Defend Your Infrastructure from Evil with Kippo/Cowrie Honeypot

Troy Mitchell – Senior Cyber Security Engineer

Today's Presentation

- Today you will be learning about Pickles, Snails and Honey!

- Q: Why did the bee get married?
- A: Because he found his honey!

source: http://www.jokes4us.com/animaljokes/beejokes.html

Disclaimers

- All opinions expressed are my own and not those of any past or present employer. The techniques and tools demonstrated today, can cause damage if misused.
- Only use these tools and techniques on systems and networks you and/or your company own… or have express, written permission to test by the owner or an authorizing party of the systems.

Agenda

- Installation / Configuration / Customizing / Maintenance
  - Cowrie Honeypot – MySQL – Kippo-Graph
- Attacking Cowrie Honeypot or Brute forcing Attacks on SSH
- Cowrie Honeypot – Data Analysis
- Honeytokens (Canarytokens by Thinkst)

History of Honeypots

- Honeypots have a unique history.
- The concept has been around for 28+ years.
- The following list summarizes some key events in the history of honeypots.

History of Honeypots

- **1990/1991— First public works** documenting honeypot concepts—Clifford Stoll's *The Cuckoo's Egg* and Bill Cheswick's "An Evening With Berferd."
- **1997—** Version 0.1 of Fred Cohen's **Deception Toolkit** was released, one of the first honeypot solutions available to the security community.
- **1998—** Development began on CyberCop Sting, one of the **first commercial** honeypots sold to the public. CyberCop Sting introduces the concept of multiple, virtual systems bound to a single honeypot.
- **1998—** Marty Roesch and GTE Internetworking begin development on a honeypot solution that eventually becomes NetFacade. This work also begins the concept of **Snort**.

Chronology sourced from: Spitzner, L. (2003). Honeypots: tracking hackers. Boston: Addison-Wesley.

History of Honeypots

- **1998—** BackOfficer Friendly is released—a free, simple-to-use Windows-based honeypot that introduced many people, including me, to honeypot concepts.
- **1999—** Formation of the Honeynet Project and publication of the "Know Your Enemy" series of papers. This work helped increase awareness and validate the value of honeypots and honeypot technologies.
- **2000/2001—** Use of honeypots to capture and study worm activity. More organizations adopting honeypots for both detecting attacks and **for researching new threats**.
- **2002—** A honeypot is used to **detect and capture** in the wild a new and unknown attack, specifically the Solaris dtspcd exploit.

Chronology sourced from: Spitzner, L. (2003). Honeypots: tracking hackers. Boston: Addison-Wesley.

Kippo Honeypot

- Kippo is a medium interaction SSH honeypot designed to log brute force attacks and, most importantly, the entire shell interaction performed by the attacker.
- Kippo is developed by Upi Tamminen (desaster).
  - https://github.com/desaster/kippo

Cowrie Honeypot

- Cowrie is a medium interaction SSH and **Telnet** honeypot designed to log brute force attacks and the shell interaction performed by the attacker.
- In the words of Michel Oosterhof:
  "Cowrie is a snail yes :) It's a shell to be particular. And it's emulating a shell. (a UNIX shell). It also uses another component called Conch (another shell) which is part of Twisted."
- Cowrie is developed by Michel Oosterhof.
  - https://github.com/micheloosterhof/cowrie

Cowrie - A Research Honeypot

- Cowrie is a Research honeypot that is designed to gain intel on attackers.
- Cowrie IS NOT a firewall.
- Cowrie IS a system that provides DATA that helps you protect your infrastructure.
  - IP Addresses
  - Usernames and Passwords
  - SSH Client Version (Banners)

Installation Type

- Raspberry PI
  - 3 MODEL B
- Virtual Machines
  - VMware Workstation
  - VMware ESXi
  - Oracle VM Virtualbox
- Docker Containers

Cowrie – New Features

SFTP support

Cowrie now supports the SFTP protocol to **upload** and **download** files. Uploaded files are placed into the **'dl/' directory,** like files that were downloaded by 'wget'. SFTP offers a file system interface to the pickled fs and you can list any file available in there.
Downloads are also supported, if the contents are in honeyfs. Programmatically it is based on the Conch code, and uses a UNIX file system like interface to the pickled file system.

```
scp linuxprivchecker.py rocky@192.168.1.120:~
```

Cowrie – New Features (cont.)

Exec support

One way to run remote commands is ‘exec’ commands. Like so:

This includes logging and executing these commands.

```
ssh user@host 'cat /etc/passwd'
```

```
cat malware | ssh user@host 'cat>./malware;chmod u+x ./malware; ./malware; rm -f ./malware'.
```

Kippo-Graph

- Kippo-Graph is a full featured script to visualize statistics for a Kippo based SSH honeypot.
- Kippo-Graph is developed by Ioannis “Ion” Koniaris.
  - https://github.com/ikoniaris/kippo-graph
  - https://bruteforcelab.com/kippo-graph

Installation

Installation

- Cowrie Honeypot Installation
  - 23 steps.
- MySQL Installation
  - 11 steps.
- Kippo-Graph Installation
  - 8 steps.
- Live Demo – Deploying the Google Drive Cowrie/Kippo-Graph (OVA)

Cowrie Honeypot Installation

Step # 1
sudo apt-get update && sudo apt-get upgrade
Step # 2
sudo apt-get install git python-virtualenv libssl-dev libffi-dev build-essential libpython-dev python2.7-minimal authbind
Step # 3
sudo apt-get install libmysqlclient-dev
Step # 4
sudo sed -i 's/^Port .*/Port 9022/g' /etc/ssh/sshd_config
Step # 5
sudo service ssh restart
Step # 6
sudo adduser --disabled-password cowrie
Step #7
sudo su - cowrie

Cowrie Honeypot Installation (cont.)

Step # 8
git clone https://github.com/**tmitchell5280**/cowrie.git
Step # 9
cd cowrie
Step # 10
virtualenv cowrie-env
Step # 11
source cowrie-env/bin/activate
Step # 12
pip install --upgrade pip
Step # 13
pip install --upgrade -r **requirements.txt**
Step # 14
python path here – skip for now

Cowrie Honeypot Installation

Step # 15
cp cowrie.cfg.dist cowrie.cfg
Step # 16
exit
Step # 17
sudo iptables -t nat -A PREROUTING -p tcp --dport 22 -j REDIRECT --to-port 2222
Step # 18
sudo apt-get install iptables-persistent
Save current IPv4 rules? **Yes**
Save current IPv6 rules? **Yes**

Cowrie Honeypot Installation

Step # 19
sudo su - cowrie
Step # 20
cd cowrie
Step # 21
source cowrie-env/bin/activate
Step # 22
bin/cowrie start
Step # 23
exit

We are done with the basic Cowrie Installation.

If you are planning to use Kippo-Graph, please continue!

MySQL Installation

Step # 1
sudo apt-get install mysql-server
New password for the MySQL "**root**" user:
Select the **OK** button.
Repeat password for the MySQL "**root**" user:
Select the **OK** button.
Step # 2
sudo apt-get install mysql-client
Step # 3
sudo service mysql start
Step # 4
mysql -u root -p

MySQL Installation (cont.)

Step # 5
Enter your MySQL **root** password here.
mysql>
CREATE DATABASE cowrie;
GRANT ALL ON cowrie.* TO 'cowrie'@'localhost' IDENTIFIED BY **'PASSWORD HERE'**;
FLUSH PRIVILEGES;
exit
Step # 6
cd /home/cowrie/cowrie/
Step # 7
mysql -u cowrie -p

MySQL Installation (cont.)

Step # 8
Enter your MySQL **cowrie** password here.
mysql>
USE cowrie;
source ./doc/sql/mysql.sql;
exit
Step # 9
sudo nano -c /home/cowrie/cowrie/cowrie.cfg
Activate output to mysql
[output_mysql] - (line # 416)
host = localhost – (line # 417)
database = **cowrie** (line # 418)
username = **cowrie** (line # 419)
password = **PASSWORD HERE** (line # 420)
port = 3306 (line # 421)
debug = **false** (line # 422)

MySQL Installation (cont.)

Step # 10
sudo apt-get install acl
Step # 11
sudo setfacl -Rm g:www-data:rx /home/cowrie/cowrie/log/tty/

Kippo-Graph Installation

Step # 1
sudo apt-get update && sudo apt-get install -y libapache2-mod-php5 php5-mysql php5-gd php5-curl
Step # 2
sudo /etc/init.d/apache2 restart
Step # 3
sudo git clone https://github.com/**tmitchell5280**/kippo-graph.git
Step # 4
sudo mv kippo-graph /var/www/html
cd /var/www/html
cd kippo-graph
Step # 5
sudo chmod 777 generated-graphs
Step # 6
sudo cp config.php.dist config.php

Kippo-Graph Installation (cont.)

Step # 7
sudo **nano -c** config.php
Change db settings
define('DB_HOST', 'localhost'); (Line #22)
define('DB_USER', '**cowrie**'); (Line #23)
define('DB_PASS', '**PASSWORD HERE**'); (Line #24)
define('DB_NAME', '**cowrie**'); (Line #25)
define('DB_PORT', '3306'); (Line #26)

Kippo-Graph Installation (cont.)

Step # 8
MySQL Commands
mysql -u **cowrie** -p
Enter your MySQL Cowrie password here.
mysql>
USE cowrie;
show tables;
show columns from clients;
exit

Securing Kippo-Graph

Install the Apache Utilities Package

```
sudo apt-get update

sudo apt-get install apache2 apache2-utils
```

Securing Kippo-Graph (Cont.)

Create the Password File

sudo htpasswd -c /etc/apache2/.htpasswd **username**

You will be asked to supply and confirm a password for the user.

Example:

```
sudo htpasswd -c /etc/apache2/.htpasswd tmitchell
New password:
Re-type new password:
Adding password for user tmitchell
```

Leave out the -c argument for any additional users you wish to add:

```
sudo htpasswd /etc/apache2/.htpasswd another_user
```

```
cat /etc/apache2/.htpasswd
```

```
tmitchell:$apr1$jFGW9RBK$AX.4VMZ3mW1aYiijIsubX0
another_user:$apr1$p1E9MeAf$kiAhneUwr.MhAE2kKGYHK
```

Securing Kippo-Graph (cont.)

Configure Apache Password Authentication

sudo nano /etc/apache2/apache2.conf

Find the <Directory> block for the /var/www directory that holds the document root. Turn on .htaccessprocessing by changing the AllowOverride directive within that block from "None" to "All": Around line # 166/167

```
/etc/apache2/apache2.conf
. . .
<Directory /var/www/>
   Options Indexes FollowSymLinks
   AllowOverride All
   Require all granted
</Directory>
```

Securing Kippo-Graph (cont.)

We need to add an .htaccess file to the directory we wish to restrict.

```
sudo nano -c /var/www/html/kippo-graph/.htaccess
```

Default file contents, leave in place.

```
<IfModule mod_rewrite.c>
   RewriteEngine On
   RewriteRule ^results\.php$ http://www.garyshood.com/virus/results.php$1
[R=301,L]
</IfModule>
```

Move to next Step!

Securing Kippo-Graph (cont.)

Add these lines above the Default file contents

```
AuthType Basic
AuthName "Restricted Content"
AuthUserFile /etc/apache2/.htpasswd
Require valid-user
```

Final file contents: Your .htaccess file should look just like this one.

```
AuthType Basic
AuthName "Restricted Content"
AuthUserFile /etc/apache2/.htpasswd
Require valid-user

<IfModule mod_rewrite.c>
   RewriteEngine On
   RewriteRule ^results\.php$ http://www.garyshood.com/virus/results.php$1
[R=301,L]
</IfModule>
```

```
sudo service apache2 restart
```

Configuration

Configuration

General Honeypot Options - hostname

```
# Hostname for the honeypot. Displayed by the shell prompt of the virtual
# environment
#
# (default: svr04)
hostname = svr04
```

Cowrie Honeypot - Hostname from default (svr04) (Around line #23/24)

File Location: /home/cowrie/cowrie$

File Name: **cowrie.cfg**

Use a name that is as realistic as possible.

Using a prefix like city / airport code: SFO

SFO-Server01

DEN-Production-SRV01

SFO-17241-Server01

You will need to either start or restart Cowrie for changes to take effect.

Configuration

General Honeypot Options - contents_path

```
# Directory where virtual file contents are kept in.
#
# This is only used by commands like 'cat' to display the contents of files.
# Adding files here is not enough for them to appear in the honeypot – the
# actual virtual filesystem is kept in filesystem_file (see below)
#
# (default: honeyfs)contents_path = honeyfs
```

Configuration

General Honeypot Options - filesystem_file

```
# File in the Python pickle format containing the virtual filesystem.
#
# This includes the filenames, paths, permissions for the Cowrie filesystem,
# but not the file contents. This is created by the bin/createfs utility from
# a real template linux installation.
#
# (default: fs.pickle)
filesystem_file = data/fs.pickle
```

Configuration

General Honeypot Options - interactive_timeout

```
# Interactive timeout determines when logged in sessions are
# terminated for being idle. In seconds.
# (default: 180)
interactive_timeout = 360
```

Configuration

SSH Specific Options - version

```
# SSH Version String
#
# Use these to disguise your honeypot from a simple SSH version scan
# Examples:# SSH-2.0-OpenSSH_5.1p1 Debian-5# SSH-1.99-OpenSSH_4.3
# SSH-1.99-OpenSSH_4.7
# SSH-1.99-Sun_SSH_1.1
# SSH-2.0-OpenSSH_4.2p1 Debian-7ubuntu3.1
#
# (default: "SSH-2.0-SSH-2.0-OpenSSH_6.0p1 Debian-4+deb7u2")
version =SSH-2.0-OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.7
```

Configuration

Default User Names

Which Usernames and password, will Cowrie Honeypot except as valid:

File Location: /home/cowrie/cowrie/data$

File Name: **userdb.txt**

These are the default ones that come with Cowrie out of the box:

root:x:!root
root:x:!123456
root:x:*
richard:x:*
richard:x:fout

Note the Exclamation mark / Bang (Will not allow access with this password).

Update - This currently does not work - **BUG**.

Configuration

Default User Names (cont.)

Note: the Asterisk (Will allow any password entered with the Username in front of it).

root:x:*
richard:x:*

With these settings, no will be able to connect: I recommend it this way to start off.
#root:x:!root
#root:x:!123456
#root:x:*
#richard:x:*
#richard:x:fout
With a leading number sign, hash, or pound sign.

This way you can get familiar with the Cowrie Honeypot system.
Basically in **logging mode only**.
You can change and add Username and Passwords anytime without a Cowrie Service restart.

Customizing

Customizing

Pickle File System – File Shares

- Shares
  - Accounting
  - Finance
  - Human Resources
  - IT Support
  - Legal Department
  - Marketing
  - Operations
  - Production
  - Purchasing
  - Sales

Customizing

Pickle File System – Home Directories

- Home Directories
  - Many Users
- Scripts

Customizing

Pickle File System – Usernames and Passwords

- etc/passwd File
- etc/shadow File

- Scripts
  - Create Users
- Fake Name Generator

Maintenance

- Ubuntu Server (Host System) – apt-get install updates
- MySQL Database Backups
  - Cowrie Database

Pickle File System Editor

File Name and Location:

```
tmitchell@ubuntu:/home/cowrie/cowrie/bin$ ./fsctl
Usage: fsctl <fs.pickle>
tmitchell@ubuntu:/home/cowrie/cowrie/bin$
```

Command Example:

```
/home/cowrie/cowrie/bin$ ./fsctl /home/cowrie/cowrie/data/basic-starter-users-file-system-
01.pickle
/home/cowrie/cowrie/data/basic-starter-users-file-system-01.pickle

Kippo/Cowrie file system interactive editor
Donovan Hubbard, Douglas Hubbard, March 2013
Type 'help' for help

basic-starter-users-file-system-01.pickle:/$
```

Starting Cowrie Honeypot

- **Starting Cowrie Honeypot**
  - sudo su - cowrie
  - cd cowrie
  - source ./cowrie-env/bin/activate
  - (cowrie-env)cowrie@ubuntu:~/cowrie$ bin/cowrie start

```
mjones@ubuntu:~$ sudo su - cowrie
cowrie@ubuntu:~$ cd cowrie/
cowrie@ubuntu:~/cowrie$ source ./cowrie-env/bin/activate
(cowrie-env)cowrie@ubuntu:~/cowrie$ bin/cowrie start
Using activated Python virtual environment "/home/cowrie/cowrie/cowrie-env"
Starting cowrie: [twistd   --umask 0022 --pidfile var/run/cowrie.pid --logger cowrie.python.logfile.logger cowrie ]...
(cowrie-env)cowrie@ubuntu:~/cowrie$
```

Stopping Cowrie Honeypot

- **Stopping Cowrie Honeypot**
  - sudo su - cowrie
  - cd cowrie
  - source ./cowrie-env/bin/activate
  - (cowrie-env)cowrie@ubuntu:~/cowrie$ bin/cowrie stop

```
mjones@ubuntu:~$ sudo su - cowrie
cowrie@ubuntu:~$ cd cowrie/
cowrie@ubuntu:~/cowrie$ source ./cowrie-env/bin/activate
(cowrie-env)cowrie@ubuntu:~/cowrie$ bin/cowrie stop
Stopping cowrie...
(cowrie-env)cowrie@ubuntu:~/cowrie$
```

Restarting Cowrie Honeypot

- **Restarting Cowrie Honeypot**
  - sudo su - cowrie
  - cd cowrie
  - source ./cowrie-env/bin/activate
  - (cowrie-env)cowrie@ubuntu:~/cowrie$ bin/cowrie restart

```
mjones@ubuntu:~$ sudo su - cowrie
cowrie@ubuntu:~$ cd cowrie/
cowrie@ubuntu:~/cowrie$ source ./cowrie-env/bin/activate
(cowrie-env)cowrie@ubuntu:~/cowrie$ bin/cowrie restart
Stopping cowrie...
Using activated Python virtual environment "/home/cowrie/cowrie/cowrie-env"
Starting cowrie: [twistd   --umask 0022 --pidfile var/run/cowrie.pid --logger cowrie.python.logfile.logger cowrie ]...
(cowrie-env)cowrie@ubuntu:~/cowrie$
```

Status Checking Cowrie Honeypot

- **Status Checking the Cowrie Honeypot Service State**
  - sudo su - cowrie
  - cd cowrie
  - source ./cowrie-env/bin/activate
  - (cowrie-env)cowrie@ubuntu:~/cowrie$ bin/cowrie status

```
mjones@ubuntu:~$ sudo su - cowrie
cowrie@ubuntu:~$ cd cowrie/
cowrie@ubuntu:~/cowrie$ source ./cowrie-env/bin/activate
(cowrie-env)cowrie@ubuntu:~/cowrie$ bin/cowrie status
cowrie is running (PID: 2163).
(cowrie-env)cowrie@ubuntu:~/cowrie$
```

Attacking

Attacking Cowrie Honeypot

- Brute force SSH – Hydra
- Brute force SSH – Medusa (Live Demo)
- Brute force SSH – Ncrack
- Brute force SSH - Nmap

Brute force SSH – Tools

- **Test** all of these tools in a lab environment.
  - Review the results.
    - From each of the tools.
    - Also from Cowrie (**Logging**), including Kippo-Graph (**Website**).
- Read the **man pages**.
  - They are the user manual.
- Measure speed and performance of the tools.
- One tool fits all, does not apply here.

Brute force SSH – Hydra

With specific Username and Password List

```
hydra -l root -P passwords.txt 192.168.1.120 ssh
```

With Username List and Password List

```
hydra -L usernames.txt -P passwords.txt 192.168.1.120 ssh
```

With Username List and Password List – Multiple IP Addresses

```
hydra -L usernames.txt -M ssh_host_list.txt -P passwords.txt ssh
```

Brute force SSH – Medusa

With specific Username and Password List

```
medusa -u root -P passwords.txt -h 192.168.1.120 -M ssh
```

With Username List and Password List

```
medusa -U usernames.txt -P passwords.txt -h 192.168.1.120 -M ssh
```

With Username List and Password List - Multiple IP Addresses

```
medusa -U usernames.txt -P passwords.txt -H ssh_host_list.txt -M ssh
```

Brute force SSH – Ncrack

With specific Username and Password List

```
ncrack -u root -P passwords.txt 192.168.1.120 -p 22
```

With Username List and Password List

```
ncrack -U usernames.txt -P passwords.txt 192.168.1.120 -p 22
```

With Username List and Password List - Multiple IP Addresses

```
ncrack -U usernames.txt -P passwords.txt -iL ssh_host_list.txt -p 22
```

Brute force SSH - Nmap

With Username List and Password List

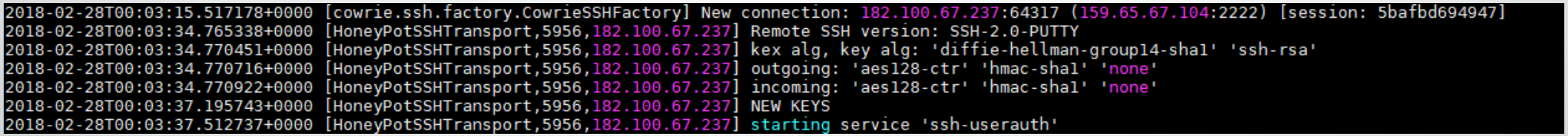
```
nmap --script ssh-brute --script-args userdb=./usernames.txt,passdb=./passwords.txt 192.168.1.120 -p 22
```

With Username List and Password List - Multiple IP Addresses

```
nmap --script ssh-brute --script-args userdb=./usernames.txt,passdb=./passwords.txt -iL ssh_host_list.txt -p22
```

Cowrie Log File – New Connection

MobaXterm - Screenshot



[cowrie.ssh.factory.CowrieSSHFactory] New connection: 182.100.67.237:64317 (159.65.67.104:2222) [session: 5bafbd694947]
[HoneyPotSSHTransport,5956,182.100.67.237] Remote SSH version: SSH-2.0-PUTTY
[HoneyPotSSHTransport,5956,182.100.67.237] kex alg, key alg: 'diffie-hellman-group14-sha1' 'ssh-rsa'
[HoneyPotSSHTransport,5956,182.100.67.237] outgoing: 'aes128-ctr' 'hmac-sha1' 'none'
[HoneyPotSSHTransport,5956,182.100.67.237] incoming: 'aes128-ctr' 'hmac-sha1' 'none'
[HoneyPotSSHTransport,5956,182.100.67.237] NEW KEYS
[HoneyPotSSHTransport,5956,182.100.67.237] starting service 'ssh-userauth'

Failed SSH Attempts

```
2018-02-28T00:17:34.307702+0000 [SSHService 'ssh-userauth' on HoneyPotSSHTransport,5965,182.100.67.237] 'root' trying auth 'none'
2018-02-28T00:17:34.576065+0000 [SSHService 'ssh-userauth' on HoneyPotSSHTransport,5965,182.100.67.237] 'root' trying auth 'password'
2018-02-28T00:17:34.576741+0000 [SSHService 'ssh-userauth' on HoneyPotSSHTransport,5965,182.100.67.237] login attempt [root/7758521] failed
2018-02-28T00:17:35.579756+0000 [-] 'root' failed auth 'password'
2018-02-28T00:17:35.580072+0000 [-] unauthorized login:
2018-02-28T00:17:47.232003+0000 [SSHService 'ssh-userauth' on HoneyPotSSHTransport,5965,182.100.67.237] 'root' trying auth 'password'
2018-02-28T00:17:47.232810+0000 [SSHService 'ssh-userauth' on HoneyPotSSHTransport,5965,182.100.67.237] login attempt [root/master123] failed
2018-02-28T00:17:48.236242+0000 [-] 'root' failed auth 'password'
2018-02-28T00:17:48.236528+0000 [-] unauthorized login:
2018-02-28T00:17:48.520230+0000 [SSHService 'ssh-userauth' on HoneyPotSSHTransport,5965,182.100.67.237] 'root' trying auth 'password'
2018-02-28T00:17:48.520754+0000 [SSHService 'ssh-userauth' on HoneyPotSSHTransport,5965,182.100.67.237] login attempt [root/dabe] failed
2018-02-28T00:17:49.523401+0000 [-] 'root' failed auth 'password'
2018-02-28T00:17:49.523707+0000 [-] unauthorized login:
2018-02-28T00:17:49.772913+0000 [HoneyPotSSHTransport,5965,182.100.67.237] Got remote error, code 11
        reason:
2018-02-28T00:17:49.773685+0000 [HoneyPotSSHTransport,5965,182.100.67.237] connection lost
2018-02-28T00:17:49.773913+0000 [HoneyPotSSHTransport,5965,182.100.67.237] Connection lost after 37 seconds
```

[SSHService 'ssh-userauth' on HoneyPotSSHTransport,5965,182.100.67.237] 'root' trying auth 'none'
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,5965,182.100.67.237] 'root' trying auth 'password'
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,5965,182.100.67.237] login attempt [root/7758521] **failed**
[-] 'root' failed auth 'password'
[-] unauthorized login:
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,5965,182.100.67.237] 'root' trying auth 'password'
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,5965,182.100.67.237] login attempt [root/master123] **failed**
[-] 'root' failed auth 'password'
[-] unauthorized login:
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,5965,182.100.67.237] 'root' trying auth 'password'
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,5965,182.100.67.237] login attempt [root/dabe] **failed**
[-] 'root' failed auth 'password'
[-] unauthorized login:
[HoneyPotSSHTransport,5965,182.100.67.237] Got remote error, code 11
reason:
[HoneyPotSSHTransport,5965,182.100.67.237] connection lost
[HoneyPotSSHTransport,5965,182.100.67.237] Connection lost after 37 seconds

Cowrie Log File – Keyboard-interactive

[cowrie.ssh.factory.CowrieSSHFactory] New connection: 192.168.1.103:51926 (192.168.1.120:2222) [session: 5ead774255ba]
[-] output_mysql: MySQL query: INSERT INTO `sessions` (`id`, `starttime`, `sensor`, `ip`) VALUES (%s, FROM_UNIXTIME(%s), %s, %s) (u'5ead774255ba', 1520178973.006745, 1L, u'192.168.1.103')
[HoneyPotSSHTransport,6688,192.168.1.103] Remote SSH version: SSH-2.0-OpenSSH_7.6p1 Debian-2
[HoneyPotSSHTransport,6688,192.168.1.103] kex alg, key alg: 'ecdh-sha2-nistp256' 'ssh-rsa'
[HoneyPotSSHTransport,6688,192.168.1.103] outgoing: 'aes128-ctr' 'hmac-sha1' 'none'
[HoneyPotSSHTransport,6688,192.168.1.103] incoming: 'aes128-ctr' 'hmac-sha1' 'none'
[-] output_mysql: MySQL query: UPDATE `sessions` SET `client` = %s WHERE `id` = %s (36, u'5ead774255ba')
[HoneyPotSSHTransport,6688,192.168.1.103] NEW KEYS
[HoneyPotSSHTransport,6688,192.168.1.103] starting service 'ssh-userauth'
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,6688,192.168.1.103] 'root' trying auth 'none'
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,6688,192.168.1.103] 'root' trying auth **'keyboard-interactive'**
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,6688,192.168.1.103] output_mysql: MySQL query: INSERT INTO `auth` (`session`, `success`, `username`, `password`, `timestamp`) VALUES (%s, %s, %s, %s, FROM_UNIXTIME(%s)) (u'5ead774255ba', 0, u'root', u'hackme', 1520178985.302153)
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,6688,192.168.1.103] login attempt [root/hackme] failed
[-] 'root' failed auth 'keyboard-interactive'
[-] unauthorized login:

Cowrie Log – Exec Support

[cowrie.ssh.factory.CowrieSSHFactory] New connection: 192.168.1.103:52628 (192.168.1.120:2222) [session: c9f5b121230d]
[-] output_mysql: MySQL query: INSERT INTO `sessions` (`id`, `starttime`, `sensor`, `ip`) VALUES (%s, FROM_UNIXTIME(%s), %s, %s) (u'c9f5b121230d', 1520181694.725968, 1L, u'192.168.1.103')
[HoneyPotSSHTransport,6732,192.168.1.103] Remote SSH version: **SSH-2.0-OpenSSH_7.6p1 Debian-2**
[HoneyPotSSHTransport,6732,192.168.1.103] kex alg, key alg: 'ecdh-sha2-nistp256' 'ssh-rsa'
[HoneyPotSSHTransport,6732,192.168.1.103] outgoing: 'aes128-ctr' 'hmac-sha1' 'none'
[HoneyPotSSHTransport,6732,192.168.1.103] incoming: 'aes128-ctr' 'hmac-sha1' 'none'
[-] output_mysql: MySQL query: UPDATE `sessions` SET `client` = %s WHERE `id` = %s (36, u'c9f5b121230d')
[HoneyPotSSHTransport,6732,192.168.1.103] NEW KEYS
[HoneyPotSSHTransport,6732,192.168.1.103] starting service 'ssh-userauth'
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,6732,192.168.1.103] 'rocky' trying auth 'none'
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,6732,192.168.1.103] 'rocky' trying auth **'keyboard-interactive'**
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,6732,192.168.1.103] output_mysql: MySQL query: INSERT INTO `auth` (`session`, `success`, `username`, `password`, `timestamp`) VALUES (%s, %s, %s, %s, FROM_UNIXTIME(%s)) (u'c9f5b121230d', 1, u'**rocky**', u'**dellman**', 1520181717.216174)
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,6732,192.168.1.103] **login attempt [rocky/dellman] succeeded**
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,6732,192.168.1.103] 'rocky' authenticated with 'keyboard-interactive'
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,6732,192.168.1.103] starting service 'ssh-connection'
[SSHService 'ssh-connection' on HoneyPotSSHTransport,6732,192.168.1.103] got channel 'session' request
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6732,192.168.1.103] channel open
[SSHService 'ssh-connection' on HoneyPotSSHTransport,6732,192.168.1.103] got global no-more-sessions@openssh.com request

Cowrie Log – Exec Support (cont.)

[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6732,192.168.1.103] request_env: LANG=en_US.UTF-8
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6732,192.168.1.103] **executing command** "cat /etc/passwd"
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6732,192.168.1.103] Opening TTY Log: log/tty/**20180304-094159-c9f5b121230d-0e.log**
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6732,192.168.1.103] output_mysql: MySQL query: INSERT INTO `input` (`session`, `timestamp`, `success`, `input`) VALUES (%s, FROM_UNIXTIME(%s), %s , %s) (u'c9f5b121230d', 1520181719.198739, 1, u'cat /etc/passwd')
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6732,192.168.1.103] **CMD: cat /etc/passwd**
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6732,192.168.1.103] **Command found**: cat /etc/passwd
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6732,192.168.1.103] exitCode: 0
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6732,192.168.1.103] sending request 'exit-status'
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6732,192.168.1.103] sending close 0
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6732,192.168.1.103] remote close

Cowrie Log – Exec Support (cont.)

[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6732,192.168.1.103] output_mysql: MySQL query: INSERT INTO `ttylog` (`session`, `ttylog`, `size`) VALUES (%s, %s, %s) (u'c9f5b121230d', u'**log/tty/20180304-094159-c9f5b121230d-0e.log**', 1251)
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6732,192.168.1.103] Closing TTY Log: log/tty/20180304-094159-c9f5b121230d-0e.log after 0 seconds
[HoneyPotSSHTransport,6732,192.168.1.103] Got remote error, code 11
reason: disconnected by user
[HoneyPotSSHTransport,6732,192.168.1.103] avatar rocky logging out
[HoneyPotSSHTransport,6732,192.168.1.103] connection lost
[HoneyPotSSHTransport,6732,192.168.1.103] Connection lost after 24 seconds

Cowrie Log - SFTP support

[cowrie.ssh.factory.CowrieSSHFactory] New connection: 192.168.1.103:50845 (192.168.1.120:2222) [session: e4f08ea4bc6b]
[-] output_mysql: MySQL query: INSERT INTO `sessions` (`id`, `starttime`, `sensor`, `ip`) VALUES (%s, FROM_UNIXTIME(%s), %s, %s) (u'e4f08ea4bc6b', 1520182599.98988, 1L, u'192.168.1.103')
[HoneyPotSSHTransport,6744,192.168.1.103] Remote SSH version: **SSH-2.0-OpenSSH_7.6p1 Debian-2**
[HoneyPotSSHTransport,6744,192.168.1.103] kex alg, key alg: 'ecdh-sha2-nistp256' 'ssh-rsa'
[HoneyPotSSHTransport,6744,192.168.1.103] outgoing: 'aes128-ctr' 'hmac-sha1' 'none'
[HoneyPotSSHTransport,6744,192.168.1.103] incoming: 'aes128-ctr' 'hmac-sha1' 'none'
[-] output_mysql: MySQL query: UPDATE `sessions` SET `client` = %s WHERE `id` = %s (36, u'e4f08ea4bc6b')
[HoneyPotSSHTransport,6744,192.168.1.103] NEW KEYS
[HoneyPotSSHTransport,6744,192.168.1.103] starting service 'ssh-userauth'
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,6744,192.168.1.103] 'rocky' trying auth 'none'
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,6744,192.168.1.103] 'rocky' trying auth **'keyboard-interactive'**
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,6744,192.168.1.103] output_mysql: MySQL query: INSERT INTO `auth` (`session`, `success`, `username`, `password`, `timestamp`) VALUES (%s, %s, %s, %s, FROM_UNIXTIME(%s)) (u'e4f08ea4bc6b', 1, u'**rocky**', u'**dellman**', 1520182603.079561)

Cowrie Log - SFTP support (cont.)

[SSHService 'ssh-userauth' on HoneyPotSSHTransport,6744,192.168.1.103] login attempt [**rocky/dellman**] **succeeded**
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,6744,192.168.1.103] 'rocky' authenticated with 'keyboard-interactive'
[SSHService 'ssh-userauth' on HoneyPotSSHTransport,6744,192.168.1.103] starting service 'ssh-connection'
[SSHService 'ssh-connection' on HoneyPotSSHTransport,6744,192.168.1.103] got channel 'session' request
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6744,192.168.1.103] channel open
[SSHService 'ssh-connection' on HoneyPotSSHTransport,6744,192.168.1.103] got global no-more-sessions@openssh.com request
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6744,192.168.1.103] request_env: LANG=en_US.UTF-8
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6744,192.168.1.103] **executing command** "scp -t ~"
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6744,192.168.1.103] Opening TTY Log: log/tty/**20180304-095645-e4f08ea4bc6b-0e.log**
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6744,192.168.1.103] output_mysql: MySQL query: INSERT INTO `input` (`session`, `timestamp`, `success`, `input`) VALUES (%s, FROM_UNIXTIME(%s), %s , %s) (u'e4f08ea4bc6b', 1520182605.15719, 1, u'scp -t ~')
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6744,192.168.1.103] **CMD: scp -t ~**
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6744,192.168.1.103] **Command found**: scp -t ~
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6744,192.168.1.103] received eof, sending ctrl-d to command
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6744,192.168.1.103] exitCode: 0
[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6744,192.168.1.103] sending request '**exit-status**'

Cowrie Log - SFTP support (cont.)

[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6744,192.168.1.103] Not storing duplicate content **df908ddf409238272b0630e3e6181acffd8d4f55ee2873e64333da38ebafe9b7**

[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6744,192.168.1.103] output_mysql: MySQL query: INSERT INTO `downloads` (`session`, `timestamp`, `url`, `outfile`, `shasum`) VALUES (%s, FROM_UNIXTIME(%s), %s, %s, %s) (u'e4f08ea4bc6b', 1520182605.174677, u'stdin', u'**dl/df908ddf409238272b0630e3e6181acffd8d4f55ee2873e64333da38ebafe9b7**', u'df908ddf409238272b0630e3e6181acffd8d4f55ee2873e64333da38ebafe9b7')

[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6744,192.168.1.103] Saved stdin contents with **SHA-256** df908ddf409238272b0630e3e6181acffd8d4f55ee2873e64333da38ebafe9b7 to dl/df908ddf409238272b0630e3e6181acffd8d4f55ee2873e64333da38ebafe9b7

[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6744,192.168.1.103] output_mysql: MySQL query: INSERT INTO `ttylog` (`session`, `ttylog`, `size`) VALUES (%s, %s, %s) (u'e4f08ea4bc6b', u'log/tty/20180304-095645-e4f08ea4bc6b-0e.log', 10)

[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6744,192.168.1.103] Closing TTY Log: log/tty/20180304-095645-e4f08ea4bc6b-0e.log after 0 seconds

[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6744,192.168.1.103] sending close 0

[SSHChannel session (0) on SSHService 'ssh-connection' on HoneyPotSSHTransport,6744,192.168.1.103] remote close

[HoneyPotSSHTransport,6744,192.168.1.103] Got remote error, code 11
reason: disconnected by user

[HoneyPotSSHTransport,6744,192.168.1.103] avatar rocky logging out

[HoneyPotSSHTransport,6744,192.168.1.103] connection lost

[HoneyPotSSHTransport,6744,192.168.1.103] output_mysql: MySQL query: UPDATE `sessions` SET `endtime` = FROM_UNIXTIME(%s) WHERE `id` = %s (1520182605.183698, u'e4f08ea4bc6b')

[HoneyPotSSHTransport,6744,192.168.1.103] Connection lost after 5 seconds

Data Analytics

Data Analytics with Kippo-Graph

- Overview
- Input
- Playlog
- Network
- GEOIP
- Graph Gallery

Kippo-Graph - Overview

- Honeypot activity overview
  - Total login attempts
  - Distinct source IP addresses
  - Active time period
    - Start date (first attack)
    - End date (last attack)
- Graphical statistics generated from your honeypot database

Kippo-Graph – Overview (cont.)

- Top 10 passwords
- Top 10 usernames
- Top 10 user-pass combos
- Top 10 successful user-pass combos
- Top 10 SSH Clients

- Success ratio
- Successes per day/week
- Connections per IP
- Successful logins from the same IP
- Probes per day/week

Kippo-Graph - Input

- Input presentation and statistics gathered from the honeypot system
- Overall post-compromise activity
  - Post-compromise human activity
    - Total number of commands
    - Distinct number of commands
  - Downloaded files
    - Total number of downloads
    - Distinct number of downloads

Kippo-Graph - Playlog

- Replay input by attackers captured by the honeypot system
  - Total logs
  - ID
  - Timestamp
  - Size
  - Input Commands
  - Action

Kippo-Graph - Network

IP activity gathered from the honeypot system

- Total identified IP addresses:
  - IP address
  - Geolocation
  - Sessions count
  - Success
  - Last seen
- Total connection attempts from:
  - Timestamp
  - IP
  - Session
  - Username
  - Password
  - Success

Kippo-Graph - GEOIP

Geolocation from the top 10 IP addresses probing the system

- ID
- IP Address
- Probes
- City
- Region
- Country Name
- Code
- Latitude
- Longitude
- Hostname
- IP Lookup

Kippo-Graph – Graph Gallery

- 26 Graphs are generated in the Gallery.
- It helps you to easily interpret the data, using charts and graphs.
- Great for creating Reports and Presentations.

Kippo-Graph – Live Demo

- Live Demo
- Demo Lab Instance of Kippo-Graph (http://192.168.183.165)
- Demo DO Instance of Kippo-Graph (http://159.65.67.104/)

Kippo-Graph - GEOIP

- Internet Storm Center (dshield)- https://secure.dshield.org
- IPVOID - http://www.ipvoid.com
- Robtex.com - https://robtex.com
- Fortinet - FortiGuard Labs - http://www.fortiguard.com
- Alien Vault -https://www.alienvault.com

Kippo-Graph - GEOIP

- WatchGuard - Reputation Authority - http://www.reputationauthority.org
- McAfee (Threat Intelligence - https://www.mcafee.com
- IP-ADDRESS.com - http://ip-address.com
- Virus Total - https://virustotal.com

Parsing the Data

- Mini Summary of the data from Cowrie Logs, using custom Shell Scripts.
- To pass along IOCs to other Security Teams, for scanning and blocking.
  - Vulnerability Management Teams (Usernames and Passwords).
  - Firewall and Web Proxy Teams (IP Addresses and Domain Names).

Parsing Scripts – Failed Attempts

./parse_cowrie_failed_passwords.sh cowrie.log

- admin
- admin1
- admin123
- admin1234
- admintrup
- aerohive
- changeme
- motorola
- password
- pfsense
- raspberry
- raspberrypi

Parsing Scripts – Succeeded Attempts

./parse_cowrie_succeeded_passwords.sh cowrie.log

- 7ujMkooadmin
- 1q2w3e4r
- 1qazxsw2
- !QAZ@WSX
- 1q2w3e
- qazwsx
- !qAZxsw2
- 1qaz2wsx3edc

Parsing Scripts – Cowrie Log Report

./run_cowrie_log_report.sh cowrie.log

- Failed Usernames with frequency:
- Failed Username and Password Combinations:
- Succeeded Usernames with frequency:
- Passwords used in successful logins:

Cowrie – Playlog

Playlog – File Location

```
/home/cowrie/cowrie/bin$
```

Cowrie tty Log – File Location

```
./playlog /home/cowrie/cowrie/log/tty/
```

Using the Playlog Command

```
/home/cowrie/cowrie/bin$ ./playlog /home/cowrie/cowrie/log/tty/20180217-102849-4f145cfc77b2-0i.log
```

Files that you need to know

cowrie.cfg - Cowrie's configuration file. Default values can be found in cowrie.cfg.dist
data/fs.pickle - Fake File Sytem
data/userdb.txt - Credentials allowed or disallowed to access the honeypot
dl/ - files transferred from the attacker to the honeypot are stored here
honeyfs/ - file contents for the fake filesystem
log/cowrie.json - transaction output in JSON format
log/cowrie.log - log/debug output
log/tty/*.log - session logs
txtcmds/ - file contents for the fake commands
bin/createfs - used to create the fake filesystem
bin/playlog - utility to replay session logs

Cowrie Honeypot Placement

- This is my personal preference:
  - External (EXT).
  - Internal (INT).
  - De-militarized zone (DMZ).

Cowrie Honeypot Placement (cont.)

- Planning – Map out where on your network you want to place your Honeypots.
- Everywhere you have systems that can be attacked.
- Placed throughout the network.
  - Insider Threats.
  - Outsider Threats.
- Placed on each Subnet that you have identified through your planning.

Honeytokens

- Canarytokens.
  - https://canarytokens.org/generate
- Microsoft Word Document (Token Type).
  - Get alerted when a document is opened in Microsoft Word.
- Provide an email address.
- Reminder note when this token is triggered.

Additional Things to Do

- Setup Syslog.
- Setup Splunk.
- Setup Elasticsearch (ELK).
- JSON Logging (Easy processing in Log Management Solutions).
- Send login attempt information to SANS Dshield.
- Alerting via Email.
- HPfeeds (Honeynet Project generic authenticated datafeed protocol).

Honeypot Books

- Books that I recommend on the subject of Honeypots.
  - **Honeypots: Tracking Hackers**
    - by Lance Spitzner
  - **Honeypots: A New Paradigm to Information Security**
    - by R. C. Joshi and Anjali Sardana
  - **Honeypots for Windows (The Experts Voice)**
    - by Roger A. Grimes
  - **Virtual Honeypots: From Botnet Tracking to Intrusion Detection**
    - by Niels Provos and Thorsten Holz

Contributors

- Many people have contributed to Cowrie over the years. Special thanks to:
- Upi Tamminen (desaster) for all his work developing Kippo on which Cowrie was based.
- Dave Germiquet (davegermiquet) for TFTP support, unit tests, new process handling.
- Olivier Bilodeau (obilodeau) for Telnet support.
- Ivan Korolev (**fe7ch**) for many improvements over the years.
- And many, many others.

Presentation Resource Links

- Google Drive (SnowFroc-2018)
- Github
- VMware Workstation
- Oracle VM Virtualbox
- Raspberrypi.org

References

- Spitzner, L. (2003). Honeypots: tracking hackers. Boston: Addison-Wesley.
  - Honeypots: Tracking Hackers
- Github (www.github.com)
- Michel Oosterhof's (http://www.micheloosterhof.com/cowrie/)
- FakeNameGenerator (https://www.fakenamegenerator.com/order.php)

Thank You!

- For attending my Presentation today.
- Happy Honeypotting!

Questions

My Contact Information

- Troy Mitchell – Senior Cyber Security Engineer (Jacobs Engineering)
  - troy@troymitchell.net
  - www.troymitchell.net
- LinkedIn
  - https://linkedin.com/in/troy-Mitchell-b318729