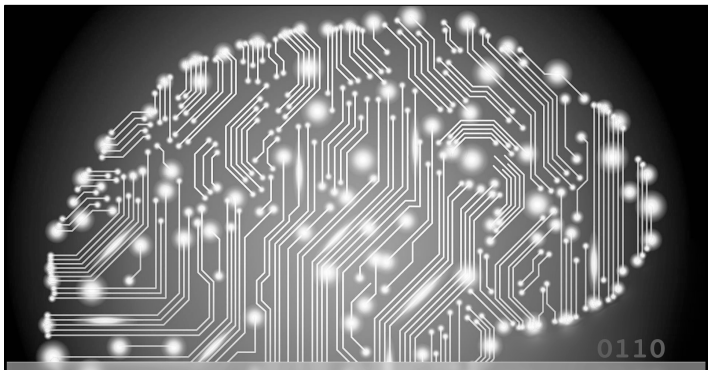


Honeypots: Sweet Spot in Network Security

Kevin Capwell – META
Pat Zielke – Viroqua

Honeypot Security

Disclaimer

Any actions and or activities related to the material contained within this session are solely your responsibility. The misuse of the information in this presentation can result in criminal charges brought against the person(s) in question.

Furthermore, this presentation contains materials that can be potentially damaging or dangerous. If you do not fully understand something in this presentation, then do not attempt to use them! These materials are for educational and research purposes only. The following presentation and its content should not be viewed – by anyone...

Honeypot Security

School District of Onalaska

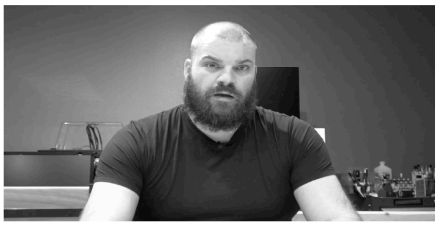
- Kevin Capwell
  fmr → Data Systems Director (24 years)
- Enrollment: 3,166
- Total Staff: 415
- Buildings:
  High School, Middle School, three Elementary Schools, District Office, Pupil Service and School Nutrition (~12 sq. mi.)
- Computers: Desktop 1400, Chrome-books 1400, Other mobile 200.

Honeypot Security

Viroqua Area Schools

- Pat Zielke
  Technology Coordinator – 19 years
- Enrollment: 1,191
- Total Staff: 184
- Buildings:
  Shared High School/Middle School a separate Elementary all on the same campus.
- Computers: Desktop 400, Chrome-books 200, Other mobile 90.

Honeypot Security



Pat Zielke

Technology Coordinator Viroqua.

Honeypot Security

Kev and Pat have a security chat…

- Good security comes in layers
- Passwords – good, bad and ugly!
- Patch / vulnerability scan (Nessus)
- Penetration testing (Kali)
- Monitor critical points (SNMP v3)
- Centralized tools with integrated management (Netsight, OneView)
- Monitor top user statistics
- Check / test backups / offsite
- Logs, Logs, Logs! (Scalyr, Nagios LS)

Honeypot Security

Pat Zielke

Technology Coordinator
Viroqua.

Honeypot Security

Honeypot

is a physical or virtualized server designed to attract attacks upon itself. This tool flaunts its intended vulnerabilities to tempt the unwary into tripping your network alarm. The moment anyone connects with this server it will report the attempt and document the date and time.

Honeypot Security

What is a script kiddie?

- Unskilled hacker who resorts to other programmer's scripts or applications to attack computer systems, networks and servers.
- A script kiddie could be any age.
- This type of hacker can be just as disruptive as a skilled hacker.
- Their objective is to attempt to impress their peers, or to gain credit in computer hacking circles.

Honeypot Security

What are the common honeypots?

- Production - are placed inside the network to improve security.
- Research - used to assess the current threat level. Primarily used by research, military, or government.
- High-interaction - mimics high value servers with a multitude of services.
- Medium-interaction - mimics a server in a very controlled environment.
- Low-interaction - simulate the services frequently exploited by attackers.

Honeypot Security

What is the intent of a honeypot?

- Early warning honeypots are set up to simulate one or more fake systems that would immediately indicate malicious intent if even slightly probed.
- Early warning honeypots excel at catching hackers and malware.
- Research honeypots can capture and quarantine malware and new hacker exploits that are encountered.

Honeypot Security

Where should I place the honeypot?

- Physically near the systems they are attempting to protect.
- They can be placed in the same datacenter or IP address space where your production servers reside.
- Add one to your DMZ as an early warning device.
- If you have multiple buildings, place your honeypots at each building where high value targets are located.

Honeypot Security

Let's show some examples!

- CentOS is a community-developed and supported alternative to RHEL. It is similar to Red Hat Enterprise Linux but lacks the enterprise-level support. CentOS is more or less a free replacement for RHEL.
- CentOS 7 System requirements: Updates through June 30th, 2024 1GB/logical CPU, 10GB/20GB (storage)
- Firewall has been disabled.

Honeypot Security

Our first example: PenTBox

- PenTBox: Open Source – cost $0
- Requires: CentOS, Ruby scripting lang
- Best as web and telnet honeypot
- $ sudo yum install ruby
- $ wget http://downloads.sourceforge.net/project/pentbox18realised/pentbox-1.8.tar.gz
- $ tar -zxvf pentbox-1.8.tar.gz
- $ cd pentbox-1.8
- $ sudo ./pentbox.rb

Honeypot Security

PenTBox: Main Menu

Honeypot Security

PenTBox: Network Tools → Honeypot

```
1- Net DoS Tester
2- TCP port scanner
3- Honeypot
4- Fuzzer
5- DNS and host gathering
6- MAC address geolocation (samy.pl)

0- Back

[   -> 3                                                                    ]

// Honeypot //

You must run PenTBox with root privileges.

 Select option.

1- Fast Auto Configuration
2- Manual Configuration [Advanced Users, more options]

[   -> 1                                                                    ]

  HONEYPOT ACTIVATED ON PORT 80 (2019-02-17 19:33:55 -0600)
```

Honeypot Security

PenTBox: the Results

Access denied

HTTP Referrer login failed

IP Address login failed

2019-02-17 19:33:55 -0600

```
 INTRUSION ATTEMPT DETECTED! from [redacted]:64546 (2019-02-17 19:34:25 -0600)
 ------------------------------
GET /favicon.ico HTTP/1.1
Host: [redacted]
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.14; rv:60.0) Gecko/20100101
Firefox/60.0
Accept: */*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: keep-alive
```

Honeypot Security

PenTBox: Wait! There's more…

IP Range: [redacted] to [redacted] IP Range

Hostname: [redacted] IP↑ Netmask Start

IP	Ping	Hostname	Ports [0+]
[redacted]	0 ms	[n/a]	[n/s]
[redacted]	[n/a]	[n/s]	[n/s]
[redacted]	[n/a]	[n/s]	[n/s]
[redacted]	6 ms	[n/a]	[n/s]

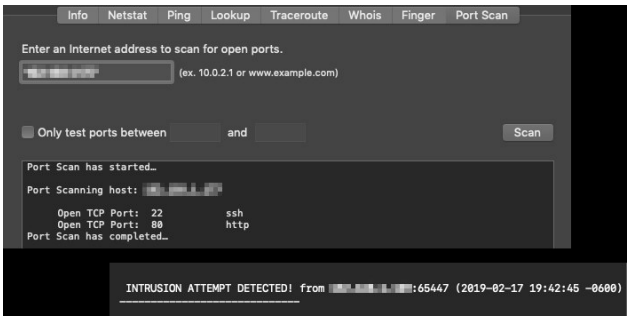
```
 INTRUSION ATTEMPT DETECTED! from [redacted]:64987 (2019-02-17 19:37:41 -0600)
 ------------------------------

 INTRUSION ATTEMPT DETECTED! from [redacted]:64988 (2019-02-17 19:37:42 -0600)
 ------------------------------

 INTRUSION ATTEMPT DETECTED! from [redacted]:64989 (2019-02-17 19:37:43 -0600)
 ------------------------------
```

Honeypot Security

… and more…

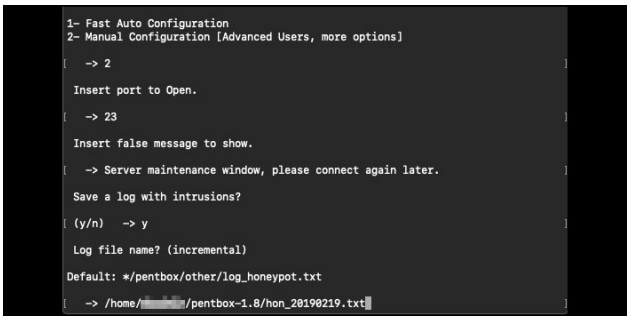


Honeypot Security

What's behind the curtain…

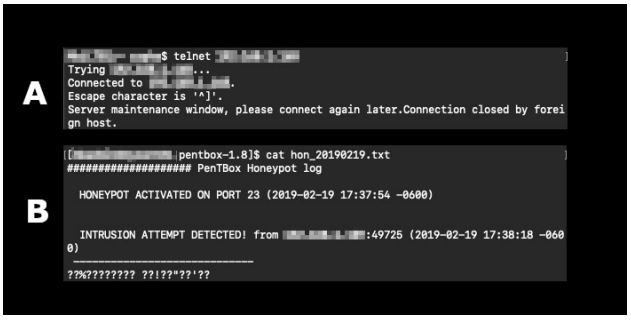
Honeypot Security

Can we log all of the events?

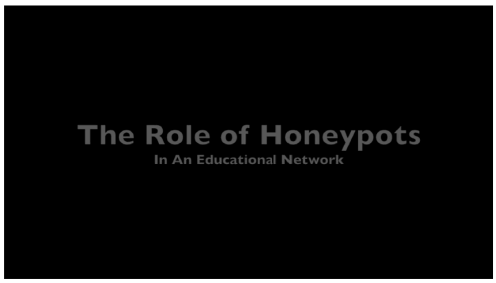


Honeypot Security

PenTBox: the Results!



Honeypot Security



Pat Zielke

Technology Coordinator Viroqua.

Honeypot Security

Our second example: Cowrie

- Cowrie: Open Source – cost $0
- Requires: CentOS, GCC, Python, git, pip, python-virtualenv, pycrypto
- Best as ssh and telnet honeypot
- Virtual filesystem displays Debian 5.0
- Filesystem allows add/remove files
- False file data to misdirect hackers
- Session logs are stored with timing
- Virtual accounts and passwords protect the honeypot's true OS and files.

Honeypot Security

Cowrie example: ssh, root and logs.

```
[redacted]$ ssh -l root [redacted]
Password:
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@server13:~# ls -al
drwx------ 1 root root 4096 2013-04-05 07:25 .
drwx------ 1 root root 4096 2013-04-05 07:25 ..
drwx------ 1 root root 4096 2013-04-05 06:58 .aptitude
-rw-r--r-- 1 root root  570 2013-04-05 06:52 .bashrc
-rw-r--r-- 1 root root  140 2013-04-05 06:52 .profile
drwx------ 1 root root 4096 2013-04-05 07:05 .ssh
root@server13:~# Connection to [redacted] closed.
                2019-02-20T01:44:35.701352-0600 [SSHChannel session (0) on SSHService 'ssh-conne
                ction' on HoneyPotSSHTransport,0,[redacted]] getting shell
                2019-02-20T01:44:39.840084-0600 [SSHChannel session (0) on SSHService 'ssh-conne
                ction' on HoneyPotSSHTransport,0,[redacted]] CMD: ls -al
                2019-02-20T01:44:39.842008-0600 [SSHChannel session (0) on SSHService 'ssh-conne
                ction' on HoneyPotSSHTransport,0,[redacted]] Command found: ls -al
                2019-02-20T01:44:45.568345-0600 [SSHChannel session (0) on SSHService 'ssh-conne
                ction' on HoneyPotSSHTransport,0,[redacted]] Received CTRL-D, exiting..
```

Honeypot Security

Cowrie example: telnet, root and logs.

```
[redacted]$ telnet [redacted]
Trying [redacted]...
Connected to [redacted].
Escape character is '^]'.
login: root
Password:
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@server13:~# ls -al

                2019-02-20T01:44:58.312965-0600 [CowrieTelnetTransport,0,[redacted]] login at
                tempt [root/nowthis] succeeded
                2019-02-20T01:44:58.315088-0600 [CowrieTelnetTransport,0,[redacted]] Initiali
                zed emulated server as architecture: linux-x64-lsb
                2019-02-20T01:45:07.098278-0600 [CowrieTelnetTransport,0,[redacted]] CMD: ls
                -al
                2019-02-20T01:45:07.100194-0600 [CowrieTelnetTransport,0,[redacted]] Command
                found: ls -al
                2019-02-20T01:45:12.677137-0600 [CowrieTelnetTransport,0,[redacted]] Received
                 CTRL-D, exiting..
```

Honeypot Security

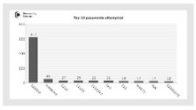
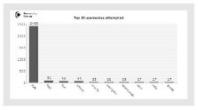
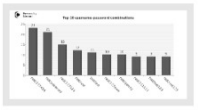
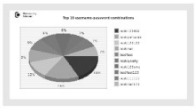
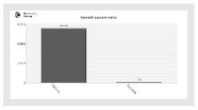
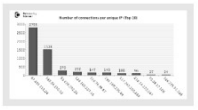
More info: Cowrie

- Cowrie is a fork of Kippo
- Kippo has not been updated
- Kippo is detectable by hackers
- SFTP and SCP support for file upload
- Support for SSH exec commands
- Logging of SSH proxying
- Logging in JSON for easy importing
- fs.pickle can be customized

Honeypot Security

Kippo-Graph is optional.

Provided you have visited all the other pages/components (for the graphs to be generated) you can see all the images in this single page with the help of fancybox

Honeypot Security

```
tiziano@debianEEE:~/cowrie$ ./utils/playlog.py
CHANGELOG.md    cowrie.cfg      cowrie.pid      dl/             .git/           .gitignor
cowrie/         cowrie.cfg.dist  data/           doc/            .gitattributes  honeyfs/
tiziano@debianEEE:~/cowrie$ ./utils/playlog.py log/tty/
20150727-193523-9a1630fc.log  20150728-004904-fa0d5fe.log  20150728-005220-8040d805.log  2015
tiziano@debianEEE:~/cowrie$ ./utils/playlog.py log/tty/20150728-0052
20150728-005220-8040d805.log  20150728-005253-c5067c99.log
tiziano@debianEEE:~/cowrie$ ./utils/playlog.py log/tty/20150728-005220-8040d805.log
```

Cowrie

script kiddie
is foiled by
a honeypot.

Honeypot Security

Our third example: HoneyDrive 3

- Constant CLI interaction is a bummer
- Please let me use a mouse!
- Ten pre-installed honeypot packages
- Dionaea malware honeypot + scripts
- Distributed as a single OVA file
- Import the appliance on your VM manager
- Includes security, forensics and anti-malware tools
- All of the notes are on the desktop

Honeypot Security

Let's see the graphical goodness!

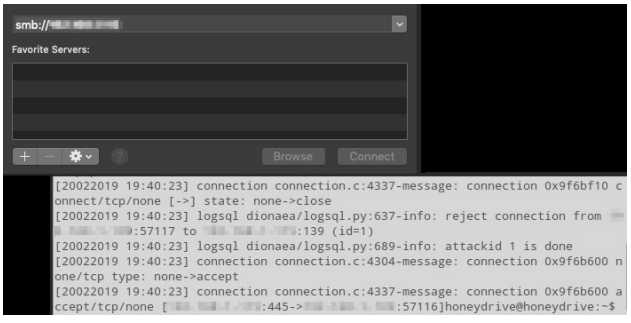
Honeypot Security

Dionaea in action.

```
honeydrive@honeydrive:~$ sudo netstat -antp | grep dionaea
tcp        0      0 0.0.0.0:1433            0.0.0.0:*               LISTEN
2673/dionaea
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN
2673/dionaea
tcp        0      0 0.0.0.0:445             0.0.0.0:*               LISTEN
2673/dionaea
tcp        0      0 0.0.0.0:5060            0.0.0.0:*               LISTEN
2673/dionaea
tcp        0      0 0.0.0.0:5061            0.0.0.0:*               LISTEN
2673/dionaea
tcp        0      0 0.0.0.0:135             0.0.0.0:*               LISTEN
2673/dionaea
tcp        0      0 0.0.0.0:42              0.0.0.0:*               LISTEN
2673/dionaea
tcp        0      0 0.0.0.0:21              0.0.0.0:*               LISTEN
2673/dionaea
```

Honeypot Security

Logging a SMB (445) connection.



Honeypot Security

Logging a HTTPS (443) connection.

Directory listing for /

- ./

```
[20022019 19:44:05] connection connection.c:4337-message: connection 0x9f71300 a
ccept/tls/none [[redacted]:443->[redacted]:57138] state: none->handshake
[20022019 19:44:05] connection connection.c:168-warning: getpeername failed (Tra
nsport endpoint is not connected)
[20022019 19:44:05] connection connection.c:4337-message: connection 0x9f71300 a
ccept/tls/handshake [[redacted]:443->[redacted]:57138] state: handshake->e
stablished
[20022019 19:44:05] connection connection.c:4337-message: connection 0x9f71300 a
ccept/tls/established [[redacted]:443->[redacted]:57138] state: establishe
d->close
```

Honeypot Security

Logging a FTP (21) connection.

```
[redacted]$ ftp [redacted]
Connected to [redacted].
220 Welcome to the ftp service
Name ([redacted]): anonymous
331 Guest login ok, type your email address as password.
Password:
230 Anonymous login ok, access restrictions apply.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
227 Entering Passive Mode ([redacted]).
150 File status okay; about to open data connection.
226 Tra[20022019 19:57:42] connection connection.c:4337-message: connection 0x9f71300 a
ftp> ex[redacted]ccept/tcp/established [[redacted]:21->[redacted]:57261] state: established
221 Goo->close
[20022019 19:57:42] logsql dionaea/logsql.py:689-info: attackid 4 is done
[20022019 19:58:42] connection connection.c:4304-message: connection 0x9fcd570 n
one/tcp type: none->accept
[20022019 19:58:42] connection connection.c:4337-message: connection 0x9fcd570 a
ccept/tcp/none [[redacted]:21->[redacted]:57298] state: none->established
[20022019 19:58:42] connection connection.c:4304-message: connection 0x9f71300 n
one/tcp type: none->connect
```

Honeypot Security

Logging a MySQL (1433) connection.

```
[redacted]$ nmap -p 1433 [redacted]
Starting Nmap 7.70 ( https://nmap.org ) at 2019-02-20 15:53 CST
Nmap scan report for [redacted]
Host is up (0.00075s latency).

PORT     STATE SERVICE
1433/tcp open  ms-sql-s

Nmap done: 1 IP address (1 host up) scanned in 0.14 seconds
```

```
[20022019 20:13:31] connection connection.c:4337-message: connection 0x9fcd570 a
ccept/tcp/none [[redacted]:1433->[redacted]:57397] state: none->establishe
d
[20022019 20:13:31] connection connection.c:4304-message: connection 0x9f71300 n
one/tcp type: none->connect
[20022019 20:13:31] connection connection.c:850-warning: Could not connect un://
/tmp/p0f.sock:0 (Connection refused)
[20022019 20:13:31] connection connection.c:4337-message: connection 0x9f71300 c
onnect/tcp/none [->] state: none->close
[20022019 20:13:31] logsql dionaea/logsql.py:624-info: accepted connection from
[redacted]:57397 to [redacted]:1433 (id=7)
```

Honeypot Security

Common honeypot strategies

- Study hackers and capture samples of potential malware.
- Provide a tempting weak server as an alarm bell for IT staff.
- Log all attacks and easy to reset.
- Leverage data to enhance other security technologies.
- Forward ports on routers to honeytraps to allow for easy access.
- Setting geoblocking (Syria, Iran, Sudan, Cuba and Russia).

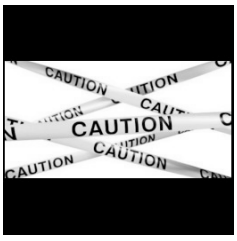
Honeypot Security

More security strategies…

- Frustrate hackers and encourage them to move on to easier targets.
- All honeypot information should be sent to a centralized log server.
- Setup alerts for honeypot alarms. This will allow for decisive action.
- Good list: https://github.com/paralax/awesome-honeypots (open-source).
- Allows IT department to become proactive on cyber security.

Honeypot Security

Honeypot planning cycle



- At least one person must install, configure, update, and monitor the honeypot.
- A neglected honeypot can become an attack platform into your network.
- Determining the prioritization of what to monitor and which alerts to send is the most time consuming aspect.

Honeypot Security

Pat Zielke

Technology Coordinator
Viroqua.

Honeypot Security

Questions?

Pat Zielke
Viroqua School District
pzielke@viroqua.k12.wi.us

Kevin Capwell
Midwest Educational Technology Association
kcapwell@brainstormk20.com