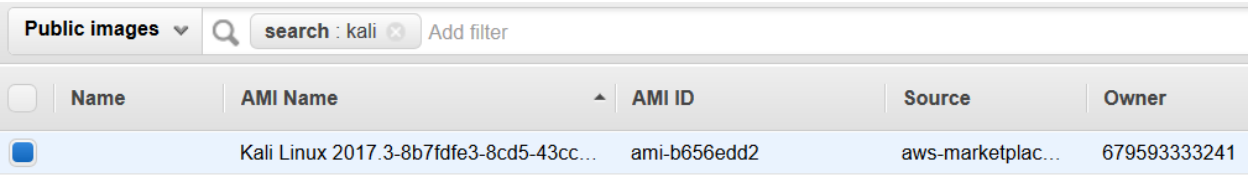
MIST

StarPhish Installation Guide

Initial Installation and setup guide for cloud deployment

Amazon AWS Kali-Linux Instance setup

You will have to setup a Kali-Linux Instance, you can use the free and public Kali AMI available in the Amazon Marketplace.



*Note: Kali is reccommended to run on at least 2GB of memory and 20GB of storage

Once Kali is running, login to the instance and update it.

sudo apt update && sudo apt full-upgrade

This will take some time. Once done, **reboot**.

Now, its time to initiate the postgresql database to enhance starphish features and to enable kingphisher for phishing your client.

sudo msfdb init

```
sudo msfdb init
Creating database user 'msf'
Enter password for new role:
Enter it again:
Creating databases 'msf' and 'msf_test'
Creating configuration file in /usr/share/metasploit-framework/config/database.y
ml
Creating initial database schema
```

Next, start the database

sudo systemctl start postgresql

Check the status to make sure it started ok

sudo systemctl status postgresql

```
● postgresql.service - PostgreSQL RDBMS
   Loaded: loaded (/lib/systemd/system/postgresql.service; disabled; vendor preset:
   Active: active (exited) since Sat 2018-01-20 22:49:18 UTC; 6min ago
  Process: 15180 ExecStart=/bin/true (code=exited, status=0/SUCCESS)
 Main PID: 15180 (code=exited, status=0/SUCCESS)
```

Then check to see what port postgresql is listening on

ss –atnp

```
State       Recv-Q Send-Q Local Address:Port               Peer Address:Port

LISTEN      0      128      0.0.0.0:22                    0.0.0.0:*
LISTEN      0      128    127.0.0.1:5432                  0.0.0.0:*
```

You'll see its listening on port **5432**

Next is a cleanup of the system.

Removing uneccessary packages from the system.

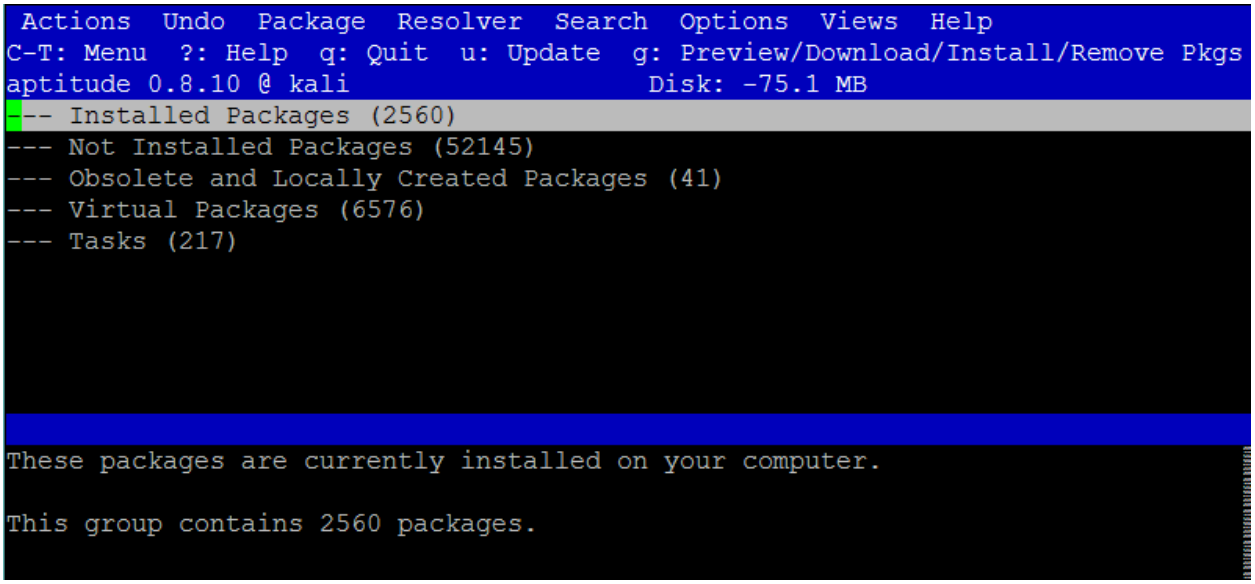
First we want to install the aptitude package manager

sudo apt install aptitude –y

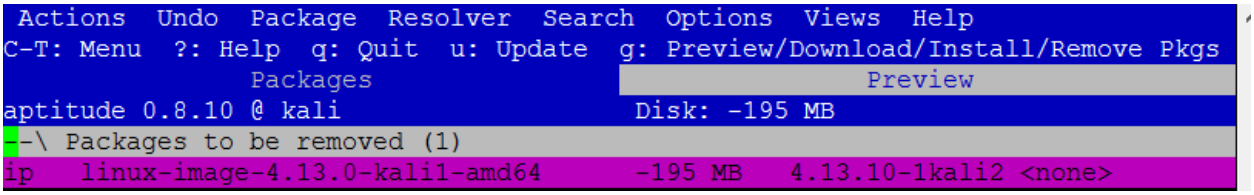
Then,

Start aptitiude

sudo aptitude



Scroll down to "Obsolete and Locally Created Packages (41)" and press **G** on your keyboard



*Note, some packages like old kernel files may be in use, reboot and relaunch aptitude to remove those packages after your initial run of aptitude.

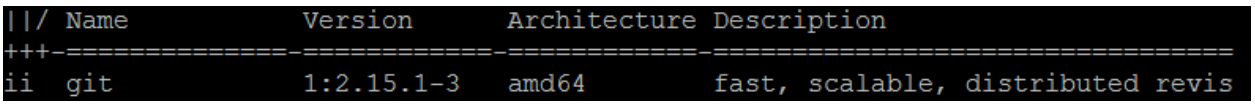
Press **G** again, to remove the purple highlighted packages.

Now that the system has been cleaned up its time to install Starphish.

Downloading and installing the Starphish Packages

First, make sure git is installed

dpkg –l git

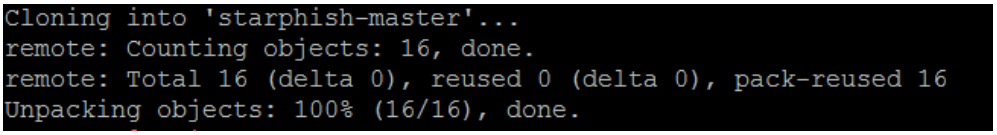


If git is not installed, install it using

apt install git –y

Next, cloning the starphish-master directory from github

git clone https://github.com/amodayus/starphish-master



Once the directory cloning has completed, change directories to ***/starphish-master/startup_scripts***

There you will find three scripts. The first script to run is dependencies.bash

sudo ./dependencies.bash

This will update or install all the required packages to setup Starphish.

The next script to run is the initial-setup.bash

sudo ./initial-setup.bash

This will download and install pymsf, a python package to interact with metasploit.

Now that the initial setup is underway, we are going to setup additional components for trojan creation.

Installing dependancies for creating Trojans

First install the latest Android NDK package for Linux 64-bit

wget https://dl.google.com/android/repository/android-ndk-####-linux-x86_64.zip

```
--2018-01-21 00:52:59--  https://dl.google.com/android/repository/android-ndk-r16b-linux-x86_6
4.zip
Resolving dl.google.com (dl.google.com)... 209.85.203.190, 209.85.203.91, 209.85.203.93, ...
Connecting to dl.google.com (dl.google.com)|209.85.203.190|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 852525873 (813M) [application/zip]
Saving to: 'android-ndk-r16b-linux-x86_64.zip'

android-ndk-r16b-linux- 100%[============================>] 813.03M  15.5MB/s    in 54s

2018-01-21 00:53:54 (15.0 MB/s) - 'android-ndk-r16b-linux-x86_64.zip' saved [852525873/8525258
73]
```

Once download is complete unzip it

unzip android-ndk-####-linux-x86_64.zip

And move the unzipped directory into /usr/share/android-ndk

sudo mv android-ndk-#### /usr/share/android-ndk

Now, we need to get the Android SDK package.

Within the starphish-master directory is a file, “android-sdk-download-link”, that unfortantly can’t be downloaded with curl or wget, instead download the file from your browser and scp it to your instance.

Once the sdk package file is transferred it needs to be installed

sudo apt install ./android-sdk_###kali0_all.deb

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'android-sdk' instead of './android-sdk_22.0.1-1kali0_all.deb'
The following NEW packages will be installed:
  android-sdk
0 upgraded, 1 newly installed, 0 to remove and 0 not upgraded.
Need to get 0 B/279 MB of archives.
After this operation, 131 MB of additional disk space will be used.
Get:1 /home/ec2-user/android-sdk_22.0.1-1kali0_all.deb android-sdk all 22.0.1-1kali0 [279 MB]
Selecting previously unselected package android-sdk.
(Reading database ... 292303 files and directories currently installed.)
Preparing to unpack .../android-sdk_22.0.1-1kali0_all.deb ...
Unpacking android-sdk (22.0.1-1kali0) ...
Setting up android-sdk (22.0.1-1kali0) ...
```

Now,

Back to the starphish-master directory, and change to the trojan_scripts directory. Now run “depedencies-builder.bash”

sudo ./dependencies-builder.bash

Click Y for all

So up to this point the dependencies have been installed and setup now its time to configure some aspects of Starphish.

Creating a keystore to sign your trojans

Create a keystore

keytool -genkey -v -keystore example.keystore -alias example-alias -keyalg RSA -keysize 2048 -validity 10000

```
Enter keystore password:
Re-enter new password:
What is your first and last name?
  [Unknown]:
What is the name of your organizational unit?
  [Unknown]:
What is the name of your organization?
  [Unknown]:
What is the name of your City or Locality?
  [Unknown]:
What is the name of your State or Province?
  [Unknown]:
What is the two-letter country code for this unit?
  [Unknown]:
Is CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=Unknown correct?
  [no]:
```

Enter information in the fields, or don’t.

Configuring and Starting the Starphish Backend.

First, create the password value for msfrpcd.

This will be part of the backend login process.

In the root directory /starphish-master/ edit the file “ds-start.bash”

Change the password value *(without the triangle brackets) then save and exit the file.

```
sudo msfrpcd -f -a 127.0.0.1 -P <passwordvalue> &
```

```
sudo msfrpcd -f -a 127.0.0.1 -P dingleberry &
```

Now, run the “ds-start.bash” file

sudo ./ds-start.bash

```
[*] MSGRPC ready at 2018-01-21 00:16:13 +0000.
```

Check to make sure the services are listening

ss –atnp

```
State       Recv-Q Send-Q  Local Address:Port                 Peer Address:Port
LISTEN      0      128           0.0.0.0:22                        0.0.0.0:*
LISTEN      0      5             0.0.0.0:5432                      0.0.0.0:*
LISTEN      0      128         127.0.0.1:55553                     0.0.0.0:*
```

You will notice port **55553** is now open.

Next configure dsconfig.py which is going to manage your connection with the backend.

Lhost - this is configured for port fowarding features should remain 127.0.0.1

Mhost - callback address for meterpreter payloads, this can be a name or IP address

Rhost - same as mhost but can be IP address only

Mport- the port used by mhost to listen for callbacks

Rootport - the port used by rhost to listen for callbacks

winport - alternative listening port for windows based devices

drupalport - alternative listening port for linux based devices

msfpass – the password the was created earlier in the “ds-start.bash” file

and example of a config file:

```
mhost = "35.182.111.78"

#Root host callback, must be ip address
#Used for mettle, mettle for android post exploitation cannot resolve names, we think.
#example: rhost="192.168.1.1"
rhost= "35.182.111.78"

#Meterpreter port for Android
#Example mport = 4444
mport = 4444

#Port used for rooted phones, windows devices, and linux (drupal) devices, mettle, same exampl
e as mport
rootport = 5555
winport = 8888
drupalport = 9999

#msfrpcd password
#password for msfrpcd, example msfpass = "thematrixhasyouneo"

msfpass = "dingleberry"
```

Once these port numbers are configured, they must be added to the security group or firewall your instance has been configured with, these ports are inbound.

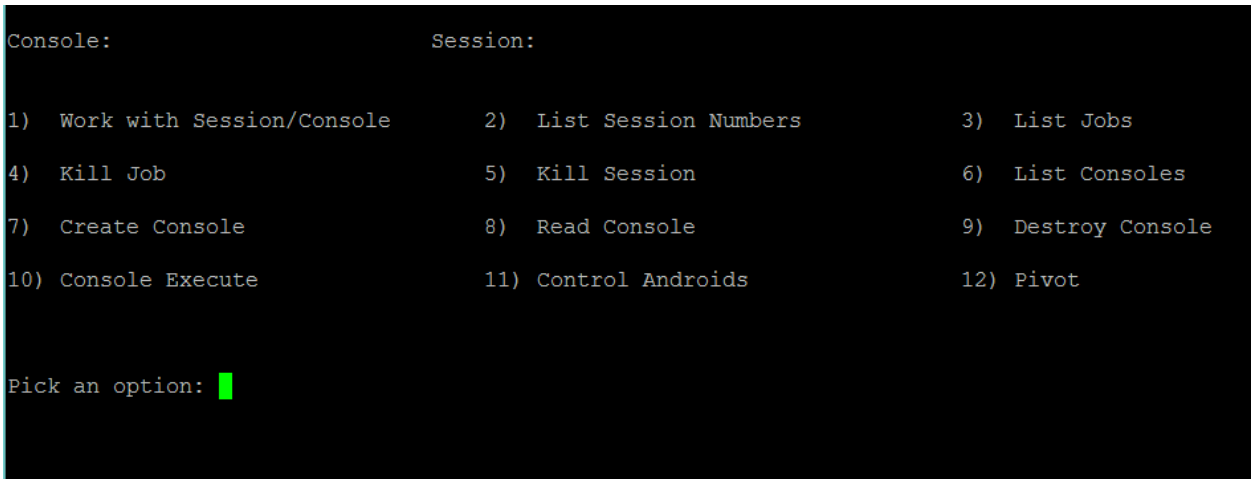
Type	Protocol	Port Range	Source
Custom TCP Rule	TCP	8000	0.0.0.0/0
SSH	TCP	22	0.0.0.0/0
Custom TCP Rule	TCP	4444	0.0.0.0/0
Custom TCP Rule	TCP	5555	0.0.0.0/0

*Note that port 8000 is used to deploy over HTTP for testing purposes.

Now,

Run ds-control.py, if setup correctly you should see the dashboard.

./ds-control.py



**Notes:

Ds-start.bash must be started on each boot before you can run ds-control.py

Dsconfig.py needs all options filled out even if not being used