

# SSH LOG FILE ANALYSIS (NETWORK BASED ATTACK SITUATION)

By Mohammed Choglay



## Contents

<b>Section 1: outlining information .....</b>	<b>2</b>
<b>Section 2: Analysis of SSH log file .....</b>	<b>2</b>
<b>Section 2.1: Hash verification (Integrity Check) .....</b>	<b>6</b>
<b>Section 3: Two Remediations to network based password attacks.....</b>	<b>7</b>
<b>Section 3.1: Password Policies .....</b>	<b>7</b>
<b>Section 3.2: Account Monitoring and Anomaly Detection .....</b>	<b>7</b>

## Section 1: Outlining information

The following analysis of the SSH log file will allow for the determination of possible malicious behaviour against that particular service. Before attempting any analysis on the log file, a hash must be taken of the log file to ensure no modifications have been made throughout the examination. The integrity of the log file is paramount.



```
[mchoglay@mohammed-vmwarevirtualplatform]~  
$ sha256sum sshlog.log  
906c0da10bffa1c46836f23444ab98ac61358e2e4e9c1941130d32366ebbf  sshlog.log  
[mchoglay@mohammed-vmwarevirtualplatform]~  
$
```

Figure 1 - hash of the log file

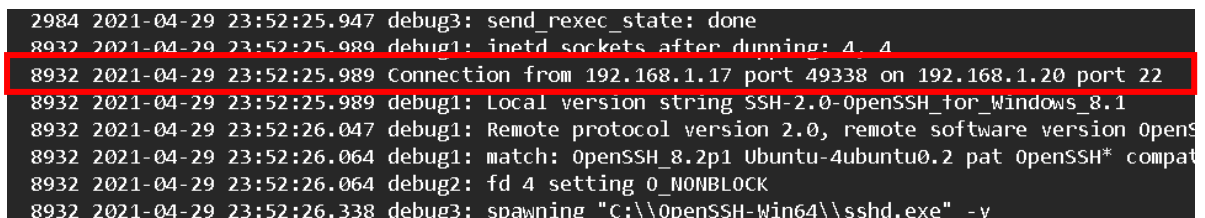
### Section 1.1: Prerequisites for analysis

In order to conduct this investigation, the following will be required:

1. Windows or Linux system
2. SSH log file that need examining
3. Log file reader with read only protections (For manual analysis)
4. Regex tool which does not modify the file

## Section 2: Analysis of SSH log file

Upon first analysis, a connection was made from 192.168.17 to 192.168.1.20. What can be determined from looking at this is that the connection is within an internal network.

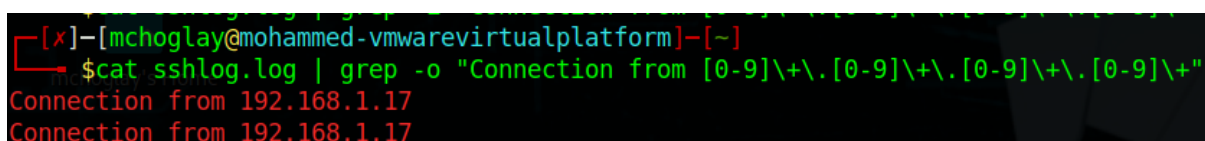


```
2984 2021-04-29 23:52:25.947 debug3: send_rexec_state: done  
8932 2021-04-29 23:52:25.989 debug1: inetd sockets after dupping: 4, 4  
8932 2021-04-29 23:52:25.989 Connection from 192.168.1.17 port 49338 on 192.168.1.20 port 22  
8932 2021-04-29 23:52:25.989 debug1: Local version string SSH-2.0-OpenSSH_for_Windows_8.1  
8932 2021-04-29 23:52:26.047 debug1: Remote protocol version 2.0, remote software version OpenSSH_8.2p1 Ubuntu-4ubuntu0.2  
8932 2021-04-29 23:52:26.064 debug1: match: OpenSSH_8.2p1 Ubuntu-4ubuntu0.2 pat OpenSSH* compat  
8932 2021-04-29 23:52:26.064 debug2: fd 4 setting O_NONBLOCK  
8932 2021-04-29 23:52:26.338 debug3: spawning "C:\\OpenSSH-Win64\\sshd.exe" -y
```

Figure 2 - 1st connection address

Using keywords, it can be determined if any other addresses were attempting to connect to the SSH services utilising regular expression (bash regex).

In Figure 3 below, the words "Connected from" and a regex expression for every possible IP address were used. A recurring result was displayed, which was "Connected from 192.168.1.17". The number of lines was checked using the "wc -l" command to prove this. This was also cross-checked against the exact IP address to ensure nothing was missed. This is shown in Figure 4 below, where 1844 results were outputted on both occasions.



```
[x]-[mchoglay@mohammed-vmwarevirtualplatform]~  
$ cat sshlog.log | grep -o "Connection from [0-9]\+\.[0-9]\+\.[0-9]\+\.[0-9]\+"  
Connection from 192.168.1.17  
Connection from 192.168.1.17
```

Figure 3 - grepping every possible address

```

[mchoglay@mohammed-vmwarevirtualplatform]~[~]
$cat sshlog.log | grep -o "Connection from [0-9]\+\.[0-9]\+\.[0-9]\+\.[0-9]\+" | wc -l
1844
[mchoglay@mohammed-vmwarevirtualplatform]~[~]
$cat sshlog.log | grep -o "Connection from 192.168.1.17" | wc -l
1844

```

Figure 4 - grep cross-reference

What was discovered close to the first initial connection was an Invalid user attempt on an admin account, which occurred on 29/04/2021 at 23:53:07.071. This invalid attempt originated from an internal IP address of 192.168.1.17. This is shown below in Figure 5.

```

4544 2021-04-29 23:53:07.071 debug3: match not found
4544 2021-04-29 23:53:07.071 Invalid user admin from 192.168.1.17 port 49342
4544 2021-04-29 23:53:07.071 debug3: mm_answer_pwnamallow: sending MONITOR_ANS_PWNAM: 0
4544 2021-04-29 23:53:07.071 debug3: mm_request_send entering: type 9
4544 2021-04-29 23:53:07.071 debug2: monitor_read: 8 used once, disabling now
4544 2021-04-29 23:53:07.072 debug3: mm_inform_authserv entering [preauth]

```

Figure 5 - 1st invalid user logged

Examining the invalid attempts, a further 1467 results were found, all within proximity and attempted at unusual times. These results can be verified in Figure 6 and Figure 7.

```

[mchoglay@mohammed-vmwarevirtualplatform]~[~]
$cat sshlog.log | grep -E "Invalid user"
4544 2021-04-29 23:53:07.071 Invalid user admin from 192.168.1.17 port 49342
8184 2021-04-30 00:00:55.187 Invalid user <username> from 192.168.1.17 port 53898
4364 2021-04-30 00:04:04.387 Invalid user jake from 192.168.1.17 port 33760
1248 2021-04-30 00:20:48.347 Invalid user root from 192.168.1.17 port 37990
2068 2021-04-30 00:20:49.093 Invalid user root from 192.168.1.17 port 37994
4168 2021-04-30 00:20:49.141 Invalid user admin from 192.168.1.17 port 37992
8340 2021-04-30 00:20:49.234 Invalid user webadmin from 192.168.1.17 port 37996
6992 2021-04-30 00:20:49.234 Invalid user sysadmin from 192.168.1.17 port 38000
7720 2021-04-30 00:20:50.577 Invalid user netadmin from 192.168.1.17 port 38002

```

Figure 6 - Invalid user attempts

```

[mchoglay@mohammed-vmwarevirtualplatform]~[~]
$cat sshlog.log | grep -w "Invalid user" | wc -l
1467

```

Figure 7 - amount of failed invalid attempts

Looking at Figure 8 closely, you can determine that a password attack is highly likely as the internal IP address is always the same, and the username constantly changes, indicating this is a password-spraying attack. The purpose behind password spraying is a type of brute force attack where the malicious actor utilises the same password on many different accounts. For example, you have a password in the password list: Choglay's-Dream321. That password in the list will be used against all the accounts like Jake, root, admin and more.

However, deeper into the log file, you can see that the malicious actor changes from a password-spraying attack to a brute force attack or dictionary attack. An example of this is shown below in Figure 9. The password attack type can't be verified as passwords are not stored in plain text or hashed within the log file.

Data has been provided in Table 1 of all the failed attempts against accounts. The total is correct as it added up to 1467, verified against Figure 7.

```

6364 2021-04-30 00:21:04.034 Invalid user user from 192.168.1.17 port 38174
1272 2021-04-30 00:21:04.034 Invalid user web from 192.168.1.17 port 38180
8716 2021-04-30 00:21:04.563 Invalid user test from 192.168.1.17 port 38182
3012 2021-04-30 00:21:04.785 Invalid user root from 192.168.1.17 port 38184
1952 2021-04-30 00:21:04.887 Invalid user admin from 192.168.1.17 port 38188
8688 2021-04-30 00:21:05.521 Invalid user webadmin from 192.168.1.17 port 38194
4140 2021-04-30 00:21:05.536 Invalid user sysadmin from 192.168.1.17 port 38192
3056 2021-04-30 00:21:05.608 Invalid user netadmin from 192.168.1.17 port 38200
5808 2021-04-30 00:21:06.617 Invalid user user from 192.168.1.17 port 38204
8828 2021-04-30 00:21:06.825 Invalid user web from 192.168.1.17 port 38208
6628 2021-04-30 00:21:07.249 Invalid user root from 192.168.1.17 port 38206
9108 2021-04-30 00:21:07.250 Invalid user test from 192.168.1.17 port 38212
5512 2021-04-30 00:21:07.304 Invalid user admin from 192.168.1.17 port 38210
8276 2021-04-30 00:21:07.656 Invalid user webadmin from 192.168.1.17 port 38218
8928 2021-04-30 00:21:07.706 Invalid user sysadmin from 192.168.1.17 port 38216
7412 2021-04-30 00:21:08.169 Invalid user netadmin from 192.168.1.17 port 38222
6880 2021-04-30 00:21:08.544 Invalid user user from 192.168.1.17 port 38226
7012 2021-04-30 00:21:08.707 Invalid user web from 192.168.1.17 port 38224
3924 2021-04-30 00:21:08.814 Invalid user test from 192.168.1.17 port 38228
1440 2021-04-30 00:21:09.033 Invalid user root from 192.168.1.17 port 38232
8892 2021-04-30 00:21:09.093 Invalid user admin from 192.168.1.17 port 38230
8092 2021-04-30 00:21:09.922 Invalid user webadmin from 192.168.1.17 port 38236
3488 2021-04-30 00:21:10.126 Invalid user sysadmin from 192.168.1.17 port 38238
8596 2021-04-30 00:21:11.870 Invalid user netadmin from 192.168.1.17 port 38242
7092 2021-04-30 00:21:11.870 Invalid user user from 192.168.1.17 port 38244
2032 2021-04-30 00:21:11.976 Invalid user web from 192.168.1.17 port 38254

```

Figure 8 - password spraying

```

764 2021-04-30 00:27:28.385 Invalid user jake from 192.168.1.17 port 40448
488 2021-04-30 00:27:28.615 Invalid user jake from 192.168.1.17 port 40450
7464 2021-04-30 00:27:28.715 Invalid user jake from 192.168.1.17 port 40452
8396 2021-04-30 00:27:28.878 Invalid user jake from 192.168.1.17 port 40454
8496 2021-04-30 00:27:28.931 Invalid user jake from 192.168.1.17 port 40456
1088 2021-04-30 00:27:29.747 Invalid user jake from 192.168.1.17 port 40458
7120 2021-04-30 00:27:29.882 Invalid user jake from 192.168.1.17 port 40464
7720 2021-04-30 00:27:29.911 Invalid user jake from 192.168.1.17 port 40460
6936 2021-04-30 00:27:30.008 Invalid user jake from 192.168.1.17 port 40462
7340 2021-04-30 00:27:30.081 Invalid user jake from 192.168.1.17 port 40466
3652 2021-04-30 00:27:30.173 Invalid user jake from 192.168.1.17 port 40468

```

Figure 9 - brute force attack or dictionary attack

Usernames	Password Attempts
admin	83
<username>	1
user	79
Jake	524
Root	84
Web	78

Test	78
webadmin	82
netadmin	82
sysadmin	82
chris	74
janet	74
sammy	74
meghan	72
<b>Total:</b>	<b>1,467</b>

*Table 1 - data of all accounts attempted*

Now that there's been a clear picture that a password attack has occurred against SSH services. Further analysis must be conducted to determine if any accounts were authenticated instead of failed.

What was discovered was an account called "Sophia" in the debug information, which was authenticated on two occasions from the attacker's internal IP address. This indicates that Sophie's account was compromised. The data for this is shown in Figure 10 to Figure 12.

```

4848 2021-04-30 00:57:52.058 debug3: mm_auth_password: user not authenticated [preauth]
7560 2021-04-30 00:57:52.469 debug3: mm_auth_password: user not authenticated [preauth]
540 2021-04-30 00:57:52.514 debug3: mm_auth_password: user not authenticated [preauth]
8200 2021-04-30 00:57:52.660 debug3: mm_auth_password: user not authenticated [preauth]
8632 2021-04-30 00:57:52.694 debug3: mm_auth_password: user not authenticated [preauth]
7300 2021-04-30 01:01:11.699 debug1: monitor_child_preauth: sophia has been authenticated by privileged process
7300 2021-04-30 01:01:11.702 debug3: mm_auth_password: user authenticated [preauth]
428 2021-04-30 01:15:58.094 debug3: mm_auth_password: user not authenticated [preauth]
7036 2021-04-30 01:15:58.148 debug3: mm_auth_password: user not authenticated [preauth]
764 2021-04-30 01:15:58.306 debug3: mm_auth_password: user not authenticated [preauth]
6568 2021-04-30 01:15:58.317 debug3: mm_auth_password: user not authenticated [preauth]
6632 2021-04-30 01:16:00.235 debug3: mm_auth_password: user not authenticated [preauth]

```

*Figure 10 - Account authenticated*

```

debug3: receive packet: type 50 [preauth]
debug1: userauth-request for user sophia service ssh-connection method password [preauth]
debug1: attempt 2 failures 1 [preauth]
debug2: input_userauth_request: try method password [preauth]
debug3: mm_auth_password entering [preauth]
debug3: mm_request_send entering: type 12 [preauth]
debug3: mm_auth_password: waiting for MONITOR_ANS_AUTHPASSWORD [preauth]
debug3: mm_request_receive_expect entering: type 13 [preauth]
debug3: mm_request_receive entering [preauth]
debug3: mm_request_receive entering
debug3: monitor_read: checking request 12
debug3: mm_answer_authpassword: sending result 1
debug3: mm_request_send entering: type 13
Accepted password for sophia from 192.168.1.17 port 42364 ssh2
debug1: monitor_child_preauth: sophia has been authenticated by privileged process
debug3: mm_get_keystate: Waiting for new keys
debug3: mm_request_receive_expect entering: type 26
debug3: mm_request_receive entering
debug3: mm_get_keystate: GOT new keys
debug3: mm_auth_password: user authenticated [preauth]
debug3: user_specific_delay: user specific delay 0.000ms [preauth]
debug3: ensure_minimum_time_since: elapsed 2.001ms, delaying 3.048ms (requested 5.049ms) [preauth]
debug3: send packet: type 52 [preauth]
debug3: mm_request_send entering: type 26 [preauth]
debug3: mm_send_keystate: Finished sending state [preauth]
debug1: monitor_read_log: child log fd closed
debug3: spawning "C:\\OpenSSH-Win64\\sshd.exe" -z
User child is on pid 9136
debug3: send_rexec_state: entering fd = 5 config len 291
debug3: ssh_msg_send: type 0
debug3: send_rexec_state: done
debug3: ssh_msg_send: type 0
debug3: ssh_msg_send: type 0
debug3: ssh_msg_send: type 0
debug3: ssh_msg_send: type 0

```

Figure 11 - Account authenticated

```

[mchoglay@mohammed-vmwarevirtualplatform]~$ cat sshlog.log | grep -wE "Accepted password"
7176 2021-04-30 00:53:25.023 Accepted password for sophia from 192.168.1.17 port 41990 ssh2
7300 2021-04-30 01:01:11.699 Accepted password for sophia from 192.168.1.17 port 42364 ssh2
[mchoglay@mohammed-vmwarevirtualplatform]~$

```

Figure 12 - amount of times account authenticated

## Section 2.1: Hash verification (Integrity Check)

The hash of the log file was reverified to ensure nothing was modified throughout this investigation. This check upholds the integrity of the log file. Figure 13 and Figure 1 match, ensuring everything has stayed the same throughout this investigation.

```

[mchoglay@mohammed-vmwarevirtualplatform]~$ sha256sum sshlog.log
906c0da10bffacd1c46836f23444ab98ac61358e2e4e9c1941130d32366ebbf  sshlog.log
[mchoglay@mohammed-vmwarevirtualplatform]~$

```

Figure 13 - Hash verification

## Section 3: Two Remediations to network based password attacks

### Section 3.1: Password Policies

A strong password policy must be implemented within any network to ensure that your end users are more protected when implementing passwords. The principle behind this will significantly reduce the risk of unauthorised access, data breaches, and cyberattacks. A password policy does work on components. Some of these are as follows:

1. Password length (should be 12+ characters)
2. Complexity (Range of uppercase, lowercase, symbols, and numbers are within the password)
3. Don't use PPI or information that can be easily enumerated from open-source systems
4. No Password Reusage (You can't use old passwords)
5. Account lockout (After three wrong attempts, the account will be locked or IP banned.)
6. Password expiry (Must be forced to change password every three months)
7. Use MFA to access all platforms and services.

### Section 3.2: Account Monitoring and Anomaly Detection

In cybersecurity, a critical element is account monitoring of end users on the network. This ensures that these attacks are spotted and prevented in real time to mitigate and reduce damage. Tools and technologies have been developed to detect and prevent such attacks from occurring:

**Continuous Surveillance:** The purpose behind account surveillance is to watch all users move from accessing files, transfers in communication, login attempts and more. With privileged accounts, closer and finer surveillance should be carried out as they are more likely to have access to critical systems.

**Log Analysis:** The principle behind this, as seen through this report, is to examine log files collected by systems and from there, these logs can be analysed in manual and automated approaches to look out for suspicious activity.

#### **5 Account mentoring tools:**

1. Splunk
2. IBM QRadar
3. Microsoft Azure Sentinel
4. LogRhythm
5. Symantec Endpoint Detection and Response (EDR)

In terms of anomaly detection, this is implementing systems where AI and different analysis techniques are implemented to detect in real-time or even predict possible attacks that could occur from gathering a wide range of information. These can consist of the following but are not limited to:

1. Machine Learning and AI
2. Threshold-Based Detection
3. Real-Time Alerts
4. Historical Data Comparison
5. Response Automation