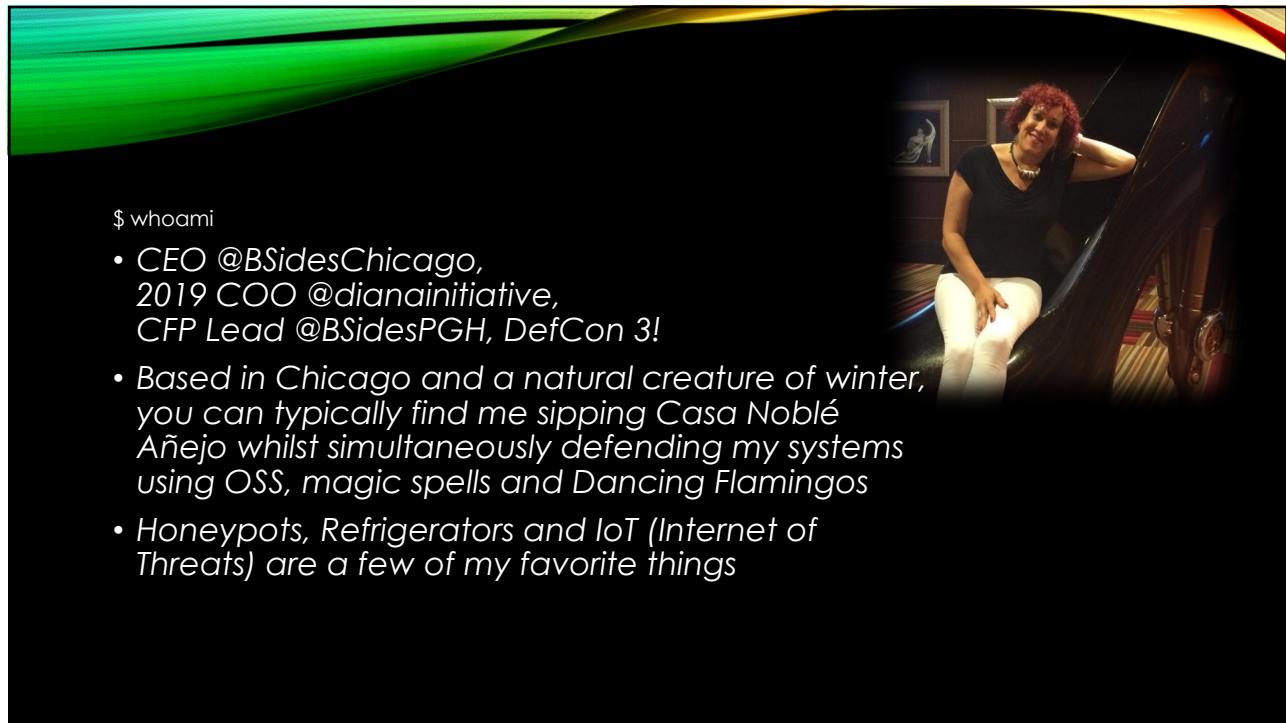




1



2

DISCLAIMER

- The views and opinions expressed in this presentation are my own and do not necessarily reflect the official policy or position of any current or previous employer. Examples of exploitations, coding and vulnerabilities discussed within this presentation are only examples and they should not be utilized in the real-world.

3

DISCLAIMER (PART 2)

- I'm obsessed(?) with home security equipment, honeypots and colos
- If you want to have a life, perhaps tone it down a bit
- YMMV

4

WHY WE ARE NOT HERE

- This is not a demo of 5000 different honeypots
- I'm not showing you all my honeypots (duh)
- Honeypots are only PART of your Security Posture



5

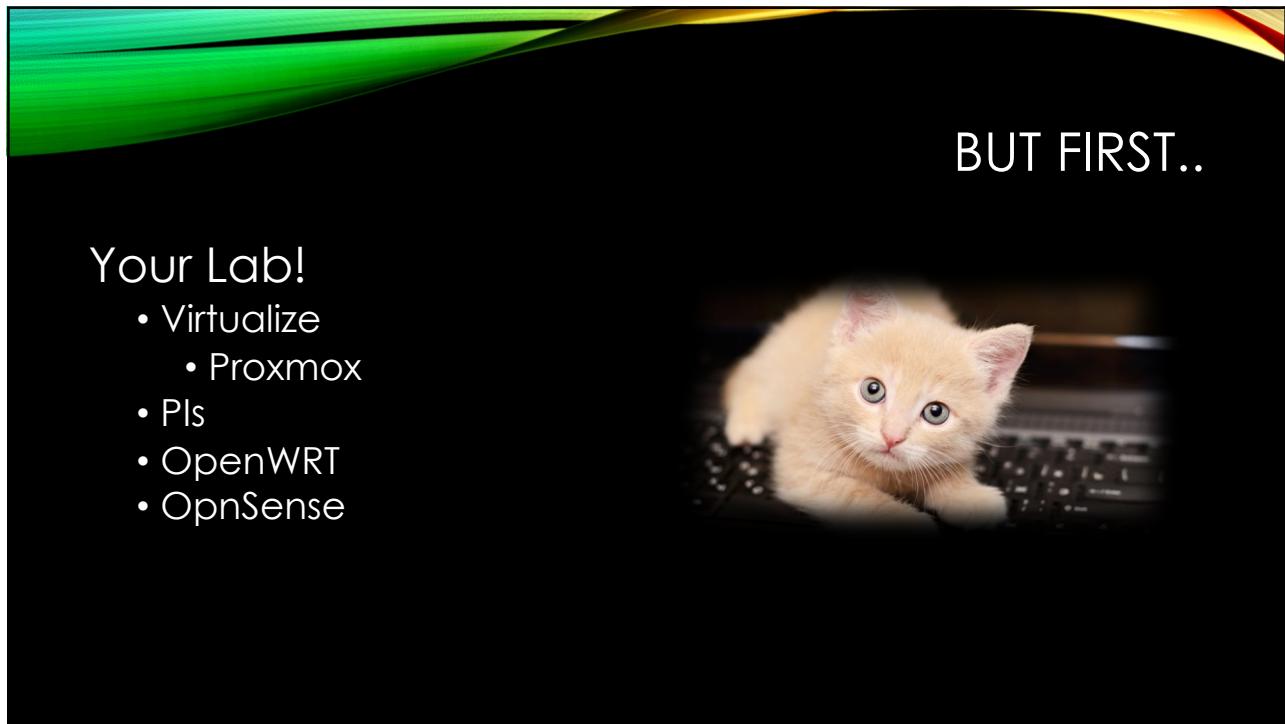
WHY WE ARE HERE

- Spending exceeded \$161 billion - 2022
- Attacks and breaches are commonplace
- Security “stuff” is vulnerable
- Lateral Movement – (*this will become more important*)
- But what about –
 - Your Security Architecture is not unique
 - What is your “typical day”



Instead of Brilliance, we have standardized mediocrity.
– John Strand, Offensive Countermeasures

6



7

VIRTUALIZE!

PROXMOX Virtual Environment 6.2-4

Datacenter

Health

Status	Nodes
Standalone node - no cluster defined	✓ Online 1 ✘ Offline 0

Guests

Virtual Machines	LXC Container
Running 1	Running 0
Stopped 0	Stopped 0

Resources

CPU	Memory	Storage
16% of 4 CPU(s)	37% 11.69 GB of 31.26 GB	15% 136.87 GB of 888.78 GB

8



9



10

MONITORING WAZUH

The dashboard displays various metrics and charts related to Wazuh monitoring:

- Alert level evolution:** A line chart showing the count of alerts over time (timestamp per 3 hours) from November 13 to November 17, 2020. The y-axis ranges from 0 to 6,000. The chart shows several peaks, with the highest peak reaching approximately 5,500.
- Top MITRE ATTACKS:** A donut chart showing the distribution of top MITRE attacks. The categories and counts are: Valid Accounts (blue), Brute Force (green), Stored Data Manipulation (purple), Exploit Public-Facing Port (red), Data Destruction (yellow-green), File Deletion (dark purple), File and Directory Disclosure (pink), Process Injection (light green), Remote Services (light pink), Disabling Security Token (light yellow), and Sudo (dark blue).
- Top 5 agents:** A donut chart showing the distribution of top 5 agents. The agents and their counts are: maul (orange), gobo (blue), keywest (red), kermit (yellow-green), and beaker (dark red).
- Alerts evolution - Top 5 agents:** A line chart showing the count of alerts for the top 5 agents over time (timestamp per 3 hours) from November 13 to November 18, 2020. The y-axis ranges from 0 to 6,000. The chart shows multiple peaks, with the highest peak reaching approximately 5,500.

```
curl -s0 https://packages.wazuh.com/4.5/wazuh-install.sh && sudo bash ./wazuh-install.sh -a
```

11

HONEYPOTS

- Honeypots vs Deception
 - A resource with no value(?)
 - Value = Use of Resource
 - Does Not Hack Back
- Important Points
 - Deployment
 - Customization = Planning! (more on this)
 - 100's of "types"

12

PICK ONE

- OpenCanary -- <https://opencanary.readthedocs.io/en/latest/>
- Adhd – <https://www.activecountermeasures.com/free-tools/adhd/>
- Honey Badger -- <https://github.com/adhdproject/honeybadger> (GEO!)
- CHN-- <https://communityhoneynetwork.readthedocs.io/en/stable/>
- Canarytokens -- <https://canarytokens.org/generate>
- T-pot -- <https://github.com/telekom-security/tpotce>
- Cowrie -- <https://github.com/cowrie/cowrie>
- Conpot -- <https://github.com/mushorg/conpot>
- PIs w/lights -- <https://github.com/mattymcfatty/HoneyPi>
- Lots more -- <https://github.com/paralax/awesome-honeypots>
- What about the “Real Thing”? Hmmmm..

13

ADHD Version: 4.0.0 | [GitHub Page](#) | [Project Page](#)

Black Hills Information Security

ADHD

- Credentials
- Artillery
 - Example 1: Running Artillery
 - Example 2: Triggering a Honeyport
 - Example 3: Adding a File to a Watched Directory
- Bear Trap
 - Example 1: Basic Usage
- BeEF
 - Example 1: Hooking a Web Browser
 - Example 2: Browser Based Exploitation With BeEF
- CanaryTokens
 - Example 1: Creating Callbacks Using Local Canary Instance
 - Example 2: Creating Callbacks Using CanaryTokens.org
- Cowrie
 - Example 1: Running Cowrie
 - Example 2: Cowrie In Action
 - Example 3: Viewing Cowrie's Logs

<https://www.activecountermeasures.com/free-tools/adhd/>

14

PORTSPOOF (1)

nmap -p200-300 gonzo

PORT STATE SERVICE

200/tcp open src

201/tcp open at-rtmp

202/tcp open at-nbp

203/tcp open at-3

204/tcp open at-echo

205/tcp open at-5

206/tcp open at-zis

207/tcp open at-7

208/tcp open at-8

209/tcp open tam

210/tcp open z39.50

211/tcp open 914c-g

212/tcp open anet

213/tcp open ipx

214/tcp open vmpwscs

215/tcp open softpc

216/tcp open atls

217/tcp open dbase

218/tcp open mpp

219/tcp open uarps

220/tcp open imap3

221/tcp open fln-spx

222/tcp open rsh-spx

223/tcp open cdc

224/tcp open masqdialer

15

PORTSPOOF (2)

nmap -A gonzo

Starting Nmap 7.80 (https://nmap.org) at 2020-02-27 09:52 EST

Stats: 0:04:43 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan

SYN Stealth Scan Timing: About 75.58% done; ETC: 09:58 (0:01:31 remaining)

Stats: 0:04:53 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan

SYN Stealth Scan Timing: About 77.22% done; ETC: 09:58 (0:01:26 remaining)

Stats: 0:07:39 elapsed; 0 hosts completed (1 up), 1 undergoing SYN Stealth Scan

16

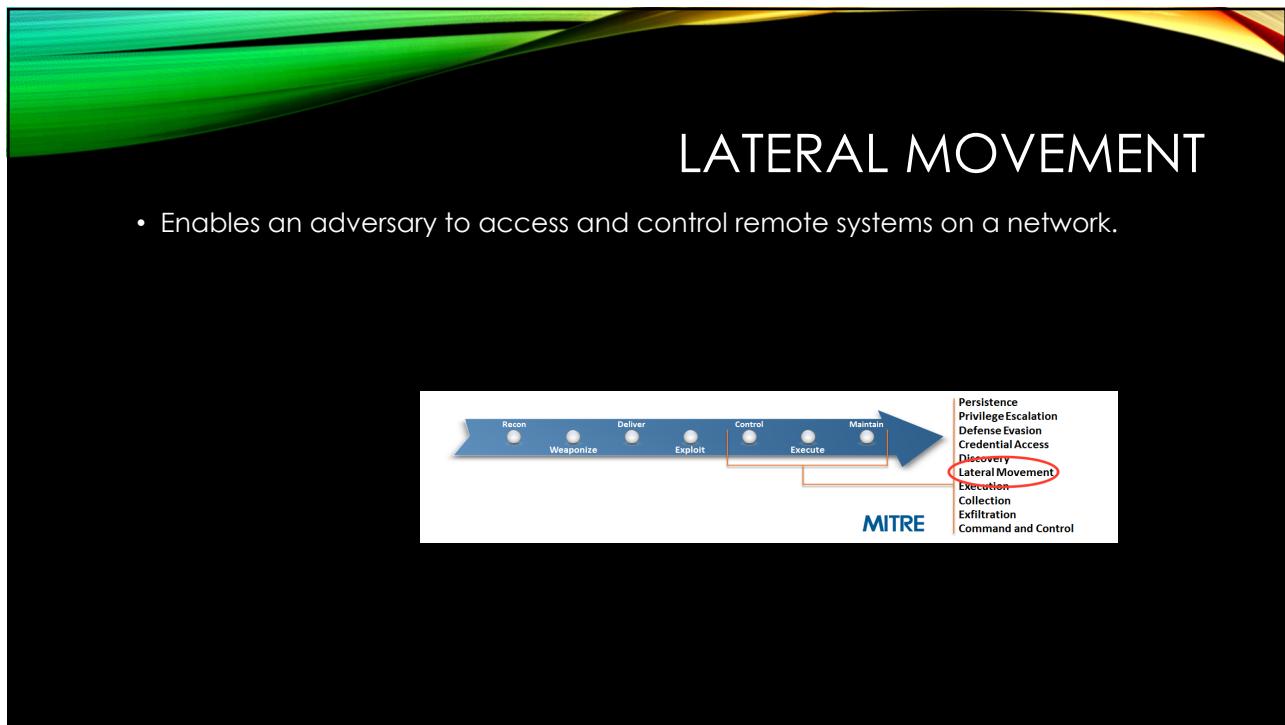
DSHIELD
5 MINUTES!

Date	Time	Source	Source Port	Target	Target Port	Protocol
<hr/>						
2020-11-19	00:16:59	110.154.181.126	14649	192.168.100.100	23	6
2020-11-19	00:17:26	74.120.14.37	54522	192.168.100.100	23	6
2020-11-19	00:17:31	167.248.133.92	64123	192.168.100.100	23	6
2020-11-19	00:17:31	167.248.133.40	39580	192.168.100.100	23	6
2020-11-19	00:17:31	45.129.33.5	56294	192.168.100.100	23	6
2020-11-19	00:18:03	192.241.217.154	44838	192.168.100.100	23	6
2020-11-19	00:18:31	64.64.104.10	15188	192.168.100.100	23	6
2020-11-19	00:18:43	185.175.93.14	49700	192.168.100.100	23	6
2020-11-19	00:19:14	83.97.20.35	55995	192.168.100.100	23	6

Showing 1 to 100 of 11,920 entries

Previous 1 2 3 4 5 ... 120 Next

17



18

OODA VS CCAD

- OODA
 - Observe
 - Orient
 - Decide
 - Act
- CCAD
 - Confuse
 - Confound
 - Annoy
 - Delay

19

		Description
data.eventid	cowrie.login.s	
data.message	login attempt	raju tried to login to honeypot.
data.password	password	
data.sensor	sasha	nproc tried to login to honeypot.
data.session	d5c6623bdec	
data.src_ip		root logged into honeypot.
data.timestamp	2022-08-07T	nproc tried to login to honeypot.
data.username	root	pinkfloyd tried to login to honeypot.
decoder.name	json	root logged into honeypot.
full_log	{"eventid":"cc succeeded","	amministratore tried to login to honeypot.
id	1659883190.	redmin tried to login to honeypot.
input.type	log	Integrity checksum changed
location	/home/aladin,	nproc tried to login to honeypot.
manager.name	honeydew	zimbra tried to login to honeypot.
rule.description	root logged in	coinfo tried to login to honeypot.
		jmlarre tried to login to honeypot.

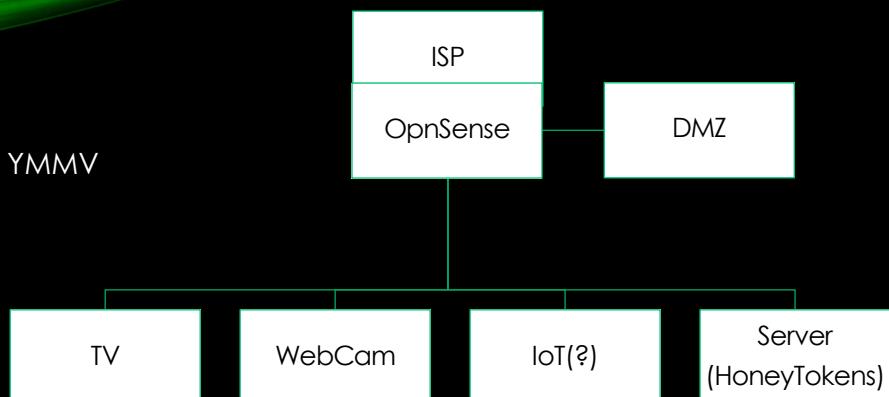
20

DEPLOYMENT

- Plan, Plan, Plan!
 - Low, Medium, ~~High~~
 - Honeypots, Honeyports, Honeytokens, Honeycreds
 - Customization ← Ding ding ding!
 - Real vs Self-Signed Certs
 - Actual Applications
 - HIDS / OSSEC / Wazuh / SIEM
 - Rules! Tuning
- Where?
 - Server Farms
 - Cloud Storage
 - IoT (Shodan is your friend!!!)
 - <https://shodan.io>
 - DMZ (Guest WiFi)
 - MX, DNS
 - PoS
 - WP, Rpi, VMs, VPS

21

- An Idea – YMMV



```
while true ; do nc -l -p 1500 -c 'echo -e "HTTP/1.1 200 OK\n\n $(date)"'; done
```

22

LET'S TALK CUSTOMIZATION

- Shodan
 - Banners
 - Versions!
 - HTML
- Certificates
- HoneyTokens
- Hosts
 - Filesystems
 - Commands
 - History
 - Processes
 - HoneyTokens
- Real Servers and Apps
 - Staging

Too legit to quit!!
Make it look real!
Rename built-in user richard to phil, it's used as detection mechanism.

23

FINDING THEM

<https://honeyscore.shodan.io/>



Honeypot Or Not?
Enter an IP to check whether it is a honeypot or a real control system:

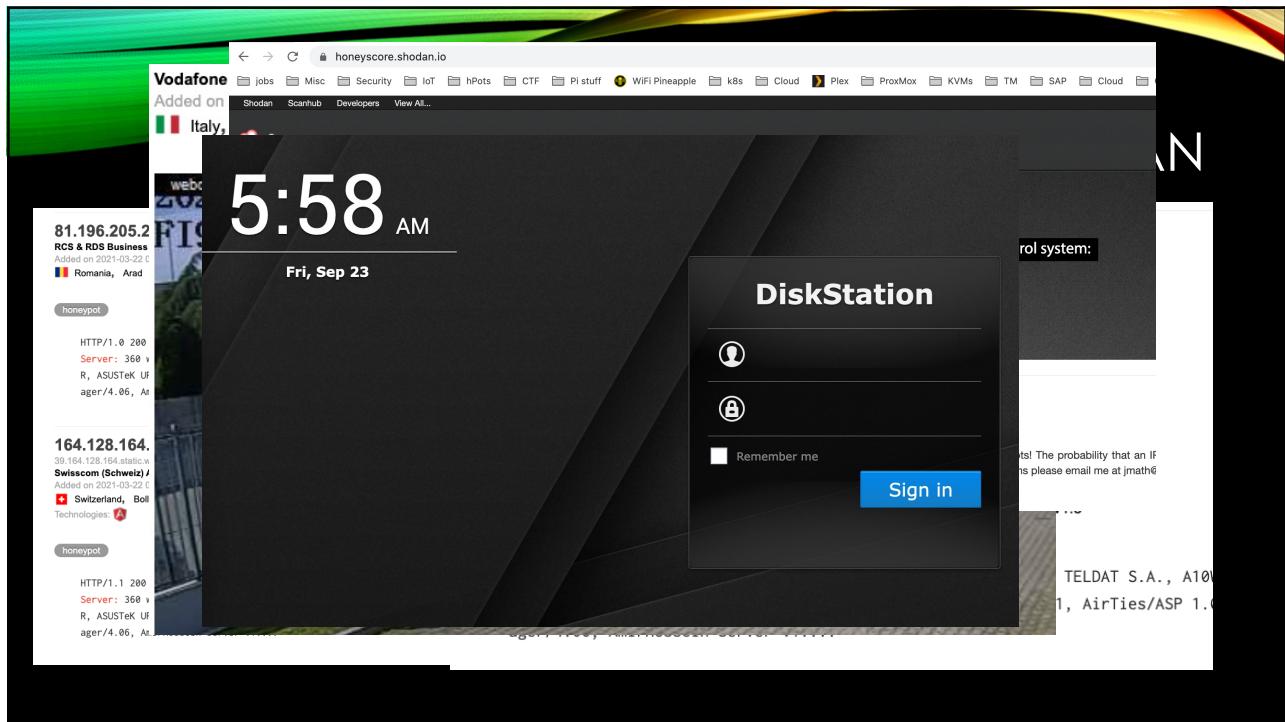
24. Check for Honeypot

Looks like a real system!

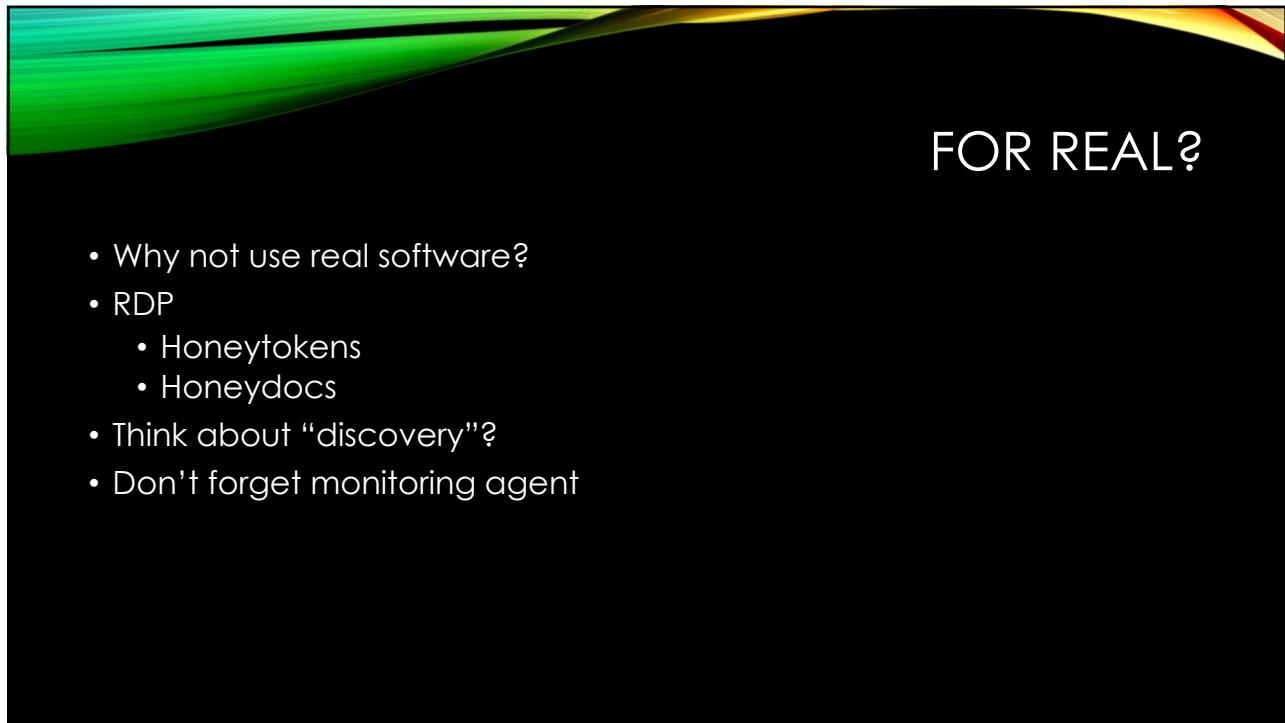
Frequently Asked Questions

1. How does it work?
The defining characteristics of known honeypots were extracted and used to create a tool to let you identify honeypots! The probability that an IP is a honeypot is captured in a "Honeyscore" value that can range from 0.0 to 1.0. This is still a prototype/ work-in-progress so if you find some problems please email me at jmath@shodan.io
2. What's the purpose?
Honeypots are a great tool for learning more about the Internet, the latest malware being used and keep track of infections. When trying to catch an intelligent attacker though, many honeypots fall short in creating a realistic environment. Honeyscore was created to raise awareness of the short-comings of honeypots.
3. What technology did you use?
The Honeyscore website and algorithm uses the following APIs/ frameworks:
 - Shodan Developer API
 - Python
 - Jade Node Template Engine

24



25



26

WHAT IS IT?

```
# uname -a
Linux RT-AC5300 2.6.36.4brcmarm #1 SMP PREEMPT Fri Oct 18 16:13:51 CST
2019 armv7l ASUSWRT

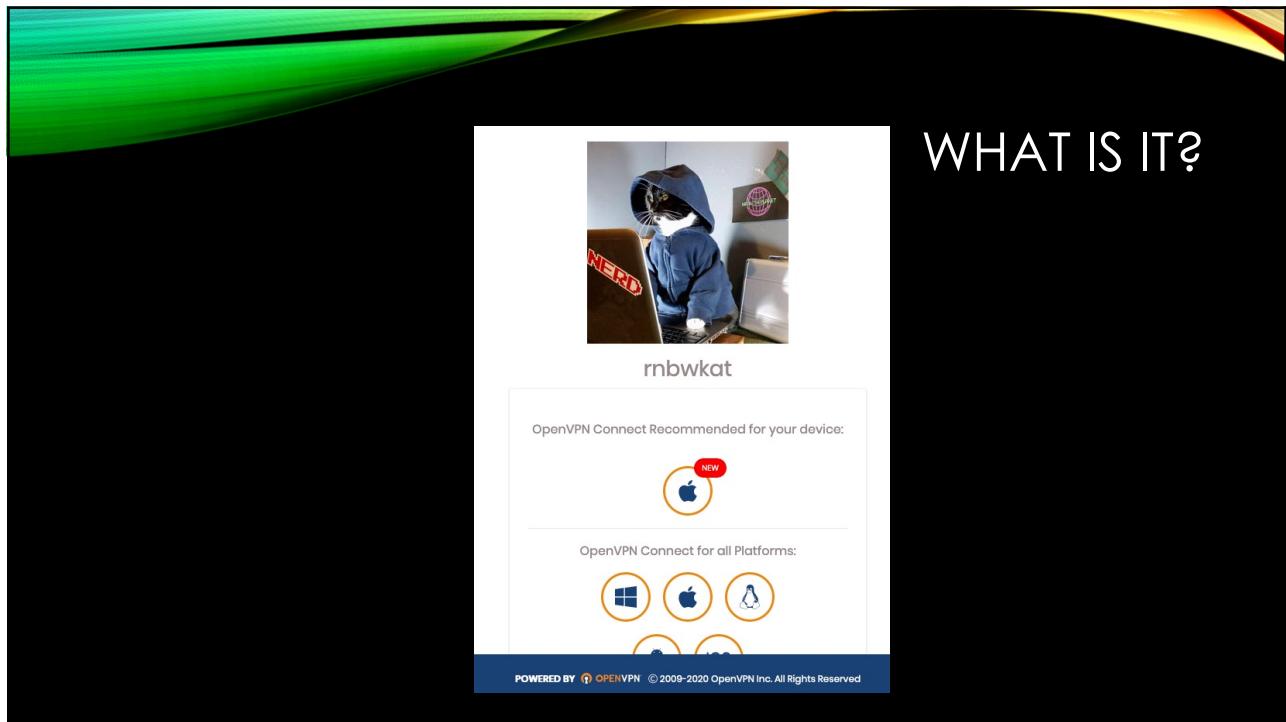
# df
Filesystem      1K-blocks    Used Available Use% Mounted on
rootfs          40960     40960        0 100% /
/dev/root       40960     40960        0 100% /
devtmpfs        257456        0 257456   0% /dev
tmpfs           257600      944 256656   0% /tmp
/dev/mtdblock4  65536     2064    63472   3% /jffs
```

27

COWRIE

```
# ls -al /etc
srwxrwxrwx  1 admin  root          0 May  5 2018 amas_lib_socket
-rw-rw-rw-  1 admin  root         1017 May  5 2018 cert.pem
drwxrwxrwx  2 admin  root         100 May  5 2018 cfg_mnt
srwxrwxrwx  1 admin  root          0 May  5 2018 cfgmnt_ipc_socket
-rw-rw-rw-  1 admin  root         380 May  5 2018 dnsmasq.conf
drwx-----  2 admin  root         100 Feb 27 13:24 dropbear
lrwxrwxrwx  1 admin  root         20 Dec 31 1969 e2fsck.conf -> /rom/etc/e2fsck.conf
drwxrwxrwx  2 admin  root         60 May  5 2018 email
lrwxrwxrwx  1 admin  root         19 Dec 31 1969 ethertypes -> /rom/etc/ethertypes
-rw-r--r--  1 admin  root          0 Dec 31 1969 fstab
-rw-r--r--  1 admin  root         52 May  5 2018 group
-rw-rw-rw-  1 admin  root          0 May  5 2018 group.custom
-rw-r--r--  1 admin  root         52 May  5 2018 gshadow
-rw-r--r--  1 admin  root        176 May  5 2018 hosts
```

28



29

- <https://www.canarytokens.org/generate>

CANARY TOKENS

What is this and why should I care?

Documentation

Select your token

Provide an email address or webhook URL (or both space separated)

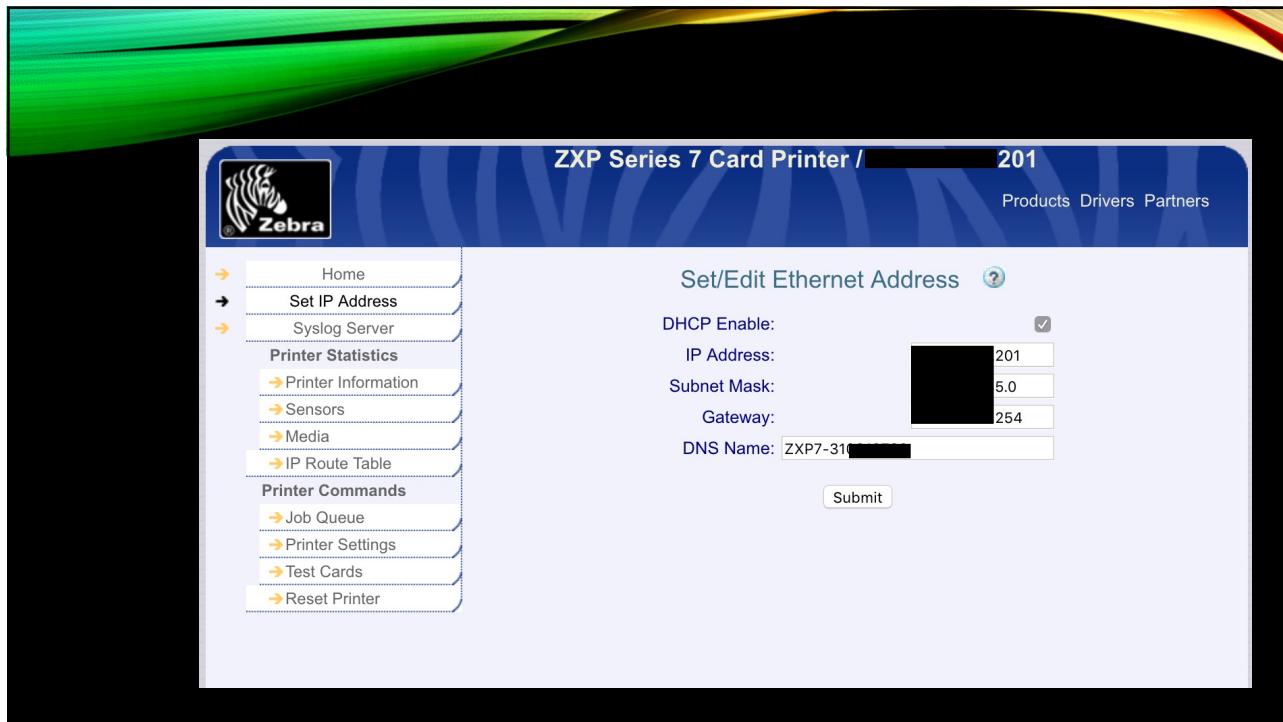
Reminder note when this token is triggered.

Fill in the fields above

Brought to you by Thinkst Canary, our insanely easy-to-use honeypot solution that deploys in just four minutes. Know. When it matters.

© Thinkst Canary 2015–2022

30



31

MORE CUSTOMIZATION

Cowrie

- The obvious

- Hostname (and MAC!)
 - openssl rand -hex 6 | sed 's/\(..\)/\1:/g; s/:\$//'
 - 00:11:32:B3:4D:F5
- Versions
- History
- Commands
 - History -s (or just copy .bash_history)
 - Targets?
- Filesystem
- Processes
- Usernames
 - Honeycreds

Monitoring!

32

CUSTOMIZATION EXAMPLES

- Banners = easy, but don't forget ssh headers/ciphers/version
- Ping?
- DNS?
- rsync is your friend – honeyfs / createfs
- ps a running system (cmdoutput.json)
 - Command, cpu, mem, pid, rss, start, stat, time, tty, user, vsz

```
$ ps -eo pcpu,%mem,pid,rss,start_time,stat,bsdtimer,tty,user,vsz,args
%CPU %MEM PID RSS START STAT TIME TT USER VSZ COMMAND
0.0 0.0 14995 4456 03:32 S 0:00 ? dovecot 50052 dovecot/imap-login [67.18.92.27 TLS proxy]
0.0 0.0 15034 3500 03:32 S 0:00 ? dovecot 49784 dovecot/imap-login
0.2 0.0 15154 91232 03:35 S 0:51 ? apache 383516 /usr/sbin/httpd -DFOREGROUND
0.0 0.0 15525 4868 Feb04 Ss 6:14 ? root 279644 php-fpm: master process
0.0 0.0 15533 6816 Feb04 S 0:00 ? emps 280408 php-fpm: pool ordinary
```
- ps -eo pcpu,%mem,pid,rss,start_time,stat,bsdtimer,tty,user,vsz,args | egrep -v '(ps - eo|jq|egrep|awk|wazuh|ossec|kat8|dovecot|exim|qemu)' | awk '{for(i=1;i<=10;i++){printf "%s\t",\$i}out=\$11; for(i=12;i<=NF;i++){out=out" "\$i}; print out}' | jq -s --slurp --raw-input -- raw-output 'split("\n") | .[1:-1] | map(split("\t")) | map({"COMMAND": .[10], "CPU": .[0]|tonumber, "MEM": .[1]|tonumber, "PID": .[2]|tonumber, "RSS": .[3]|tonumber, "START": .[4], "STAT": .[5], "TIME": .[6], "TTY": .[7], "USER": .[8], "VSZ": .[9]|tonumber}) | {"command": { "ps": .}}'

33

CUSTOMIZATION EXAMPLES (CONT)

- Users/passwords
 - rockyou
 - rockyou2021
 - Of course, default
 - Get creative with users
 - htpasswd
- Other “places”
 - Remember the real services/apps?
 - smb
 - rdp
 - ftp/ftps
- pwnt

It would take a computer about
7 QUADRILLION YEARS
 to crack your password

34

KEY TAKEAWAYS

- CCAD
- Low False Positives
 - Defend & Detect
- Lateral Movement
- Cost Effective
- Forensics
- REAL Threat Intelligence
 - It's About Thinking Differently, not "watching everything"

35

THANK YOU!!!

- Kat Fitzgerald
 - @rnbwkat
 - @rnbwkat@infosec.exchange
 - evilkat@rnbwmail.com



36