Instructions to Build a Kippo Honeypot on Amazon EC2

I. Amazon EC2 config

1. Create an Amazon EC2 instance with the following specs:
   - Small instance with default 8GB of RAM
   - Use the “Open FW” Security Group
   - Accept all the other default settings
   - Assign the new instance a static IP address under the “Elastic IPs” section of the site.
2. Launch the newly created instance and log in using the instructions provided by Amazon once the instance is launched. This will only be used during the initial login.
3. Update the system:
   - `sudo apt-get update`
4. Setup unattended-upgrades to automatically update the system daily
   a. `sudo apt-get install unattended-upgrades`
   b. `su root`
   c. `crontab -e`
   d. Add a similar line to the end of the crontab file
      - `22 20 * * * /usr/bin/unattended-upgrades`
   e. The results of unattended-upgrades will be logged to `/var/log/unattended-upgrades`
5. Create the root password:
   a. Type `su` enter
   b. Type `passwd` and enter a complex password twice
6. Create new SSH keys and delete the key used in step 2 using the following commands:
   a. `cp authorized_keys authorized_keys.orig`
   b. Go into authorized_keys and delete all entries inside
   c. Create new SSH keys:
      i. `ssh-keygen -t rsa`
      ii. enter a unique complex password that differs from the root password
   d. Two new files will be created: **id_rsa** and **id_rsa.pub**
   e. Copy the text inside of id_rsa.pub into the authorized_keys file
   f. Copy the id_rsa Private Key text to a “Text Wrangler” or “Notepad++” new document on your local system and save using any suitable file name and select “All Files” instead of “.txt”.
7. Use Puttygen to convert the Private Key from **step 6f** into a Putty recognized Private Key by using “Conversions > Import Key” feature along with the password created in **step 5b**. Save the newly generated Putty Private Key to your local system.
8. Add the Putty Private key to Pageant, again using the password created in **step 5b**.
9. Attempt to connect via Putty using the static IP address assigned, port **22**, user name **ubuntu**, and ensure **allow agent forwarding** is checked.
10. Create user **kippouser**, assign groups and set a new password:
    a. `adduser kippouser`
    b. `usermod -a -G adm,dialout,audio,dip,video,plugdev,netdev,admin kippouser`
    c. type `groups kippouser` to verify the appropriate groups have been assigned.
11. Add `AllowUser` to the end of the sshd_config file along with user **kippouser**
    a. `sudo nano /etc/ssh/sshd_config`
    b. `add "AllowUser kippouser"` to the end of the file
12. Make the following changes to the **/etc/ssh/sshd_config** file as well:
    a. Change `Port` to **2222**
    b. Change `PermitRootLogin` to **no**
    c. Ensure `PermitEmptyPasswords` is **no**
    d. Save and exit
13. Restart ssh: `service ssh restart`
14. As the Ubuntu user cat the `~/.ssh/authorized_keys` and copy the public key inside the file
15. Change user to **kippouser**: `su kippouser` and enter the password
16. `cd ~/.ssh`. If the directory does not exist you must create the directory then create the ssh_authorized key file with the proper permissions
    a. `mkdir .ssh`
    b. `chmod 700 .ssh`
       - `drwx------ 2 kippouser kippouser 4096 Feb 21 18:59 .ssh`
    c. `cd .ssh`
    d. `touch authorized_keys`
    e. `chmod 600 authorized_keys`

```
-rw------- 1 kippouser kippouser  412 Feb 21 19:00 authorized_keys
```

f. `nano authorized_keys`

g. Paste the public key copied from the Ubuntu user authorized_keys file into the newly created kippouser authorized_keys file and save.

17. Open Putty and Load the saved Kippo profile and make the following changes:
    a. Change user name to **kippouser**
    b. Change port to **2222**
    c. Save and Load and verify that kippouser is able to ssh to the kippo host

II. Install Kippo

1. Install all required software packages used by kippo:
    a. `sudo apt-get install python-dev openssl python-openssl python-pyasn1 python-twisted`
    b. `sudo apt-get install subversion`
    c. `sudo apt-get install authbind`
2. `adduser kippo`
3. Add kippo to the list of users that can use the sudo command: `visudo`
4. where we add the line `kippo ALL=(ALL:ALL) ALL` under the "root" user.
5. We finish the required steps for using port 22:
    a. `touch /etc/authbind/byport/22`
    b. `chown kippo:kippo /etc/authbind/byport/22`
    c. `chmod 777 /etc/authbind/byport/22`
6. At this point, enter the system as **kippo** user and go to the **/home directory**.
7. Download the latest Kippo version from SVN:
    a. `svn checkout http://kippo.googlecode.com/svn/trunk/ ./kippo`
8. Change the port in Kippo's configuration file from **2222** to **22**:
    a. `mv kippo.cfg.dist kippo.cfg`
    b. `nano kippo.cfg`
9. Finally, edit the Kippo start script:
    a. `nano start.sh`
10. Change the following command from `twistd -y kippo.tac -l log/kippo.log --pidfile kippo.pid` to

```
authbind --deep twistd -y kippo.tac -l log/kippo.log --pidfile kippo.pid
```

so that it uses `authbind` to **listen** on port 22, and run the honeypot: `./start.sh`

11. We check that our port has actually opened and Kippo is "listening": `sudo netstat -antp` where there should be a line like this:

```
tcp 0 0 0.0.0.0:22 0.0.0.0:* LISTEN 22627/python
```

III. Hardening the system

The following steps are a derivation of the SANS Institute "Security Consensus Operational Readiness Evaluation." http://www.sans.org/score/checklists/linuxchecklist.pdf

1. Perform a "sudo netstat –antp" to view all open ports. Your results should be similar to the figure below. The only ports that should be open at this time are ports 22 (kippo python script listening port) and 2222 (kippouser actual ssh port).

```
Active Internet connections (servers and established)
Proto Recv-Q Send-Q Local Address           Foreign Address         State       PID/Program name
tcp        0      0 0.0.0.0:2222            0.0.0.0:*               LISTEN      2936/sshd
tcp        0      0 0.0.0.0:22              0.0.0.0:*               LISTEN      12327/python
tcp        0    404 10.210.245.221:2222     174.46.232.2:34675      ESTABLISHED 12762/sshd: kippous
tcp        0      0 10.210.245.221:2222     174.46.232.2:20985      ESTABLISHED 3363/sshd: kippouse
tcp6       0      0 :::2222                 :::*                    LISTEN      2936/sshd
```

2. Check for security on key files
    a. Ensure owner and group are set to root root and permissions are set to 0644 (-rw-r—r--)

```
ubuntu@domU-12-31-39-09-F2-13:~$ ls -lrt /etc/fstab
-rw-r--r-- 1 root root 156 Feb 20 19:54 /etc/fstab
```

    b. Verify that /etc/passwd & /etc/group are owned by root and that the permissions on /etc/passwd & /etc/group are 644 (-rw-r--r--)

```
ubuntu@domU-12-31-39-09-F2-13:~$ ls -lrt /etc/passwd
-rw-r--r-- 1 root root 1044 Jan 24 07:08 /etc/passwd
ubuntu@domU-12-31-39-09-F2-13:~$ ls -lrt /etc/group
-rw-r--r-- 1 root root 681 Jan 24 07:08 /etc/group
```

    c. Verify the persmissions on /etc/shadow are 400 (-r--------)

```
ubuntu@domU-12-31-39-09-F2-13:~$ ls -lrt /etc/shadow
-r-------- 1 root shadow 874 Feb 20 20:02 /etc/shadow
```

IV. Transferring Logs to the SSH Jumphost

1. The following **pulllogs.sh** script should reside on the SSH Jumphost in the sdcuser directory:

```
#!/bin/bash
cd /home/sdcuser/kippo
#sudo rm -rf /home/ubuntu/kippo/log/*
#sudo rm -rf /home/ubuntu/kippo/tty/*
rm -rf /home/sdcuser/kippo/log/*
rm -rf /home/sdcuser/kippo/tty/*
#scp -P 2222 -r ubuntu@10.190.198.22:/tmp/kippo/log/* /home/ubuntu/kippo/log/
scp -i /home/sdcuser/.ssh/id_rsa_arst -P 2222 -r kippouser@54.235.205.46:/tmp/kippo/log/* /home/sdcuser/kippo/log/
scp -i /home/sdcuser/.ssh/id_rsa_arst -P 2222 -r kippouser@54.235.205.46:/tmp/kippo/tty/* /home/sdcuser/kippo/tty/
chmod 666 /home/sdcuser/kippo/log/*
chmod 666 /home/sdcuser/kippo/tty/*
```

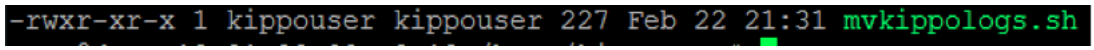
   and ensure the owner and group are sdcuser:
   - 

2. As sdcuser, create a cron job to run this script:
   a. `Crontab -e`
   b. Add the following line to the end of the `crontab -e` file

```
5,30 * * * * /home/sdcuser/pulllogs.sh
```

3. The following **mvkippologs.sh** script should reside on the Kippo honeypot in the kippouser directory:

```
#!/bin/bash
rm -rf /tmp/kippo/
mkdir /tmp/kippo/
mkdir /tmp/kippo/log
mkdir /tmp/kippo/tty
cp /home/kippo/kippo/log/kippo* /tmp/kippo/log/
cp /home/kippo/kippo/log/tty/* /tmp/kippo/tty/
chown -R kippouser:kippouser /tmp/kippo/
```

   and ensure the owner and group are set to kippouser:
   - 

4. As root, create a cron job to run this script:
   a. `Crontab -e`
   b. Add the following line to the end of the `crontab -e` file

```
0,25 * * * * /home/kippouser/mvkippologs.sh
```

V. References

- http://how-to.linuxcareer.com/deployment-of-kippo-ssh-honeypot-on-ubuntu-linux
- http://bruteforce.gr/kippo-reveals-itself-with-w-and-uptime-commands.html

-- MikeSt - 30 Dec 2015

I	Attachment	Action	Size	Date	Who	Comment
	Kippo1.png	manage	12.1 K	30 Dec 2015 - 22:41	MikeSt	
	Kippo2.png	manage	12.1 K	30 Dec 2015 - 22:42	MikeSt	
	Kippo3.png	manage	98.2 K	30 Dec 2015 - 22:42	MikeSt	
	Kippo4.png	manage	21.7 K	30 Dec 2015 - 22:43	MikeSt	
	Kippo5.png	manage	43.1 K	30 Dec 2015 - 22:43	MikeSt	
	Kippo6.png	manage	22.3 K	30 Dec 2015 - 22:43	MikeSt	
	Kippo7.png	manage	12.9 K	30 Dec 2015 - 22:44	MikeSt	
	Kippo8.png	manage	14.0 K	30 Dec 2015 - 22:44	MikeSt	

This topic: KnowledgeBase > WebHome > KBkippoInstalonAmazon

Topic revision: 30 Dec 2015, MikeSt