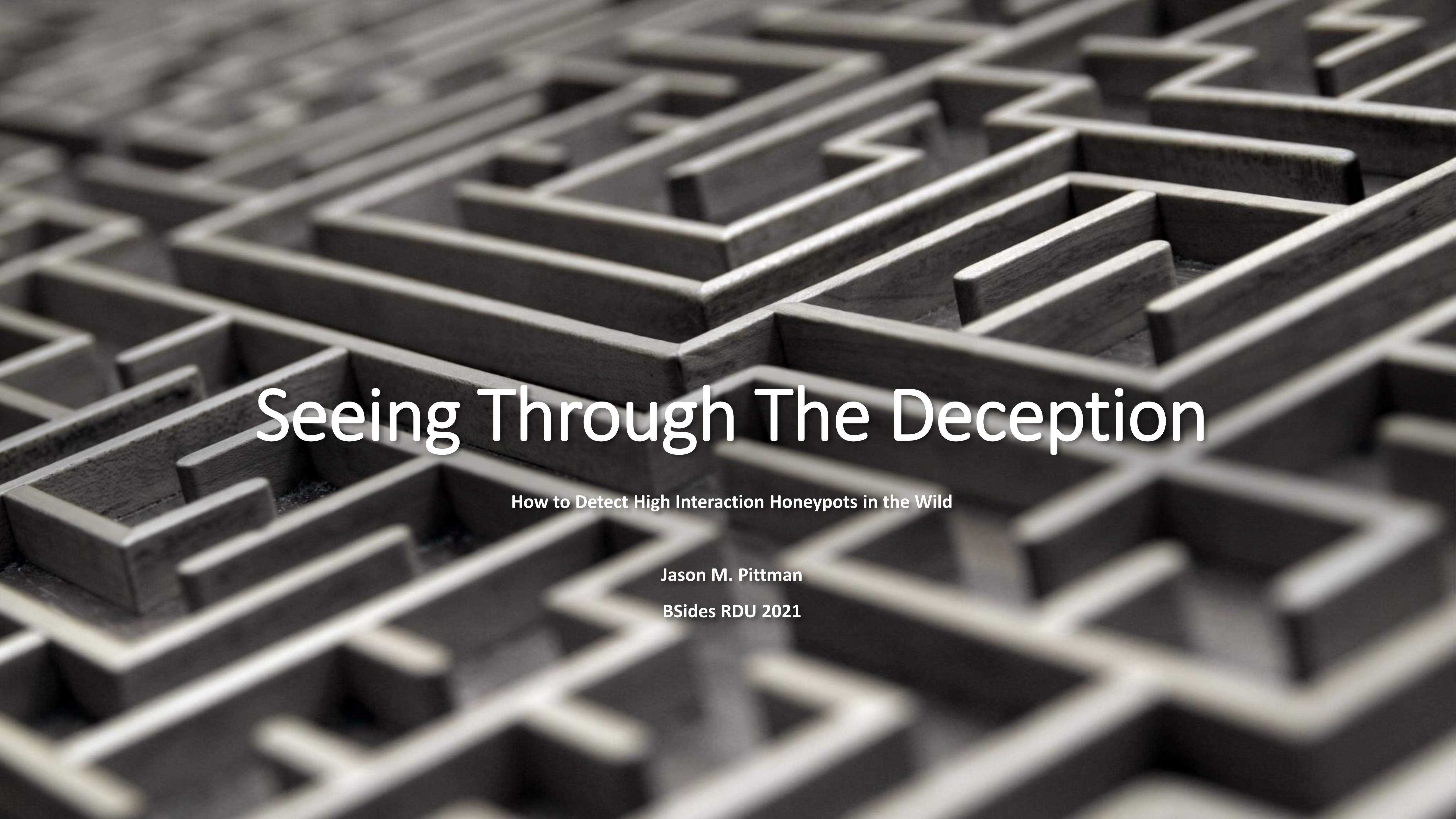


Seeing Through The Deception

How to Detect High Interaction Honeypots in the Wild

Jason M. Pittman

BSides RDU 2021

A little about me…

- 20 years in tech
- 10 years in academia
- Undergrad in English & Biology
- Grad & Doctorate in Cyber

A little about honeypots…

- What

- Why

- How

What makes a *good* honeypot…?

A little about detecting honeypots…

- For researchers…
- For adversaries…
- For the curious and adventurous

Related work…

- Model

- Results

Why not leave it alone…?

- Interest is involuntary

- Potential

- Difficulty

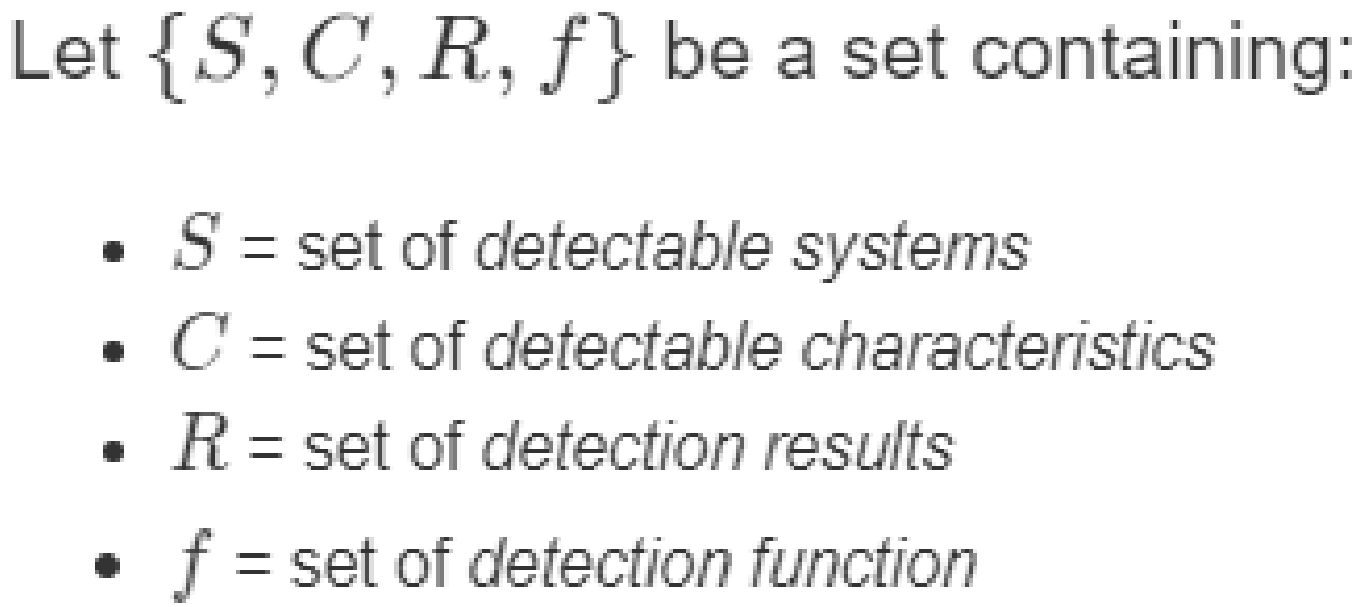
What I did…

- Develop model & identify likely characteristics
- Scan IPs (lots of IPs!)
- Ingest into detection model
- Validation experiments

How I did it…

- Nmap (full connect, OS detection, output to file)
- tcpdump (full packet capture)
- nmaptocsv (modified)
- Skull sweat (and hacking some sloppish code)

This is what the model looks like…



Let $\{S, C, R, f\}$ be a set containing:

- S = set of *detectable systems*
- C = set of *detectable characteristics*
- R = set of *detection results*
- f = set of *detection function*

$$R = f\{c(s)\} = [0|1] s \in S, c \in C$$

What are the constraints…?

- Blind
- Remote
- Passive and active.
- Fingerprinting

What characteristics can we use…?

"I want a detailed analysis, your best educated guess, and then round it out with some wild speculation."

The characteristics I selected…

- Connection
- State
- Behavior*

What I scanned…

- 184.75.224.0/20
- 199.101.120.0/21
- 206.195.144.0/20
- 216.237.192.0/19
- 216.237.224.0/20
- 66.110.224.0/20
- 66.110.240.0/20
- 68.67.240.0/20
- 72.11.32.0/20
- 72.11.48.0/20
- 74.124.160.0/20
- 74.124.176.0/20
- 97.75.128.0/20
- 97.75.144.0/20

My analysis protocol…

- Extract sample from raw data (nmaptocsv)
- Isolate 22/tcp
- Apply selection characteristics

The detection function became…

- Size of banner (f_1)
- Algorithms in protocol fingerprint (f_2)
- Empty login reaction (f_3)

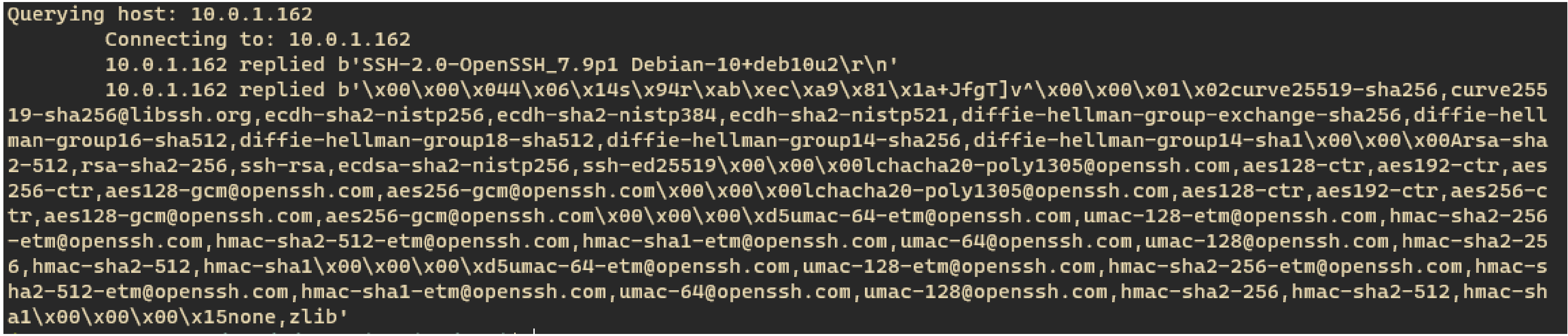
Validation experiments…

- First phase: LAN-manual
- Second phase: LAN-(semi) automated
- Third phase: Internet-(semi) automated

The results are in…

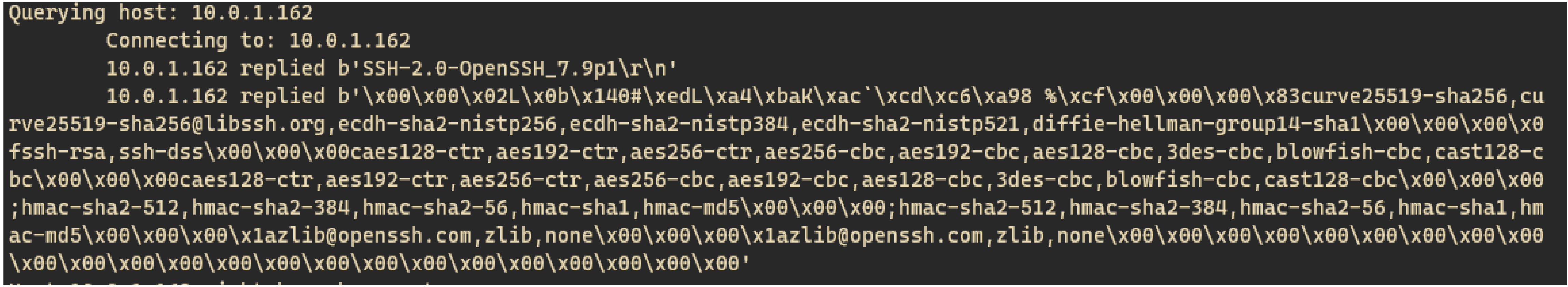
- Total number of hosts
- Hosts running SSH
- Hosts *similar* to a known honeypot
- Interesting notes

Banner examples…



```
Querying host: 10.0.1.162
        Connecting to: 10.0.1.162
        10.0.1.162 replied b'SSH-2.0-OpenSSH_7.9p1 Debian-10+deb10u2\r\n'
        10.0.1.162 replied b'\x00\x00\x044\x06\x14s\x94r\xab\xec\xa9\x81\x1a+JfgT]v^\x00\x00\x01\x02curve25519-sha256,curve255
19-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group-exchange-sha256,diffie-hell
man-group16-sha512,diffie-hellman-group18-sha512,diffie-hellman-group14-sha256,diffie-hellman-group14-sha1\x00\x00\x00Arsa-sha
2-512,rsa-sha2-256,ssh-rsa,ecdsa-sha2-nistp256,ssh-ed25519\x00\x00\x00lchacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes
256-ctr,aes128-gcm@openssh.com,aes256-gcm@openssh.com\x00\x00\x00lchacha20-poly1305@openssh.com,aes128-ctr,aes192-ctr,aes256-c
tr,aes128-gcm@openssh.com,aes256-gcm@openssh.com\x00\x00\x00\xd5umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256
-etm@openssh.com,hmac-sha2-512-etm@openssh.com,hmac-sha1-etm@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-25
6,hmac-sha2-512,hmac-sha1\x00\x00\x00\xd5umac-64-etm@openssh.com,umac-128-etm@openssh.com,hmac-sha2-256-etm@openssh.com,hmac-s
ha2-512-etm@openssh.com,hmac-sha1-etm@openssh.com,umac-64@openssh.com,umac-128@openssh.com,hmac-sha2-256,hmac-sha2-512,hmac-sh
a1\x00\x00\x00\x15none,zlib'
```

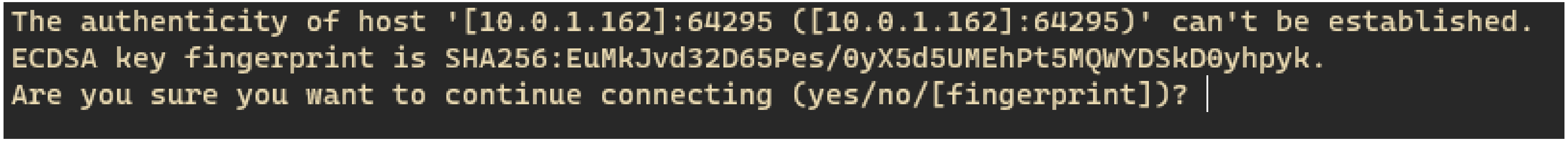
A



```
Querying host: 10.0.1.162
        Connecting to: 10.0.1.162
        10.0.1.162 replied b'SSH-2.0-OpenSSH_7.9p1\r\n'
        10.0.1.162 replied b'\x00\x00\x02L\x0b\x140#\xedL\xa4\xbaK\xac`\xcd\xc6\xa98 %\xcf\x00\x00\x00\x83curve25519-sha256,cu
rve25519-sha256@libssh.org,ecdh-sha2-nistp256,ecdh-sha2-nistp384,ecdh-sha2-nistp521,diffie-hellman-group14-sha1\x00\x00\x00\x0
fssh-rsa,ssh-dss\x00\x00\x00caes128-ctr,aes192-ctr,aes256-ctr,aes256-cbc,aes192-cbc,aes128-cbc,3des-cbc,blowfish-cbc,cast128-c
bc\x00\x00\x00caes128-ctr,aes192-ctr,aes256-ctr,aes256-cbc,aes192-cbc,aes128-cbc,3des-cbc,blowfish-cbc,cast128-cbc\x00\x00\x00
;hmac-sha2-512,hmac-sha2-384,hmac-sha2-56,hmac-sha1,hmac-md5\x00\x00\x00;hmac-sha2-512,hmac-sha2-384,hmac-sha2-56,hmac-sha1,hm
ac-md5\x00\x00\x00\x1azlib@openssh.com,zlib,none\x00\x00\x00\x1azlib@openssh.com,zlib,none\x00\x00\x00\x00\x00\x00\x00\x00\x00
\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00\x00'
```

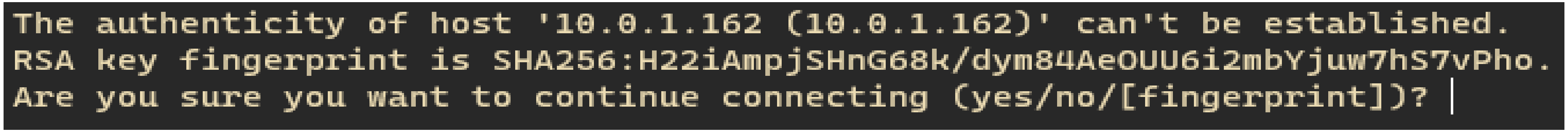
B

Algorithms in protocol fingerprint examples…



```
The authenticity of host '[10.0.1.162]:64295 ([10.0.1.162]:64295)' can't be established.
ECDSA key fingerprint is SHA256:EuMkJvd32D65Pes/0yX5d5UMEhPt5MQWYDSkD0yhpyk.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

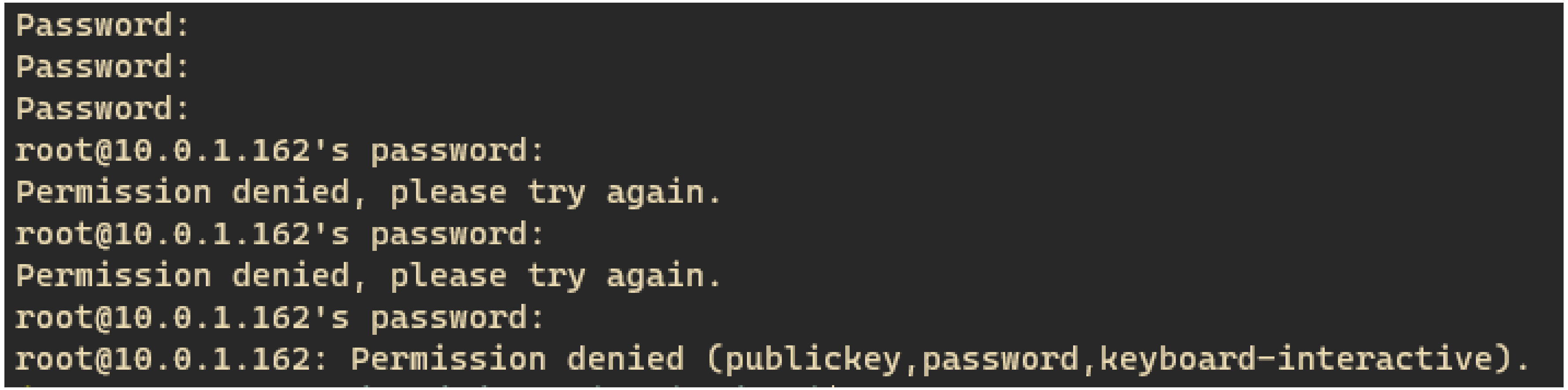
A



```
The authenticity of host '10.0.1.162 (10.0.1.162)' can't be established.
RSA key fingerprint is SHA256:H22iAmpjSHnG68k/dym84AeOUU6i2mbYjuw7hS7vPho.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

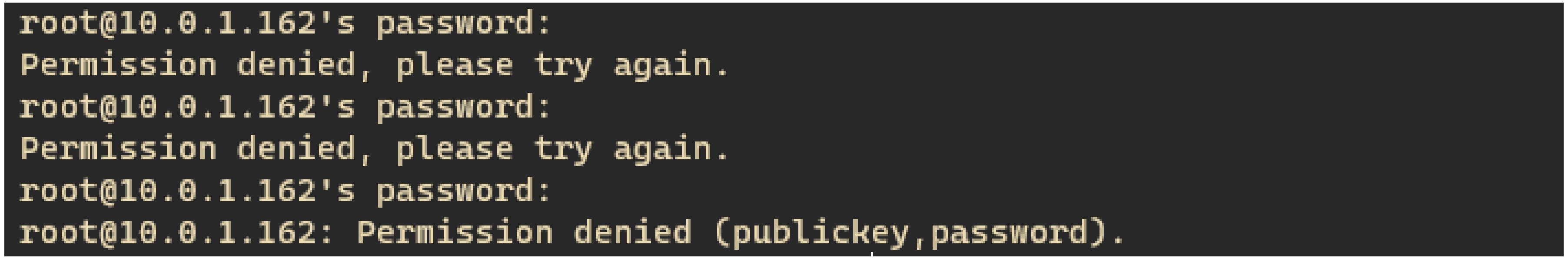
B

Empty login reaction examples…



```
Password:
Password:
Password:
root@10.0.1.162's password:
Permission denied, please try again.
root@10.0.1.162's password:
Permission denied, please try again.
root@10.0.1.162's password:
root@10.0.1.162: Permission denied (publickey,password,keyboard-interactive).
```

A



```
root@10.0.1.162's password:
Permission denied, please try again.
root@10.0.1.162's password:
Permission denied, please try again.
root@10.0.1.162's password:
root@10.0.1.162: Permission denied (publickey,password).
```

B

Hosts detected…

- Population: 59,388

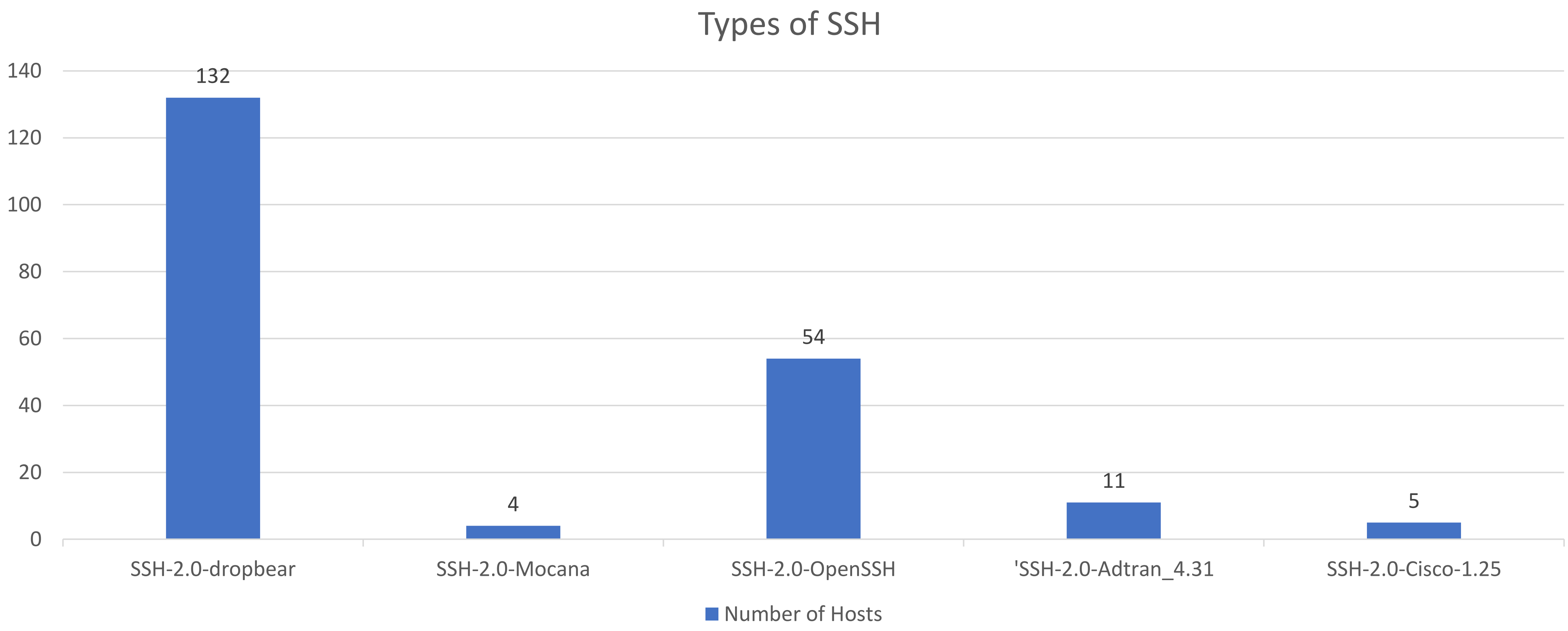
- Sample: 9891*

Hosts detected with SSH…

- Down sample: 405

- Further down: 216

Types of SSH hosts…



Hosts *similar* to a honeypot…

- 7 or *0.01% of the total population*

- Conditional probability = similarity

Some interesting side notes…

- Northstate is volatile
- Timing is unreliable
- Scanning is cool
- Honeypots conceptually flawed

What I told you I’d tell you…

- Develop model & identify likely characteristics
- Scan IPs (lots of IPs!)
- Ingest into detection model
- Validation experiments

What we can make of this…

- Honeypots are real

- The model has value

- Honeypots are flawed

Work for the future…

- Automation

- Next gen honeypots

- Overall aim: *pattern recognition receptors*

Thanks, and Questions?

- All material: https://github.com/jasonmpittman/bsidesrdu-2021
- For later: jason.pittman [at] umgc.edu