow" shows the NetFlow information from our Edge Router, which can be analysed to provide useful insight into the traffic between the WAN and our network.

Honeypot

The team has implemented a honeypot as a way to distract and monitor potential adversaries. This will allow the team to gain more information on them which will aid us in documenting and tracking any adversary's Tactics, Techniques and Procedures (TTP). The team has implemented **OWASP Honeypot** running FTP (21/TCP) and SSH (22/TCP). The honeypot services are configured to be exposed on the same domain and public IPv4 address of our web service externally.





Sea4Eggs
Winnie the Pooh

Event Type: Credential Events

Module Name: All Modules

Start Date: 20 / 11 / 2020

End Date: 21 / 11 / 2020

Rows per page: 500

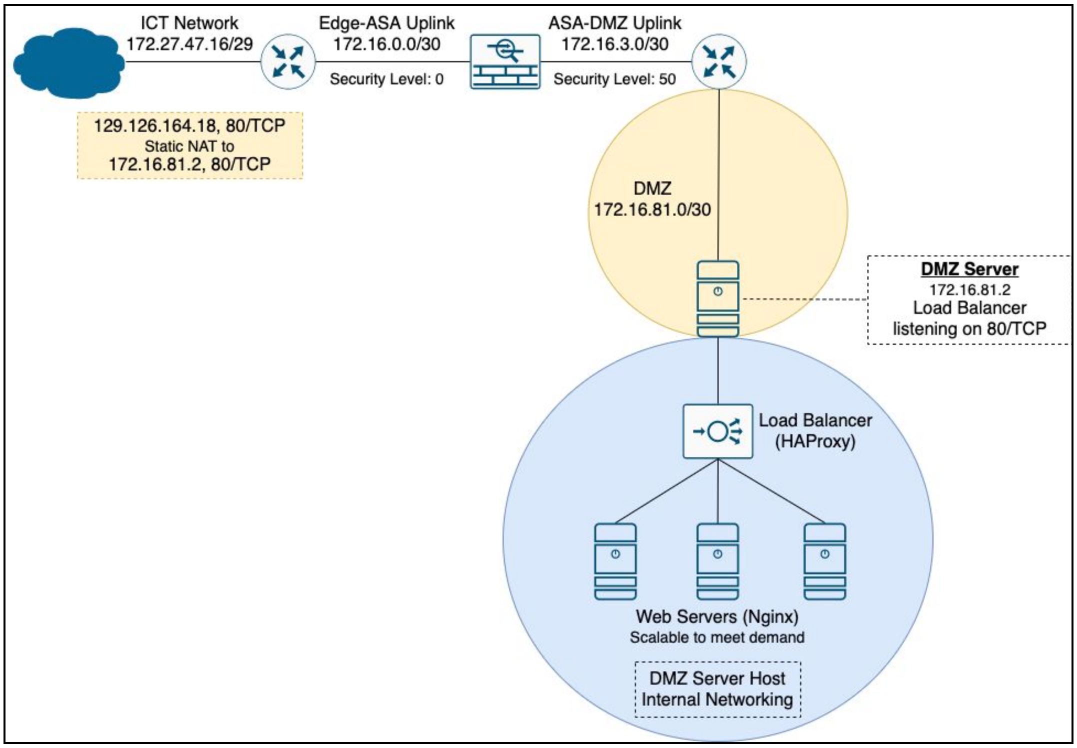
Search filter...

Country	Date	IP	Machine Name	Module Name	Password	Username
SG	2020-11-20 09:41:11	129.126.164.22	stockholm_server_1	ssh/strong_password		root
SG	2020-11-20 09:41:11	129.126.164.22	stockholm_server_1	ssh/strong_password	!root	root
SG	2020-11-20 09:41:11	129.126.164.22	stockholm_server_1	ssh/strong_password	Cisco	root
SG	2020-11-20 09:41:11	129.126.164.22	stockholm_server_1	ssh/strong_password	NeXT	root
SG	2020-11-20 09:41:11	129.126.164.22	stockholm_server_1	ssh/strong_password	QNX	root

The above screen capture shows the Log Explorer of OWASP Honeypot with the 'ssh/strong_password' module log information. This module will log all SSH attempts by an adversary and allows us to see the IP address of the adversary. This information can then be used to generate an IP blocklist on the edge router or firewall to deny further attacks from that particular adversary.

Load-Balanced Web Service Stack with Application-Layer Defences

The team has implemented load balancing to efficiently distribute HTTP traffic across multiple web servers. As a critical service that is exposed to the WAN, it is important that high availability is ensured so that legitimate users can access it, while malicious users are denied from causing havoc.



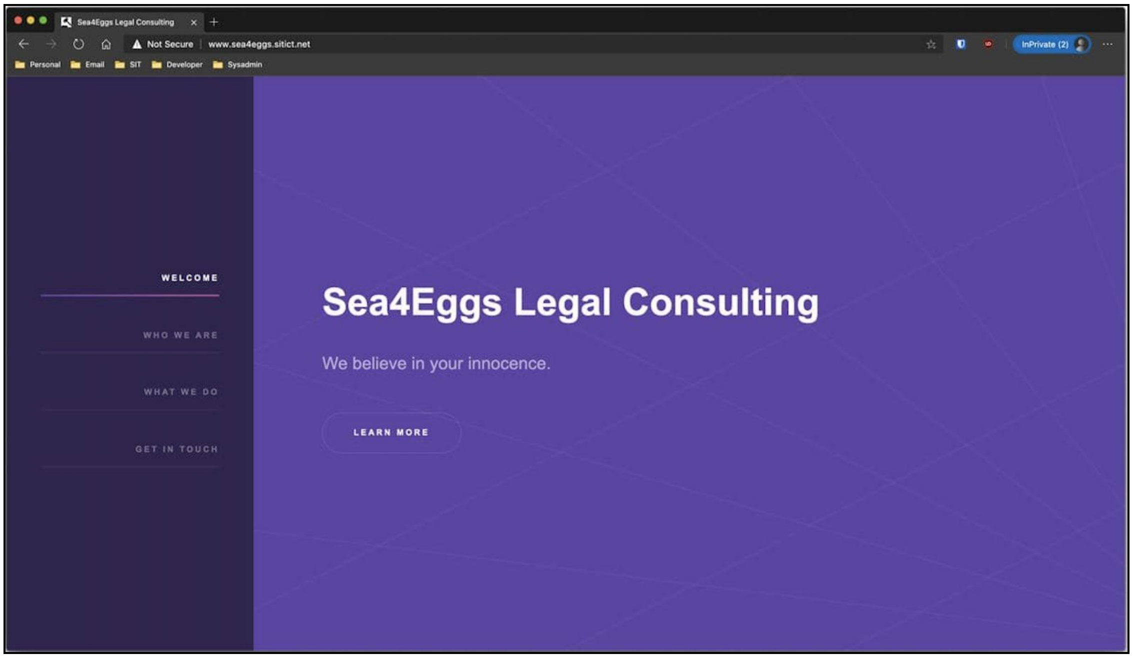
The software used for the Load Balancer in our setup, **HAProxy**, is built to handle high volumes of concurrent connections without using much system resources. Behind the Load Balancer, a few Web Servers running **Nginx** will be running to serve web content to the end-users, each capable of handling thousands of HTTP requests due to its asynchronous event-driven architecture. The number of backend Web Servers can be scaled up or down depending on the traffic volume, and with Load Balancer configured to distribute the incoming traffic to these multiple web servers, efficiency can be maximised while mitigating the impacts of fluctuations in HTTP traffic. The team has configured 5 web servers running backend for the project setup.

HTTP error rates per client is monitored since a high volume of HTTP errors within a few seconds may signify a HTTP scan or attack. Connections per client within a time frame (Measured in HTTP requests per second) is also tracked on the Load Balancer and backend Web Servers, dropping requests if a client exceeds the rate limit, mitigating Denial-of-Service (DoS) attacks. A HTTP client timeout is also set on both the Load Balancer and backend Web Servers to protect against Slowloris-type of attacks.

Such incidents will be logged and further HTTP requests from the offending client is blocked by the Load Balancer for a few minutes. HTTP traffic logs are also configured and sent to Splunk via Syslog for centralised monitoring.

Public-Facing Web Service with Security Monitoring

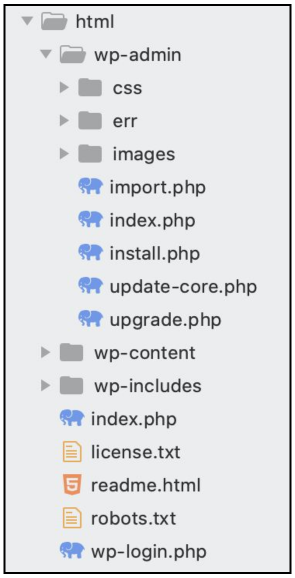
As part of the project requirements, a public corporate website has been set up and served by the load-balanced Web Service stack in the network to the WAN. This website can be accessed via the domain "www.sea4eggs.sitict.net" or via the public IP address of the web service. The screen capture below shows the landing page of the website.



Sea4Eggs Legal Consulting

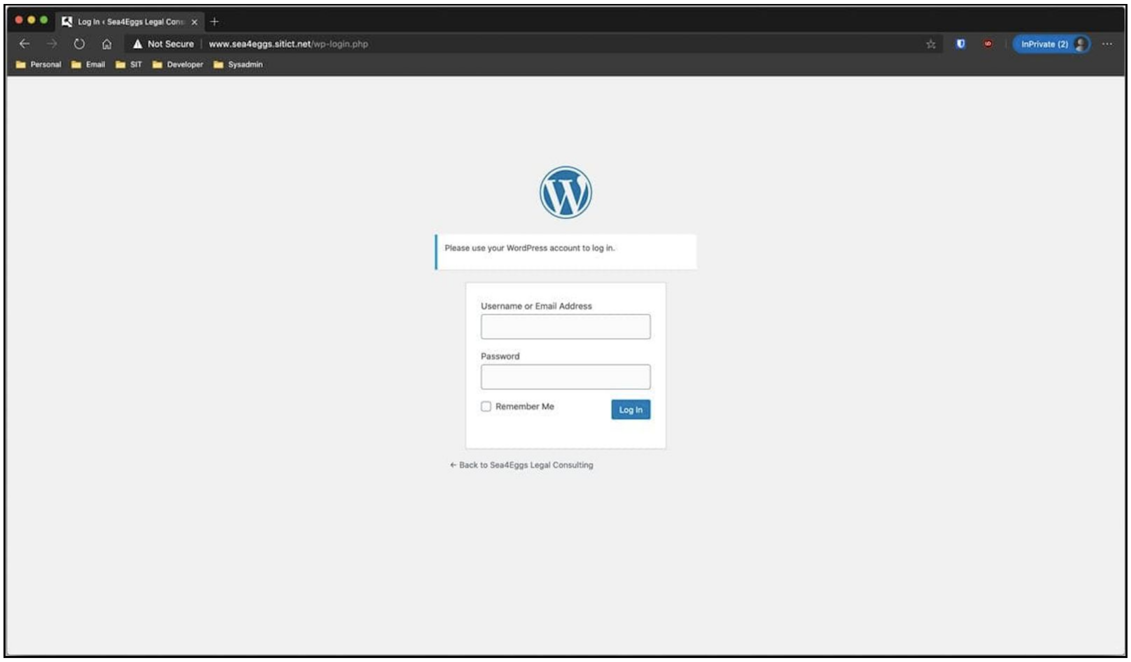
We believe in your innocence.

As part of the team's efforts to monitor and to distract potential adversaries, the structure of the website has been set up to mimic the WordPress content management system, to confuse adversaries performing reconnaissance on the Web Service, as seen in the below screen capture.



```
▼ html
    ▼ wp-admin
        ▶ css
        ▶ err
        ▶ images
          import.php
          index.php
          install.php
          update-core.php
          upgrade.php
    ▶ wp-content
    ▶ wp-includes
      index.php
      license.txt
      readme.html
      robots.txt
      wp-login.php
```

The web pages are static and contain no server-side code, even though the file extensions end with ".php".


Log In ‹ Sea4Eggs Legal Cons
Not Secure
www.sea4eggs.sitict.net/wp-login.php
Personal
Email
SIT
Developer
Sysadmin
InPrivate (2)

Please use your WordPress account to log in.

Username or Email Address

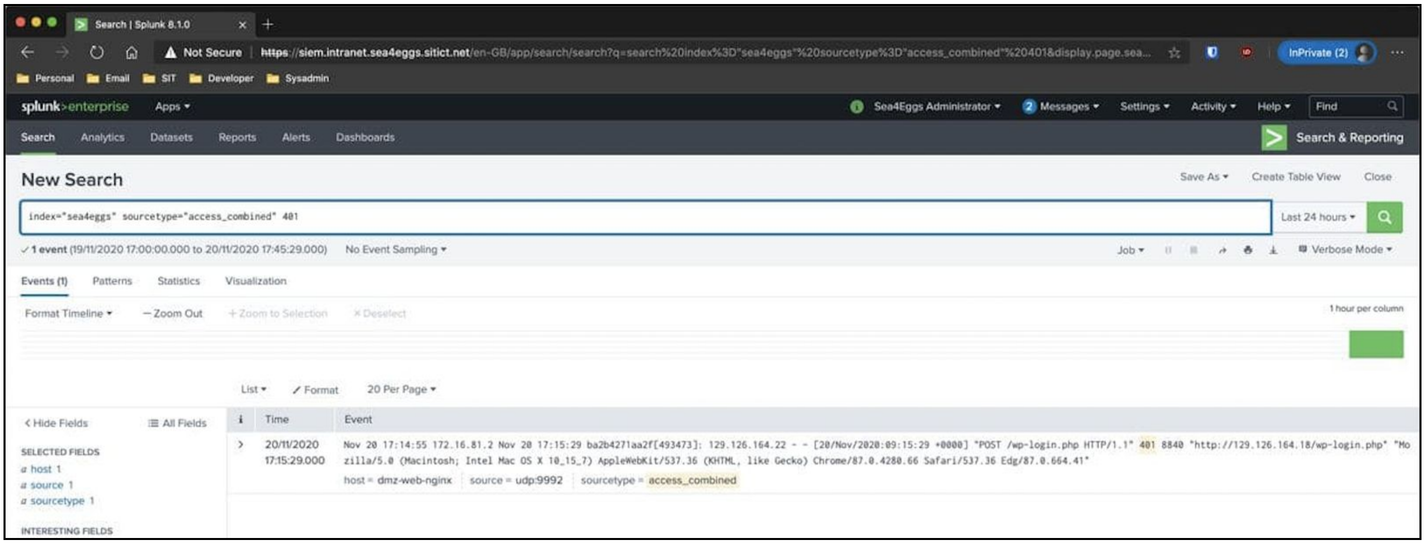
Password

□ Remember Me

Log In
← Back to Sea4Eggs Legal Consulting

A fake "WordPress Admin" login page is also set up to lure potential adversaries and detect their presence. As it is a fake static login page with no server-side code or processing logic, it has minimal to no impact on the web server. A "HTTP 401 Unauthorized" error is always returned by the Web Service if a user tries to login.

This error is logged and sent to Splunk via Syslog, allowing the team to monitor for potential adversary activity and also capture their IP address, as seen in the below screen capture.



New Search

```
index="sea4eggs" sourcetype="access_combined" 401
```

✓ 1 event (19/11/2020 17:00:00.000 to 20/11/2020 17:45:29.000) No Event Sampling

i	Time	Event
>	20/11/2020 17:15:29.000	Nov 20 17:14:55 172.16.81.2 Nov 20 17:15:29 ba2b4271aa2f[493473]: 129.126.164.22 - - [20/Nov/2020:09:15:29 +0000] "POST /wp-login.php HTTP/1.1" 401 8840 "http://129.126.164.18/wp-login.php" "Mozilla/5.0 (Macintosh; Intel Mac OS X 10_15_7) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/87.0.4280.66 Safari/537.36 Edg/87.0.664.41" host = dmz-web-nginx source = udp:9992 sourcetype = access_combined