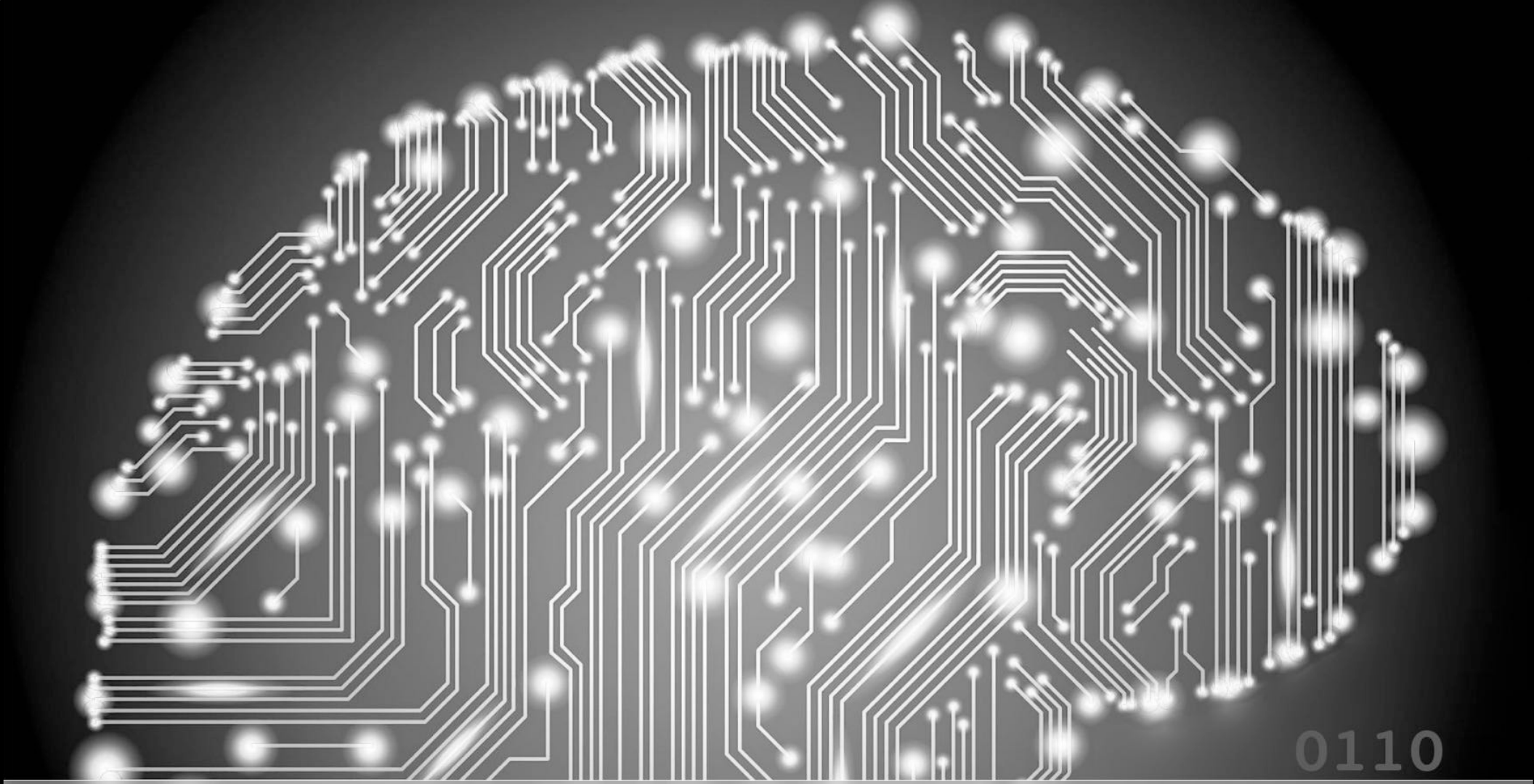


HoneyPi: Network Security

Kevin Capwell - META
Pat Zielke - Viroqua

HoneyPi Monitoring

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...

HoneyPi Monitoring

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.

HoneyPi Monitoring

Viroqua Area Schools

- Pat Zielke
  Technology Coordinator – 20 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 800, Other mobile 90.

HoneyPi Monitoring

Kev and Pat have a security chat…

- Good security comes in layers
  - *NG Firewall, AV (x2), sandboxing & alerts*
- Passwords – good, bad and ugly
- Patch / vulnerability scan (Nessus)
- Penetration testing (Kali)
  - *Use a test environment / deploy honeypot*
- Incoming / outgoing traffic (SNMP)
- Centralized management tools
- Monitor top user statistics
- Check / test backups / offsite
- Logs, Logs, Logs!

HoneyPi Monitoring

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.

HoneyPi Monitoring

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.

HoneyPi Monitoring

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.

HoneyPi Monitoring

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.

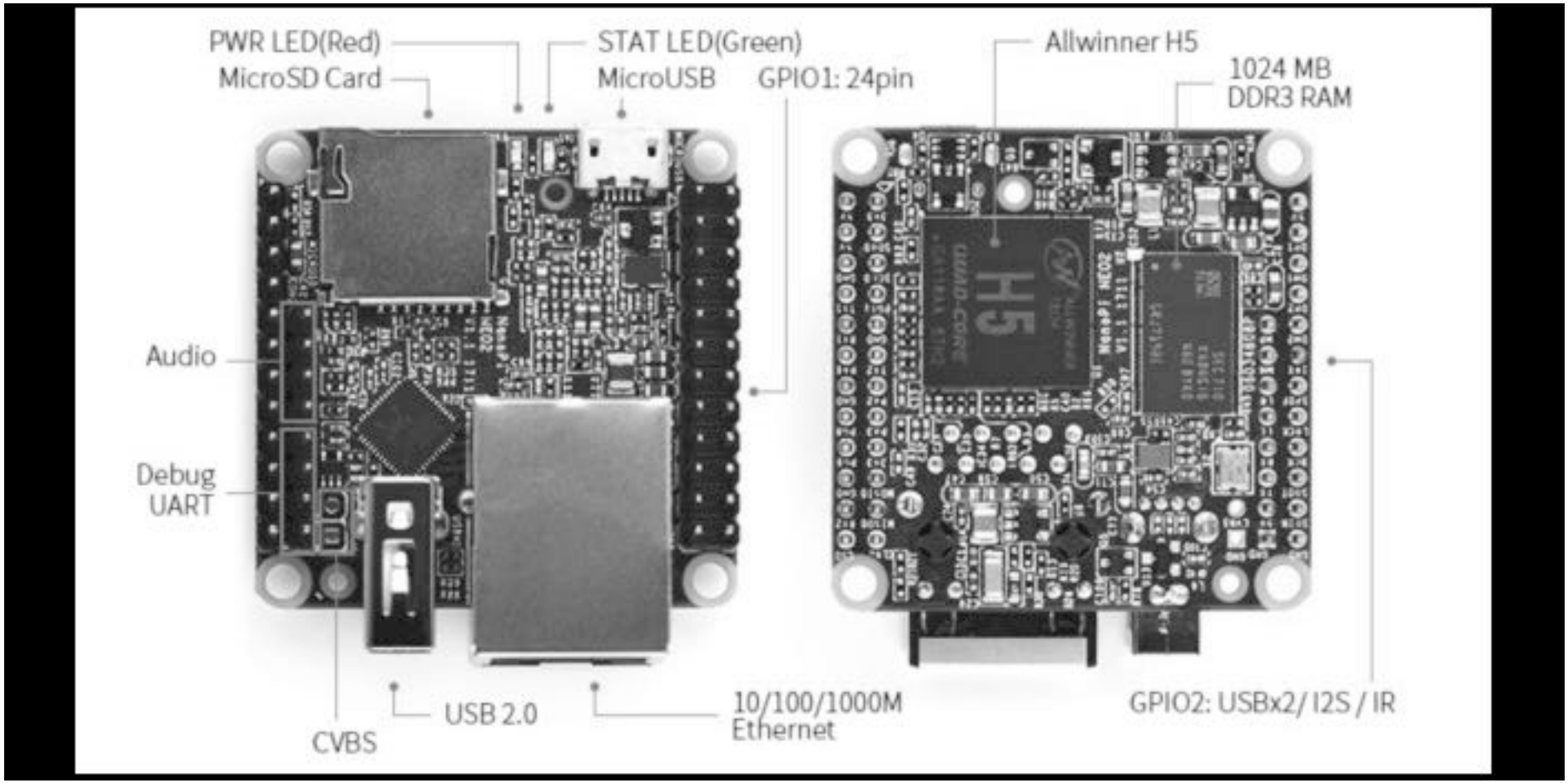
HoneyPi Monitoring

What are the Prerequisites of this Session?

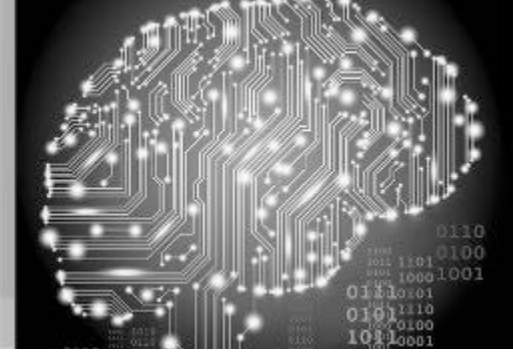
- Familiarity with the basics of Linux.
- Being comfortable working with the CLI.
- Basics of network terminology.
- Familiarity with basic TCP ports 80, 22 and 23.
- Knowledgeable with Win10 Pro.
- Need to protect your network’s servers.
- Proactively monitor your network’s security.

HoneyPi Monitoring

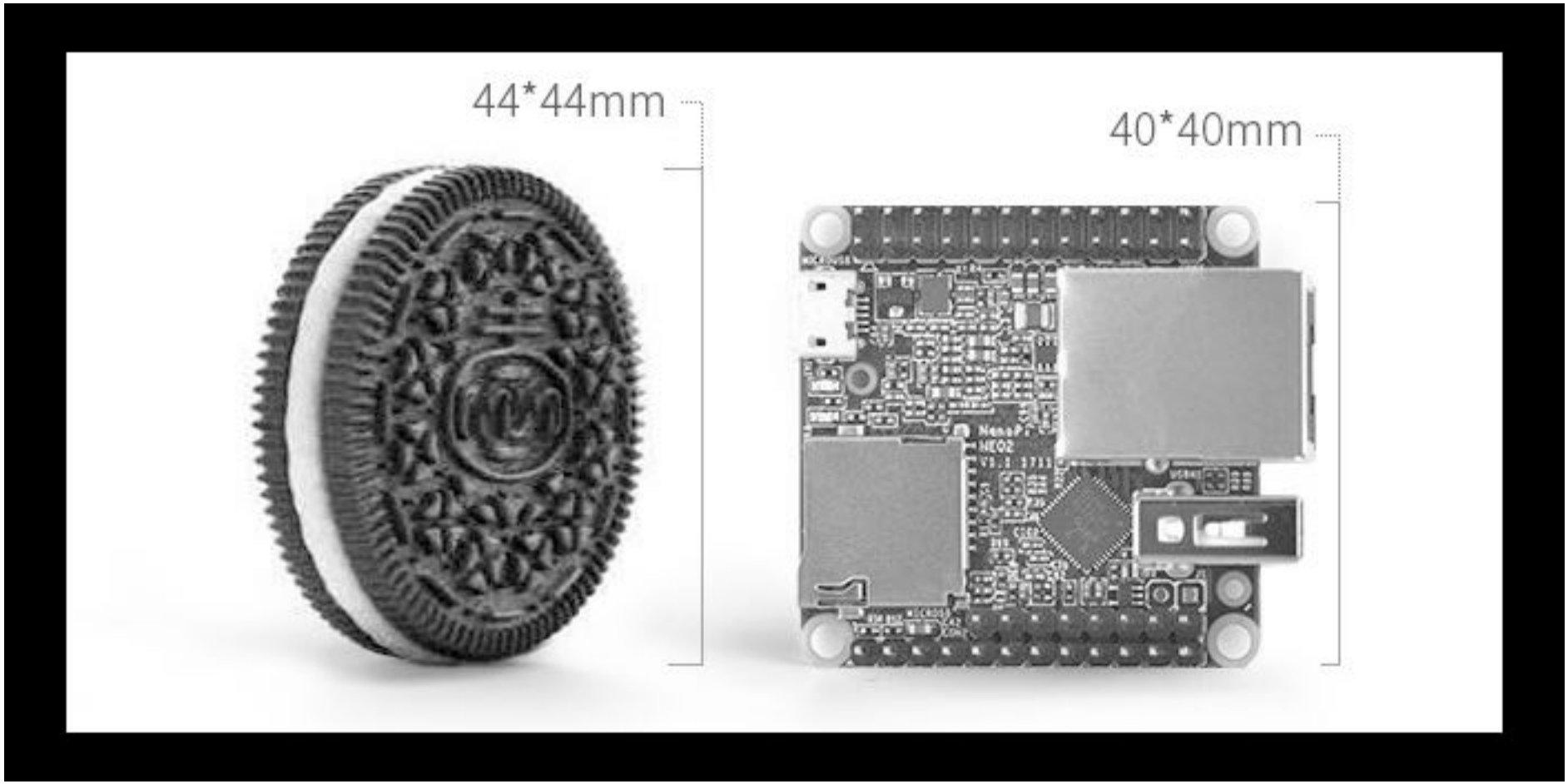
Overview of the NanoPi NEO2



HoneyPi Monitoring



Let’s show this at scale…



HoneyPi Monitoring

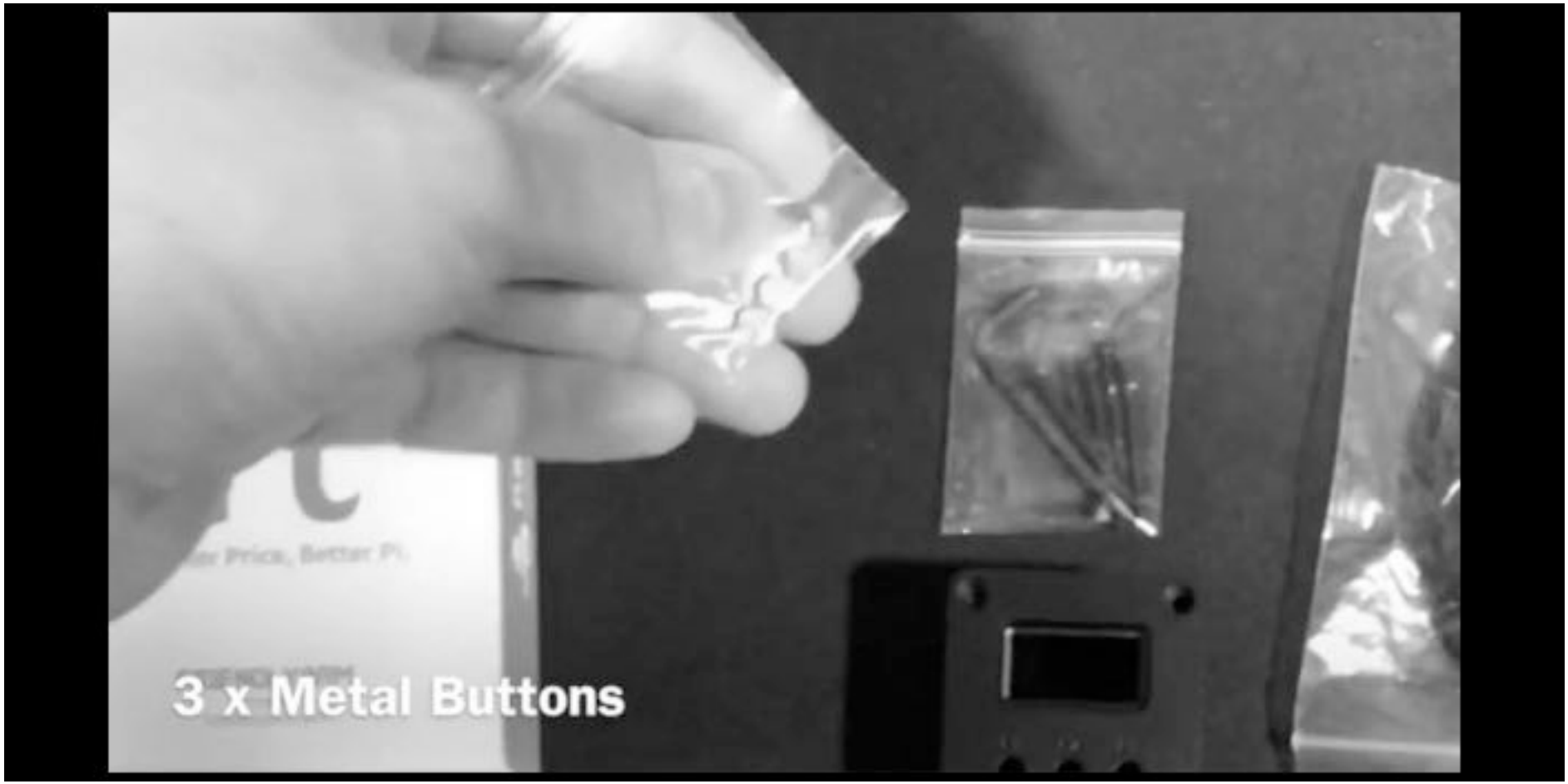
Mind = blown!

HoneyPi Monitoring

Let’s review the parts…

HoneyPi Monitoring

Let’s review the parts…

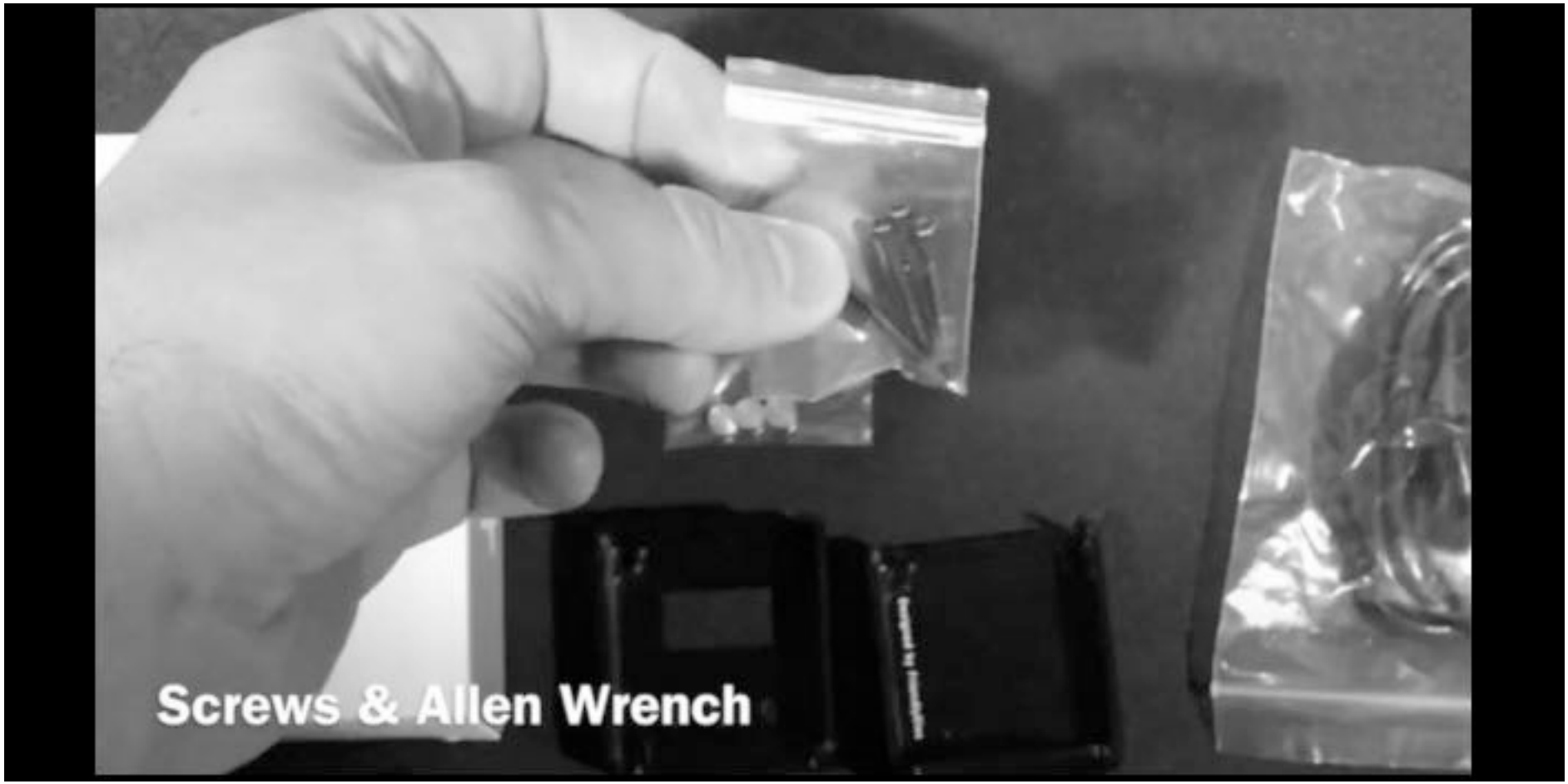


HoneyPi Monitoring

Let’s review the parts…

HoneyPi Monitoring

Let’s review the parts…



HoneyPi Monitoring

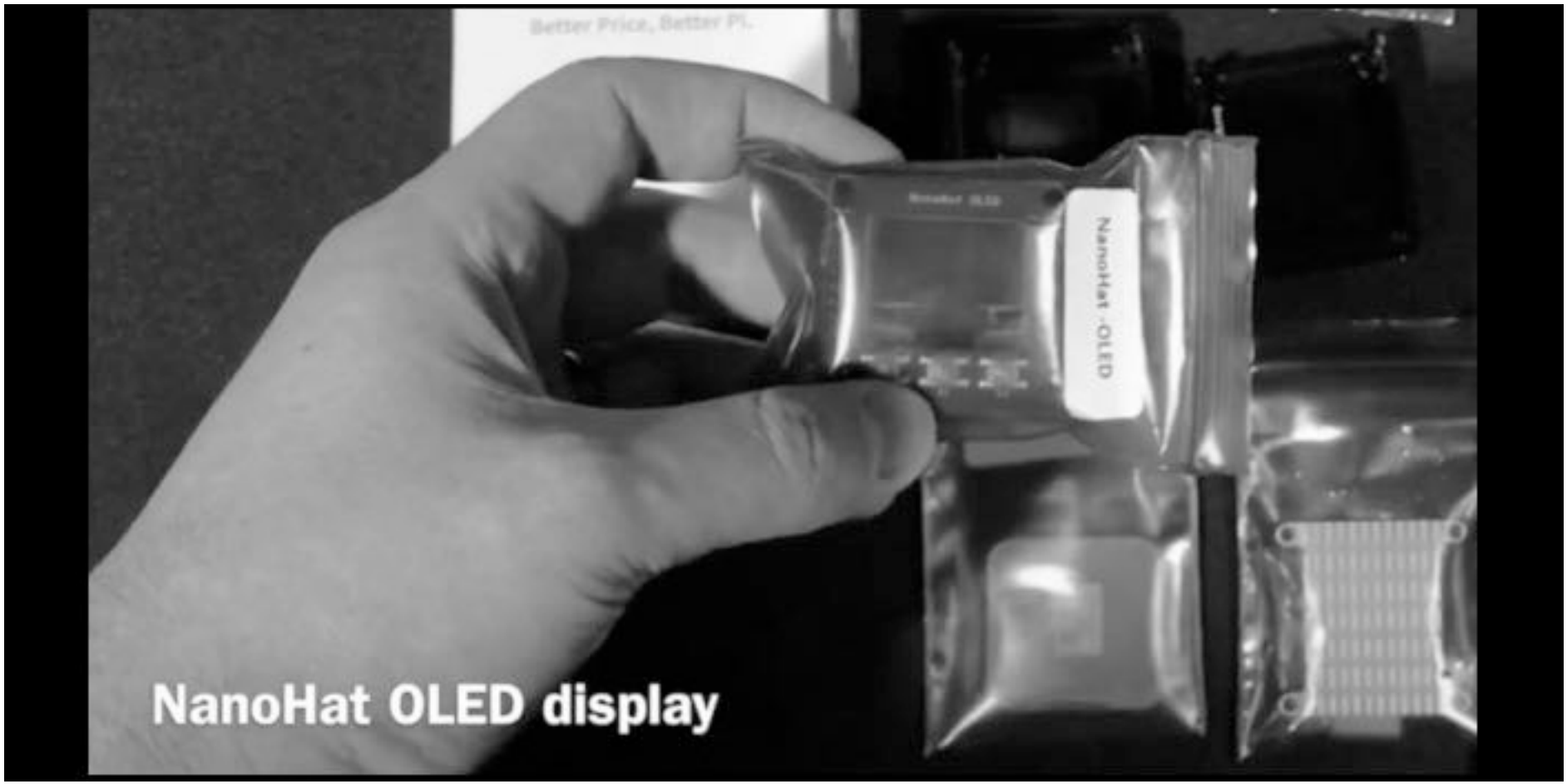
Let’s review the parts…

HoneyPi Monitoring

Let’s review the parts…

HoneyPi Monitoring

Let’s review the parts…

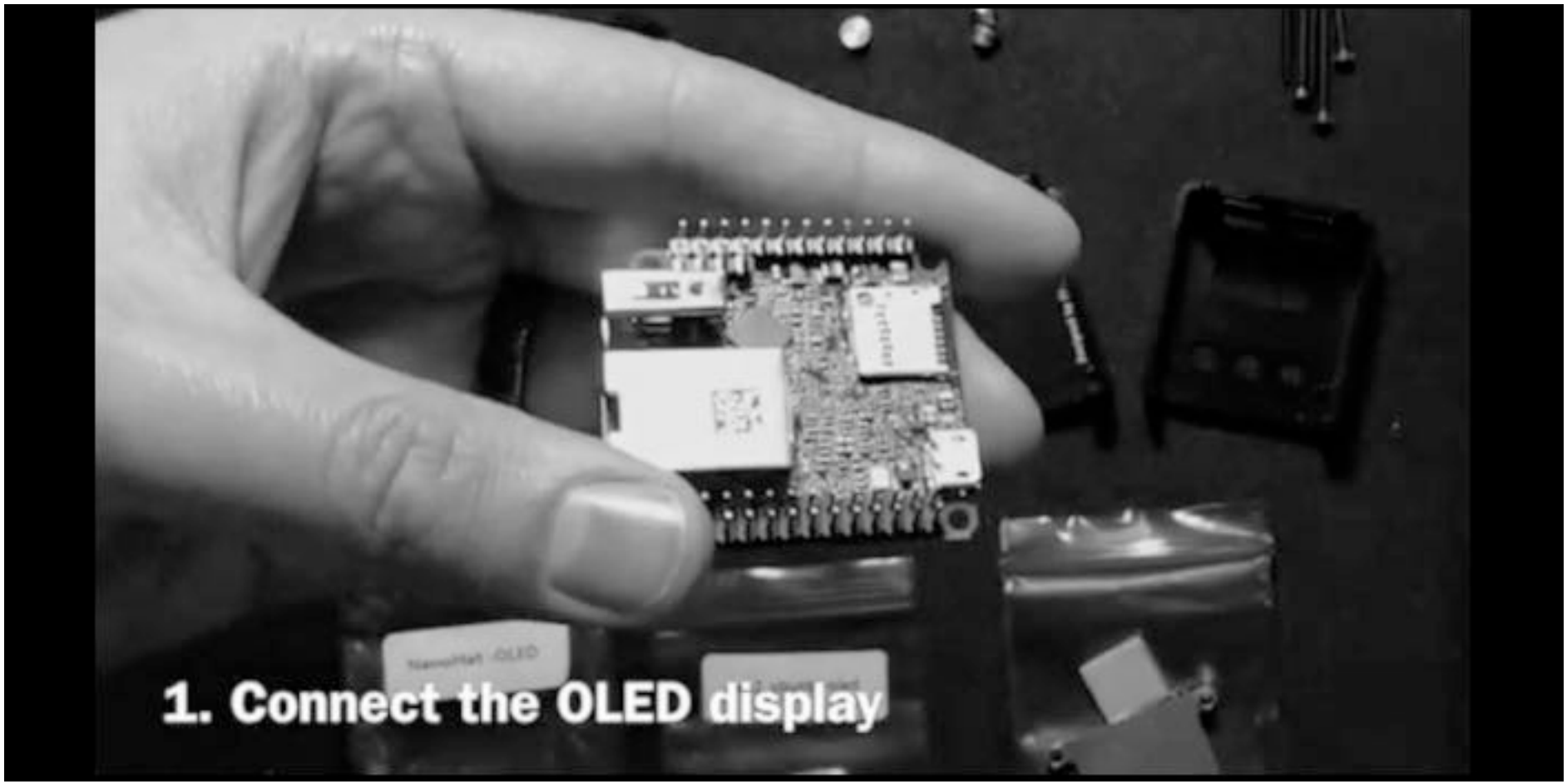


HoneyPi Monitoring

Can we build it?

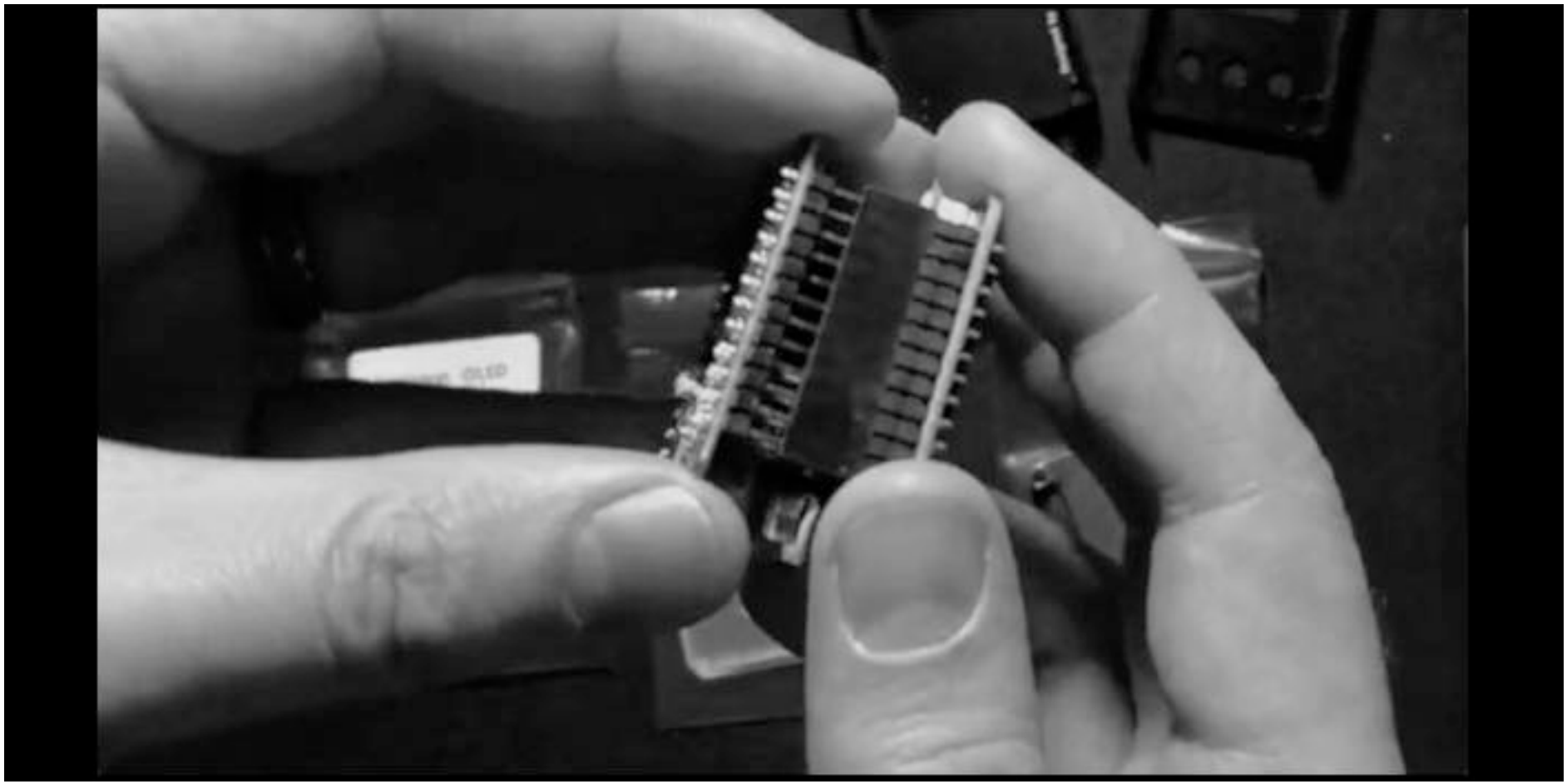
HoneyPi Monitoring

Open the NEO2, note the pins…



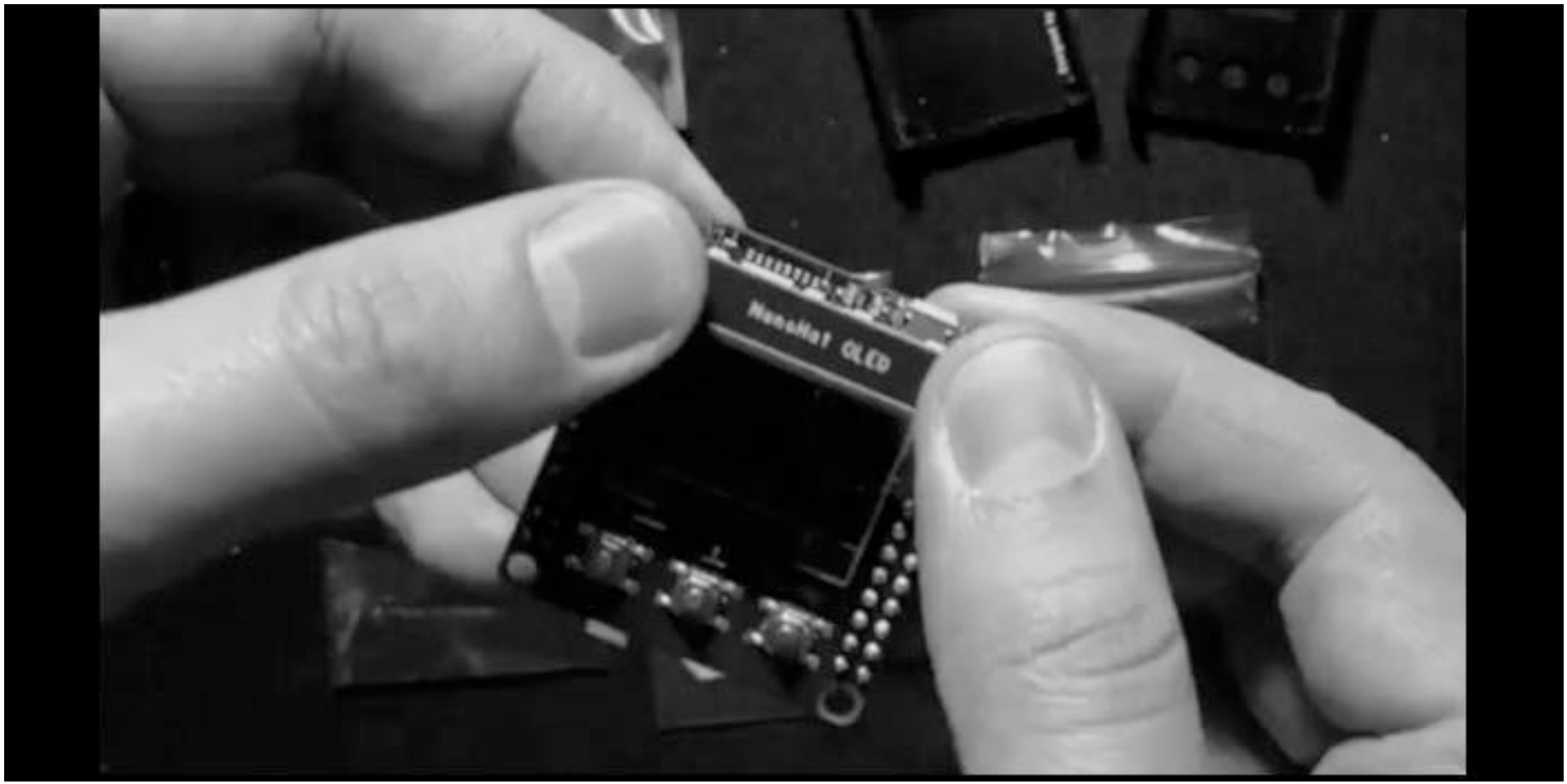
HoneyPi Monitoring

Align the pins on the OLED hat



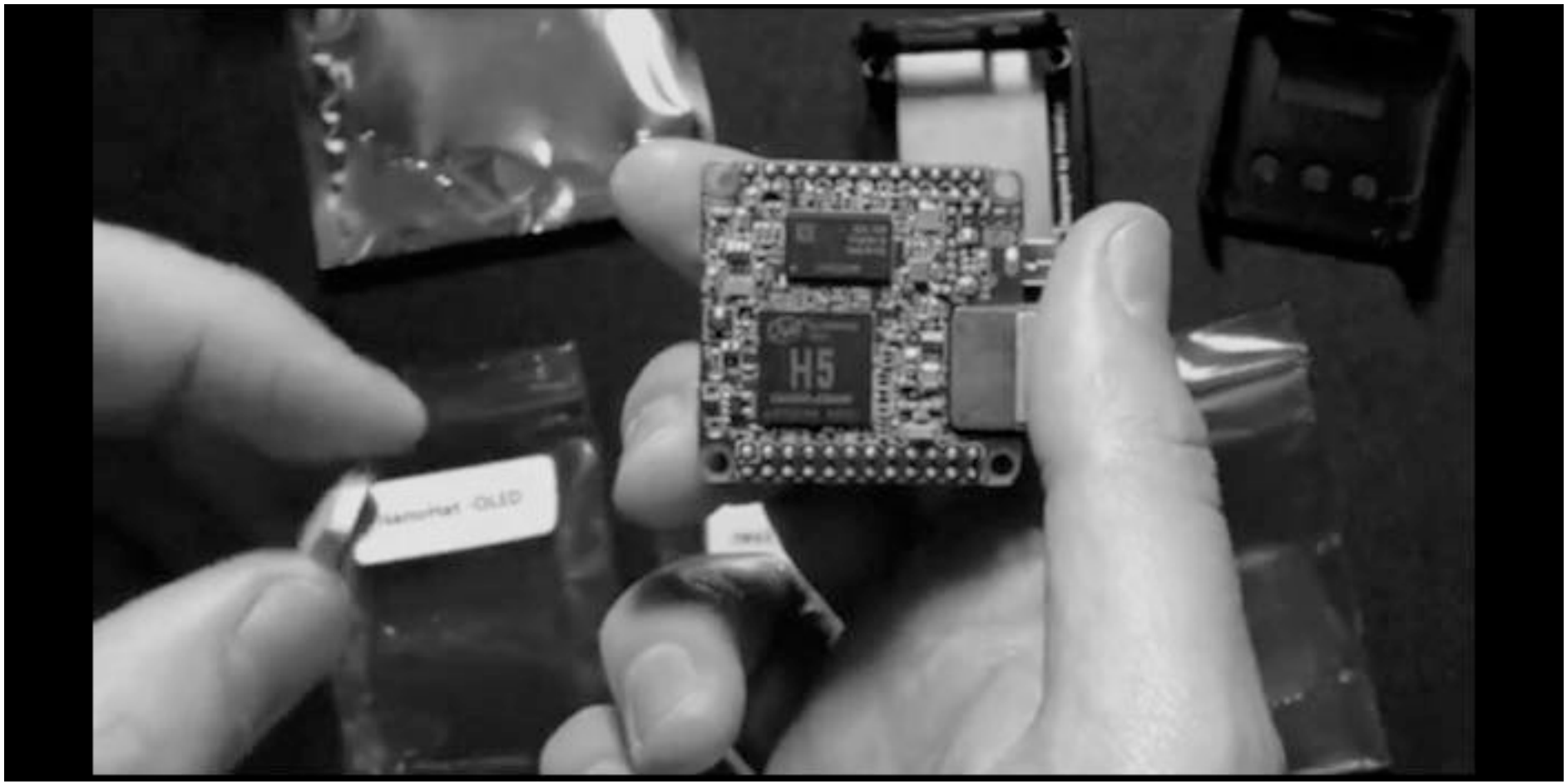
HoneyPi Monitoring

Do *not* press on the screen!



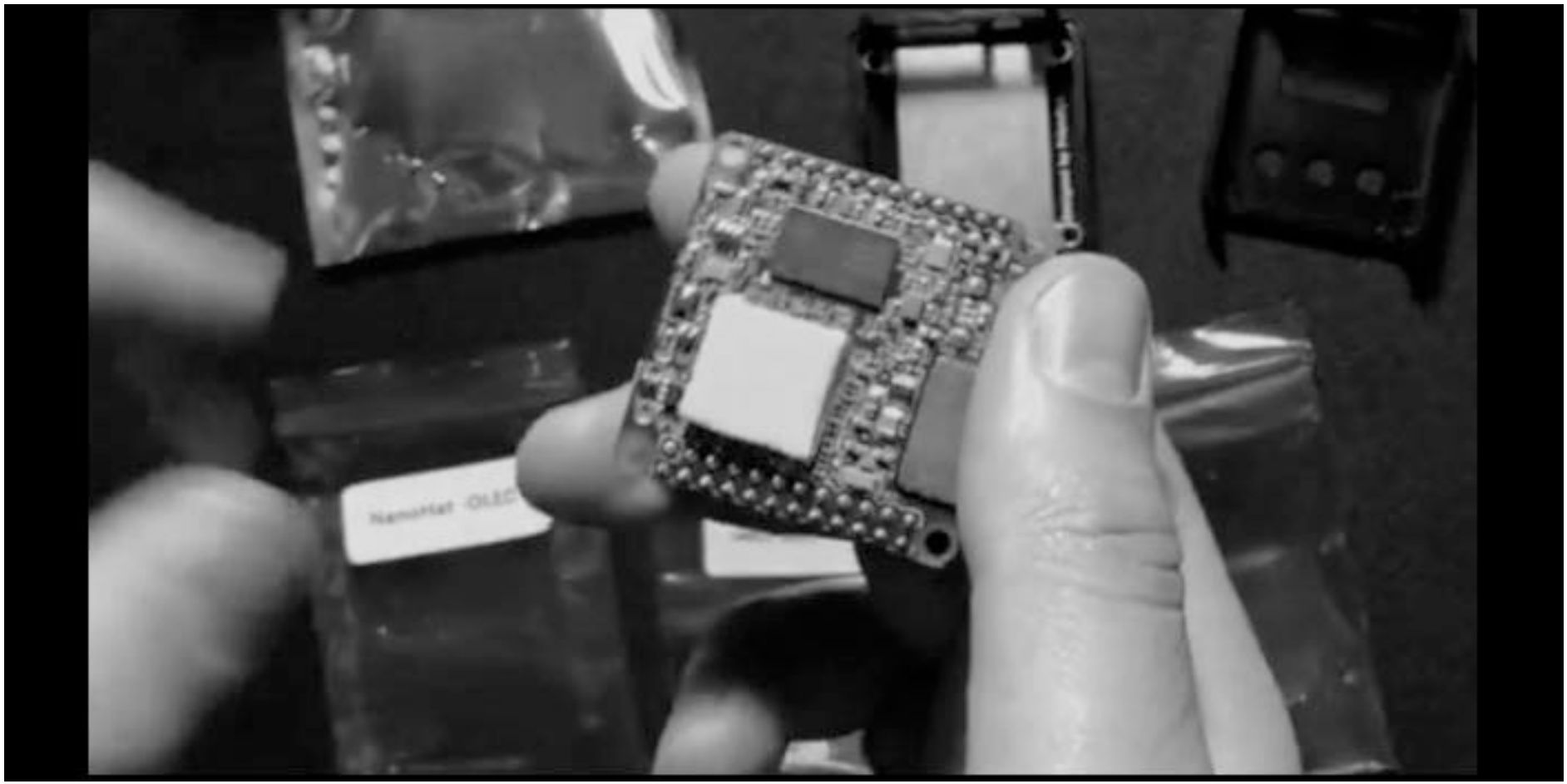
HoneyPi Monitoring

Install thermal pad on H5 SoC



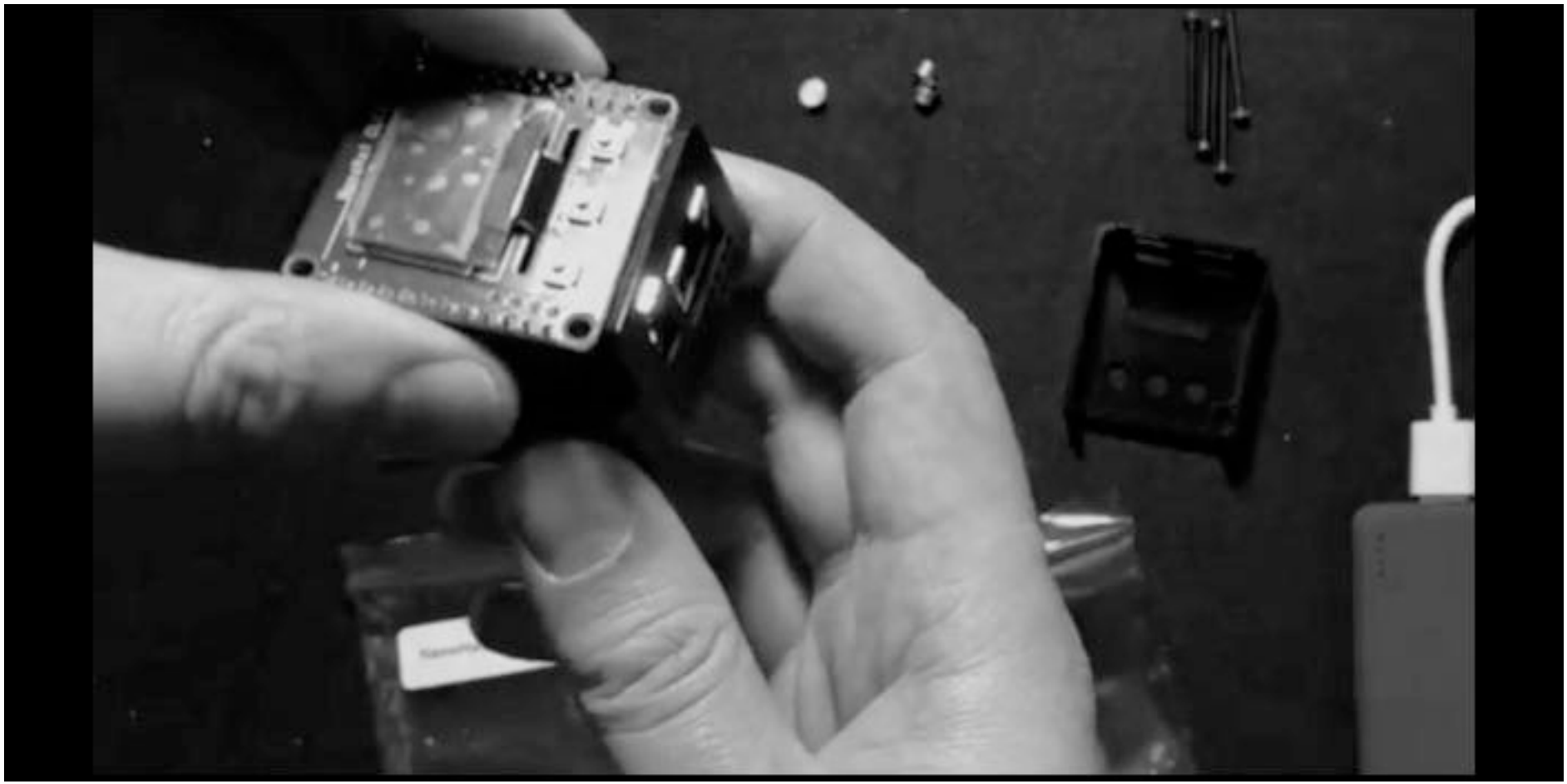
HoneyPi Monitoring

Now, remove the second film



HoneyPi Monitoring

Slide the NEO2 into the bottom case



HoneyPi Monitoring

Buttons should be flush with the top

HoneyPi Monitoring

Secure all four screws

HoneyPi Monitoring

All done! No display – yet…

HoneyPi Monitoring

Ubuntu Core 16.04.6 LTS (Xenial)

- Source model: Open-source
- Version 16.04 LTS was released on April 21, 2016
- Version 16.04.6 LTS was released in February 28, 2019
- Support transitions to "End of Life" on April 2024
- Requirements: 500 MHz single core processor, 256 MB RAM and 512 MB storage.

HoneyPi Monitoring

Our First Example: PenTBox 1.8

- PenTBox 1.8: Open Source – cost $0
- Requires: Ruby scripting language
- Best as web and telnet honeypot
- Hash Password Cracker
- Net Denial of Service (DoS) Tester
- TCP Port Scanner
- Honeypot
  - *must run PenTBox with root privileges!*
- Fuzzer (test buffer overflows)

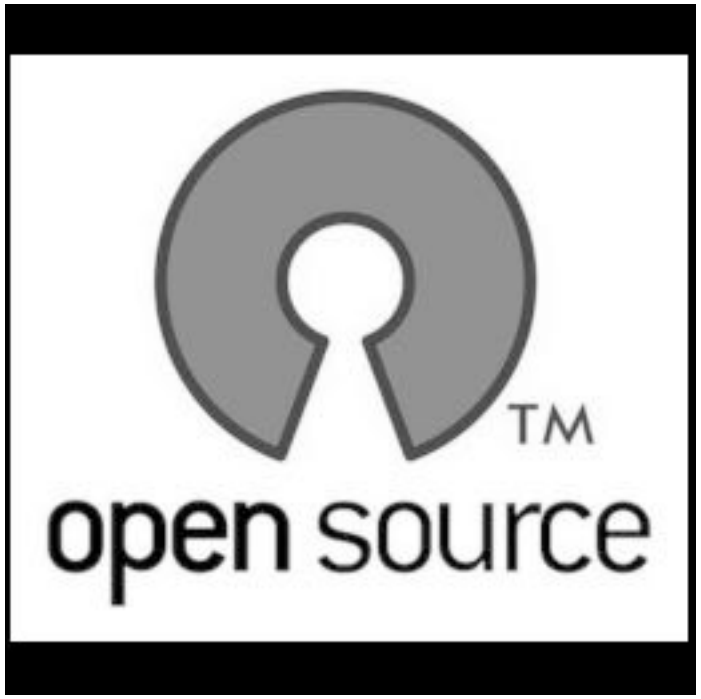
HoneyPi Monitoring

Our Second Example: Cowrie 2.0.0

- Cowrie 2.0.0: Open Source – cost $0
- Requires: GCC, Python, git, pip, python-virtualenv, pycrypto
- Best as ssh and telnet honeypot
- Virtual filesystem displays Debian 7.0
- 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.

HoneyPi Monitoring

The Supporting Cast…



- balenaEtcher 1.5.76
  "Burn" images to SD cards and USB drives
- Firefox ESR 68.5.0
  Browser for institutions: K-12 & Higher Ed
- Advanced IP Scanner 2.5.3850
  Fast and free IP scanner for Windows
- PuTTY 0.73
  Free implementation of SSH and Telnet
- WinSCP 5.15.9
  A popular SCP, SFTP & FTP client

HoneyPi Monitoring

Important Items to Remember!

- SSID of test network: HoneyPi
- Password for Wi-Fi: BrainStorm2020

- ByteSpeed laptops: user: admin | pass: Ø → running Windows 10 Pro.

- NEO2: user: **root** | pass: **Empress905**
- NEO2: user: **pi** | pass: **Ebony509**

HoneyPi Monitoring

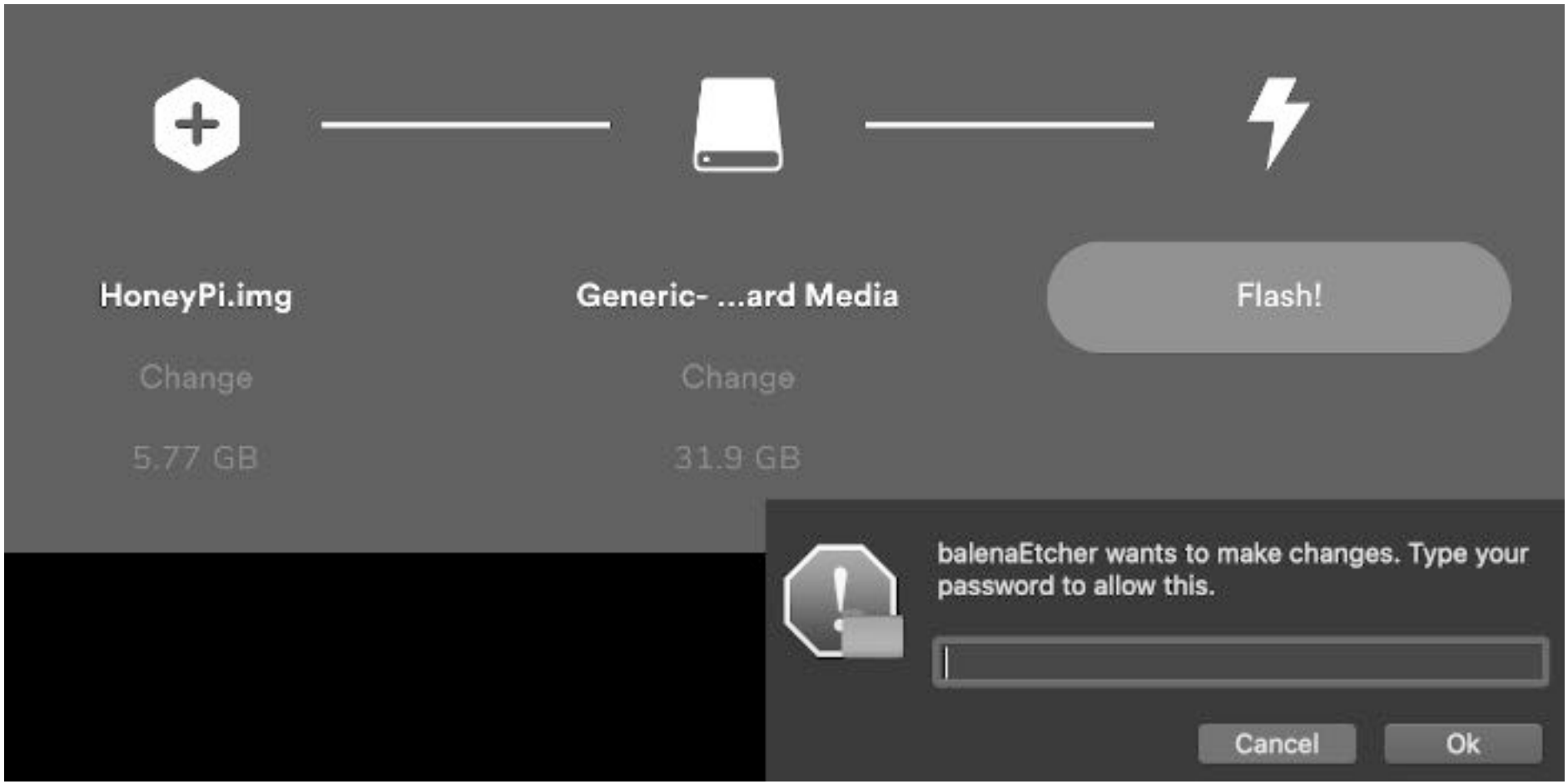
"Burning" Ubuntu Core to your microSD

- Insert the microSD into USB reader
- Place the reader into the USB 3.0 port
  (## if the system asks you to format, click ignore)

- Select the HoneyPi.img file
  - *Desktop → HoneyPi → HoneyPi.img*
- Confirm your microSD card is selected
- Click Flash *(## may ask for password)*
- When the copy has completed **successfully** (~6-8 min), quit Etcher

HoneyPi Monitoring

Using Etcher to Create the Boot Drive



HoneyPi Monitoring

Ready?

HoneyPi Monitoring

Let's Go!

HoneyPi Monitoring

Putting it all Together

- Insert the microSD card into the NEO2
  *(## **Note**: gap at the case top and the card slot!)*
- Connect your Ethernet cable to the Gigabit PoE Splitter.
- Connect the Gigabit Ethernet dongle to your NEO2 Gigabit Ethernet port.
- Finally, connect the Micro USB splitter dongle to the NEO2.
- The NEO2 can take between 20-40 seconds for the OLED to activate.

HoneyPi Monitoring

Putting it all Together

- The hostname of the NanoPi NEO2 will appear at top *(## NanoPi-NEO2)*.
- The NEO2 will register the link on the GbE port, however, it will take a few seconds for the IP address to appear.
- The DHCP server delivers a TCP/IP address to the NEO2 via Ubuntu Core.
- The switch, splitter and NEO2 are all GbE so 1000Mb/s will display onscreen.
- The F3 button will activate the menu.

HoneyPi Monitoring

NanoPi – Main Menu

- This is the NanoPi NEO2 Main Menu.
- System sub-menu.
- F1 button moves the menu selection down. The system will loop at the end back to the top of the list.
- F2 button will step one level deeper into the menu system.
- F3 button will step one level higher into the menu system.

HoneyPi Monitoring

NanoPi – System Menu – pg 1

- Shutdown sub-menu.
- Reboot sub-menu.
- Summary sub-menu (TCP/IP address, CPU, memory, disk and CPU temp).

- F1 Down, F2 Next, F3 Back.

HoneyPi Monitoring

NanoPi – System Menu – pg 2

- Summary sub-menu (TCP/IP address, CPU, memory, disk and CPU temp).
- Date/Time sub-menu (with current time zone setting).
- Version sub-menu (menu version display).

- F1 Down, F2 Next, F3 Back.

HoneyPi Monitoring

NanoPi – Shutdown Menu

- Cancel this command and return to the System Menu.
- Confirm the shutdown command and power down the NanoPi NEO2.

- F1 Down, F2 Next, F3 Back.

HoneyPi Monitoring

NanoPi – Reboot Menu

- Cancel this command and return to the System Menu.
- Confirm the reboot command and reboot down the NanoPi NEO2.

- F1 Down, F2 Next, F3 Back.

HoneyPi Monitoring

NanoPi – Summary Menu

- TCP/IP address.
- CPU Load - on a multicore system, your load should not exceed the number of cores available.
- Memory utilization.
- Disk utilization.
- CPU temperature in Celsius.

- F3 Exit.

HoneyPi Monitoring

NanoPi – Date/Time Menu

- Day of the Week.
- Date Month Year.
- Time in (24 hour format).
- Time Zone.

- F3 Back.

HoneyPi Monitoring

NanoPi – Menu Version

- Day of the Week.
- Date Month Year.
- Time in (24 hour format).
- Time Zone.

- F3 Exit.

HoneyPi Monitoring

General Linux Shortcuts

- Linux info: Prompts are important - # vs $
- Startup: crontab -e • nano /etc/rc.local
- Get version info:
  - cat /etc/*release* • lsb_release -a • uname -a
- Important keys: tab, up and down arrow keys
- CaPitaliZAtion counts!
- Run process in background add & at the end
- tail <filename> will give you the last 10 lines

HoneyPi Monitoring

Connecting to the NanoPi NEO2

- On the PC right click Start → run.
- Type: ping 192.168.136.<yourIP>
  (## 192.168.136.<yourIP> ttl=64 time=4.792 ms)

- PuTTY → SSH to 192.168.136.<yourIP>
- User: root Pass: Empress905

  (## accept the key to the cache - if prompted)
- # passwd
- # apt update
- # apt upgrade
- Cont + D to log out

HoneyPi Monitoring

Banner Page of the NanoPi NEO2

```
 _____     _                 _ _          _____ _     _____ ____
|  ___| __(_) ___ _ __   __| | |_   _  |_   _| |   | ____/ ___|
| |_ | '__| |/ _ \ '_ \ / _` | | | | | | | | |   |  _|| |
|  _|| |  | |  __/ | | | (_| | | |_| | | | | |___| |___| |__| |___
|_|  |_|  |_|\___|_| |_|\__,_|_|\__, | |_| |_____|_____\____|____|
                                 |___/

Welcome to Ubuntu 16.04.6 LTS 4.14.111
System load:   0.07              Up time:       2 days          Local users:   3

Memory usage:  6 % of 992Mb      IP:            192.168.[redacted]
CPU temp:      33°C
Usage of /:    12% of 12G

 * Documentation: http://wiki.friendlyarm.com/Ubuntu
 * Forum: http://www.friendlyarm.com/Forum/
```

HoneyPi Monitoring

Updating the NanoPi NEO2

```
Get:114 http://cdn-fastly.deb.debian.org/debian stretch-backports/non-free armhf
 Packages 2019-02-24-0811.12.pdiff [2,874 B]
Get:115 http://cdn-fastly.deb.debian.org/debian stretch-backports/non-free armhf
 Contents (deb) 2019-02-24-0811.12.pdiff [974 B]
Get:115 http://cdn-fastly.deb.debian.org/debian stretch-backports/non-free armhf
 Contents (deb) 2019-02-24-0811.12.pdiff [974 B]
Get:116 http://cdn-fastly.deb.debian.org/debian stretch-backports/non-free arm64
 Contents (deb) 2019-02-24-0811.12.pdiff [974 B]
Get:116 http://cdn-fastly.deb.debian.org/debian stretch-backports/non-free arm64
 Contents (deb) 2019-02-24-0811.12.pdiff [974 B]
100% [108 Contents-armhf rred 10.5 MB] [57 Packages store 0 B]
```

A

B

```
(Reading database ... 40354 files and directories currently installed.)
Preparing to unpack .../libnss-myhostname_232-25+deb9u9_arm64.deb ...
Unpacking libnss-myhostname:arm64 (232-25+deb9u9) over (232-25+deb9u8) ...
Preparing to unpack .../libpam-systemd_232-25+deb9u9_arm64.deb ...
Unpacking libpam-systemd:arm64 (232-25+deb9u9) over (232-25+deb9u8) ...
Preparing to unpack .../systemd_232-25+deb9u9_arm64.deb ...
Unpacking systemd (232-25+deb9u9) over (232-25+deb9u8) ...

Progress: [ 14%] [########.....................................................]
```

HoneyPi Monitoring

Rebooting the NEO2

- PuTTY → user: pi pass: Ebony509
- $ passwd *(## old pass, new pass, verify)*
- $ sudo apt install [name of package]
- $ sudo apt-cache search [keyword]
- $ sudo apt list --installed
- $ sudo systemctl list-unit-files --type=service | grep enabled

- $ sudo shutdown -r • -h now
- Cont + D to log out

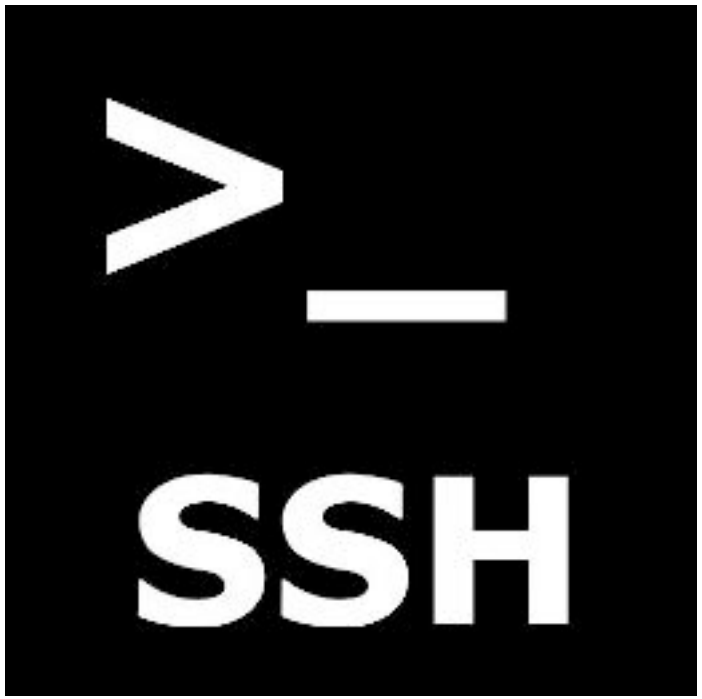
HoneyPi Monitoring

Configuring the NEO2

- PuTTY → user: root pass: <new>
- # npi-config
- 2 Hostname + <Ok>
- Change hostname + tab + <Ok>
- 4 Localization Options + <Ok>
- I2 Change Timezone
  - Set Geographic area / set Time zone
- Press tab twice to <Finish>
- If you are prompted to reboot → + tab + answer <No>

HoneyPi Monitoring

Configuring the SSH Server



- Always make a backup!
  # cp /etc/ssh/sshd_config /etc/ssh/sshd_config.bak
- # nano /etc/ssh/sshd_config
- Change Port 22 to Port 2222
- Cont + X to exit • Save = Y
- # systemctl restart ssh
- Cont + D to log out
- PuTTY → user: root pass: <new><port>
  (*## ## accept the key to the cache - if prompted)*
- # journalctl | grep sshd | grep listening

HoneyPi Monitoring

Configuring Your TCP/IP Address

- # ifconfig -a | more
- # cat /etc/network/interfaces
- Example Static IP Address (/24 net):
  auto eth0
  iface eth0 inet static
  address 192.168.10.95
  netmask 255.255.255.0
  gateway 192.168.10.1
  dns-nameservers 8.8.8.8
- # shutdown -r now

HoneyPi Monitoring

Configure: PenTBox 1.8

- PuTTY → user: root pass: <new> 2222
- # cd ~
- # pwd *(## should say /root)*
- # wget http://downloads.sourceforge.net/project/pentbox18realised/pentbox-1.8.tar.gz
- # tar -zxvf pentbox-1.8.tar.gz
- # cd pentbox-1.8

HoneyPi Monitoring

Running: PenTBox

- # ./pentbox.rb
- 2- Network tools
- 3- Honeypot
- 1- Fast Auto Configuration

- Your screen should have this output:
- HONEYPOT ACTIVATED ON PORT 80

HoneyPi Monitoring

Triggering the PenTBox Honeypot

- Open any browser on the PC
- In the URL bar type:
  http://192.168.136.<yourIP>
- Your browser should receive:
  Access denied
  IP Address login failed

  2020-02-13 15:29:16 -0600
- Close your browser window

HoneyPi Monitoring

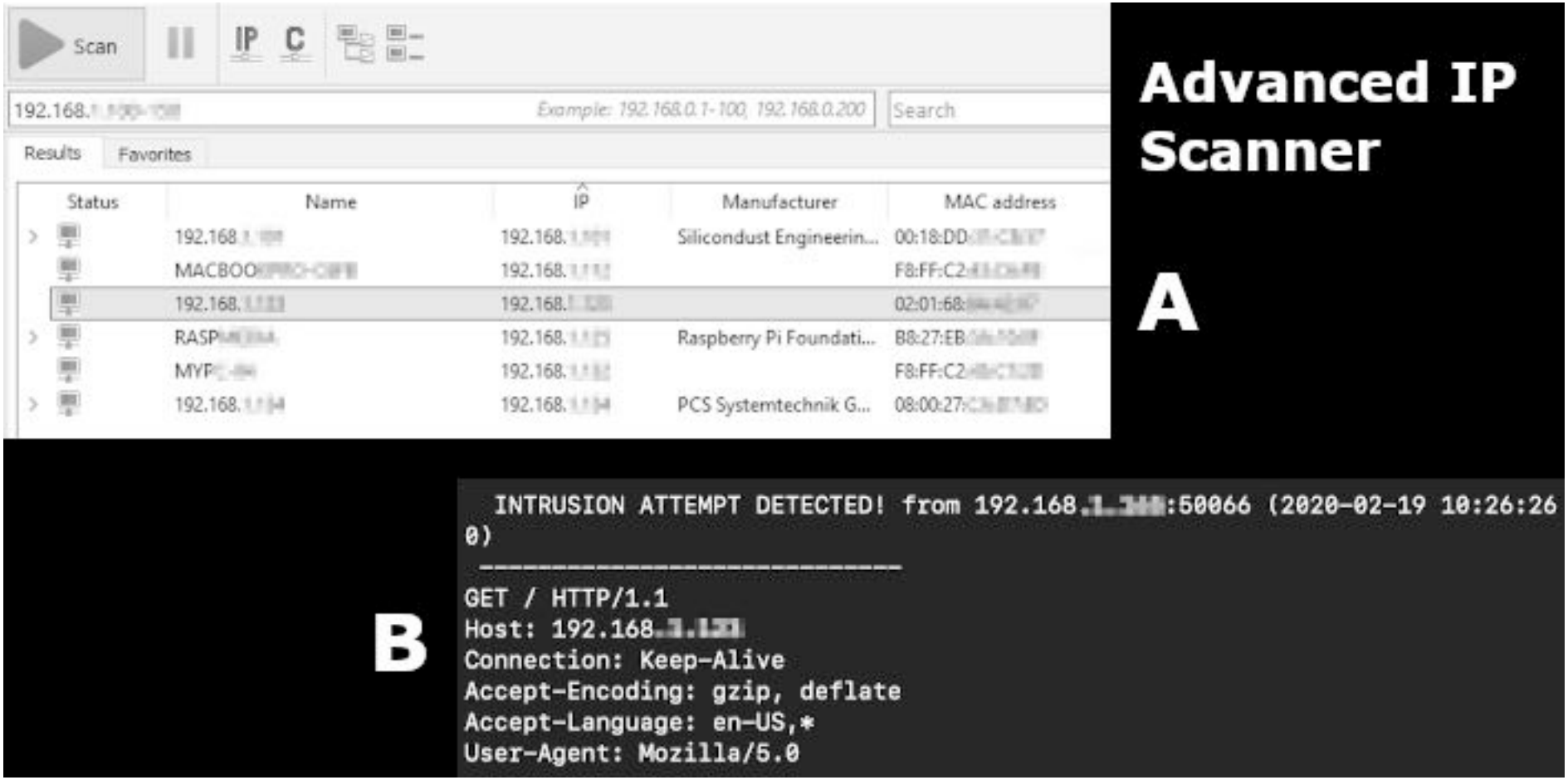
Results: PenTBox

- Now switch back to your PuTTY session
- INTRUSION ATTEMPT DETECTED! from 192.168.X.Y:59375 (2020-02-10 22:59:07 -0600)

  GET /favicon.ico HTTP/1.1
  Host: 192.168.136.<yourIP>
  User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:68.0) Gecko/20100101 Firefox/68.0

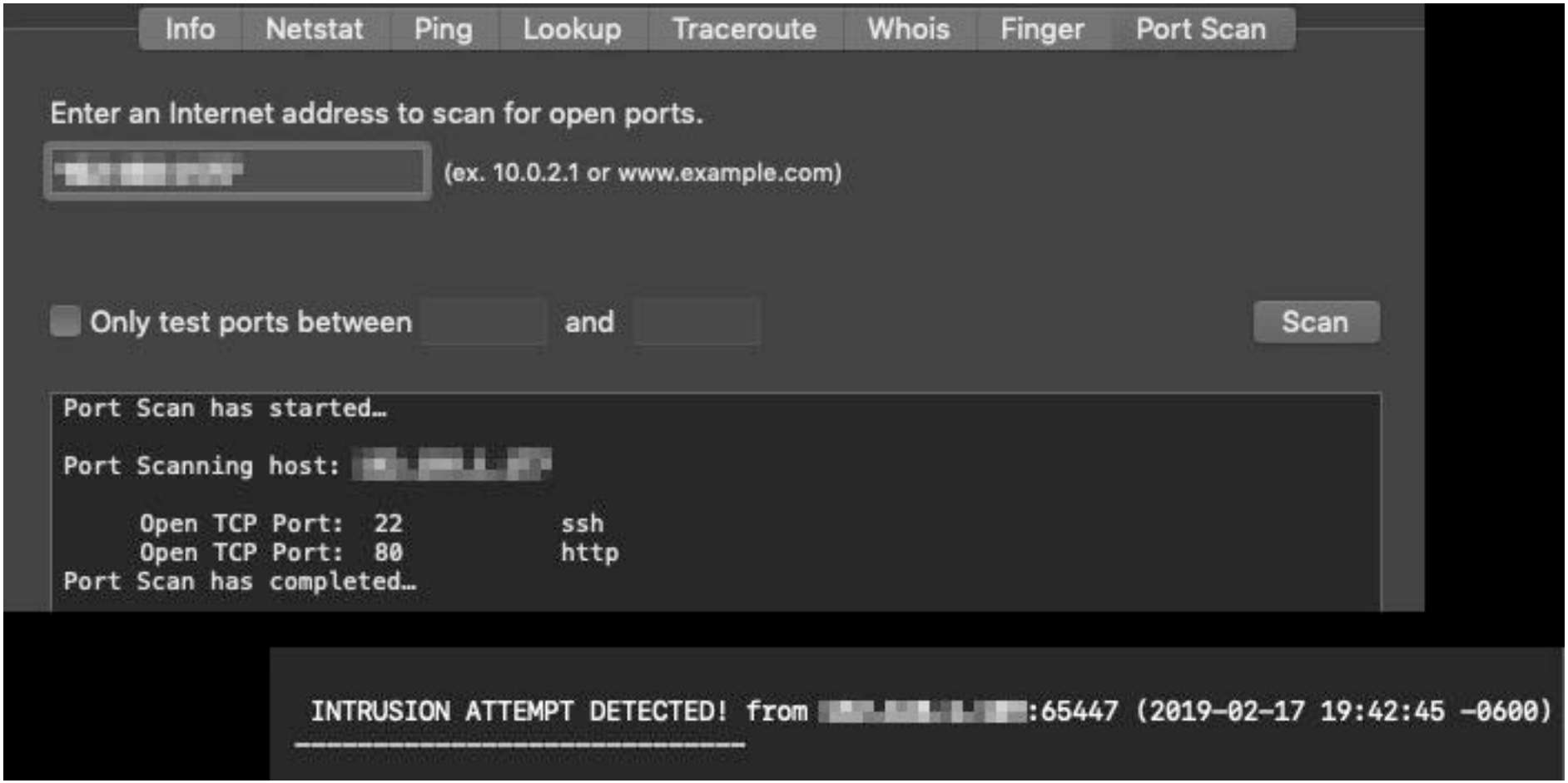
HoneyPi Monitoring

PenTBox in Action



HoneyPi Monitoring

Example Port Scan



HoneyPi Monitoring

Customizing PenTBox

- Cont + C to exit PenTBox
- ./pentbox.rb
- [Select] 2 / 3 / 2
- Port: 80
- False message: <h1>Server Maintenance</h1>
- Log: y
- /tmp/log_honeypot.txt
- Beep: n

HoneyPi Monitoring

Triggering the PenTBox Honeypot

- Open any browser on the PC
- In the URL bar type:
  http://192.168.136.<yourIP>
- Your browser should receive:
  Server Maintenance (## close browser)
- Open a second PuTTY screen
- PuTTY → user: root pass: <new> 2222
- tail /tmp/log_honeypot.txt

HoneyPi Monitoring

Results: PenTBox

- INTRUSION ATTEMPT DETECTED! from 192.168.X.Y:59375 (2020-02-10 22:59:07 -0600)

  GET /favicon.ico HTTP/1.1
  Host: 192.168.136.<yourIP>
  User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.15; rv:68.0)

HoneyPi Monitoring

Results: PenTBox

- # netstat -antp
  tcp 0.0.0.0:2222 LISTEN 593/sshd
  tcp 0.0.0.0:80 LISTEN 5053/ruby

- What other ports could we use?
- Cont + D to exit

- Switch to PenTBox SSH screen
- Cont + C to exit

HoneyPi Monitoring

Configure: Cowrie 2.0.0

- # cd ~
- # pwd *(## should say /root)*
- Run with a dedicated **non-root** user
- # adduser --disabled-password cowrie
  Full Name []:
  Room Number []:
  Work Phone []:
  Home Phone []:
  Other []:
- information correct? [Y/n] Y

HoneyPi Monitoring

Configure: Cowrie 2.0.0 – pg 2

- # su - cowrie
- $ id *(## should say uid=1002(cowrie))*
- $ pwd *(## /home/cowrie)*
- $ git clone http://github.com/cowrie/cowrie
- $ cd cowrie
- $ pwd *(## /home/cowrie/cowrie)*
- $ virtualenv --python=python3 cowrie-env
- $ source cowrie-env/bin/activate
- (cowrie-env) $ pip install --upgrade pip
- (cowrie-env) $ pip install --upgrade -r requirements.txt *(## time: ~ 6 min)*

HoneyPi Monitoring

Configure: Cowrie 2.0.0 – pg 3

- Open a second PuTTY screen
- PuTTY → user: root pass: <new> 2222
- # which authbind *(## /usr/bin/authbind)*
- # touch /etc/authbind/byport/22
- # chown cowrie:cowrie /etc/authbind/byport/22
- # chmod 770 /etc/authbind/byport/22
- # touch /etc/authbind/byport/23
- # chown cowrie:cowrie /etc/authbind/byport/23
- # chmod 770 /etc/authbind/byport/23
- Cont + D to log off

HoneyPi Monitoring

Configure: Cowrie 2.0.0 – pg 4

- Return to the Cowrie PuTTY screen
- $ cd etc/ *(## note: **no** leading "/")*
- $ pwd *(## /home/cowrie/cowrie/etc)*
- $ cp cowrie.cfg.dist cowrie.cfg
- $ cp userdb.example userdb.txt
- $ nano cowrie.cfg
- Cont + w (search) hostname =
- Change hostname = svr04 to FServer03
- Search: Enable SSH support
- ssh is enabled by default

HoneyPi Monitoring

Configure: Cowrie 2.0.0 – pg 5

- Search: tcp:2222 *(## arrow down)*
- listen_endpoints = tcp:2222:interface=0.0.0.0 to tcp:22
- Search: [telnet] *(## down to enabled = false)*
- change enabled = false to true
  (## arrow down to non-commented line)
- change listen_endpoints = tcp:2223:interface=0.0.0.0 to tcp:23
- Cont + X to exit and save cowrie.cfg

HoneyPi Monitoring

Running: Cowrie

- $ nano userdb.txt
  (## arrow down to root:x:)*
- change root:x:* to root:x:password
  (## ! will reject | !123456 invalid for scripts)
- Cont + X to exit and save userdb.txt
- $ cd ..
- $ bin/cowrie start
- netstat -antp
  tcp 0.0.0.0:22 LISTEN 4416/python3
  tcp 0.0.0.0:23 LISTEN 4416/python3

HoneyPi Monitoring

Triggering the Cowrie Honeypot

- Open a second PuTTY screen
- PuTTY → user: root pass: password 22
  (## note root@FServer03:~# (prompt))
- # df -h
- # cat /etc/passwd
- # wget https://commons.wikimedia.org/static/apple-touch/commons.png
- # ls -al
- # rm /etc/passwd
- Cont + D log off

HoneyPi Monitoring

Results: Cowrie

- Return to Cowrie SSH session
- $ cat var/log/cowrie/cowrie.log | grep userauth

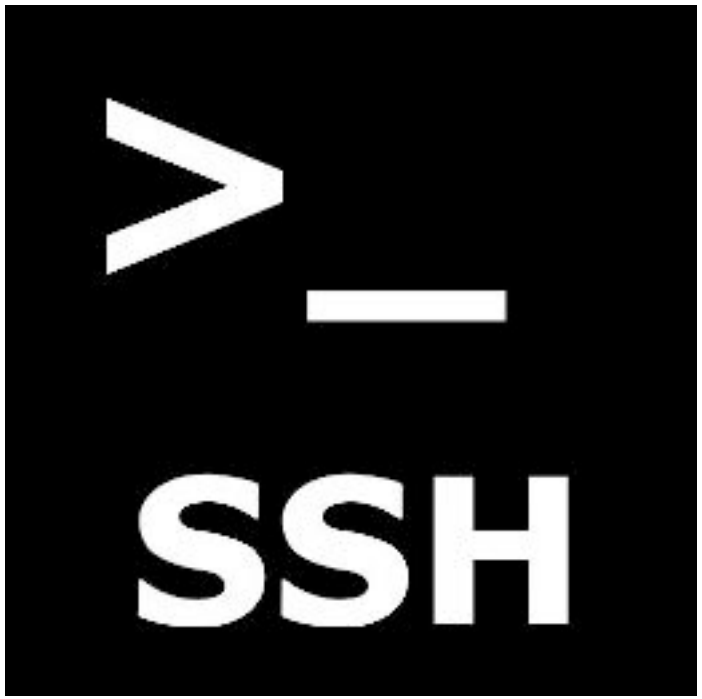
  (## 2020-02-20T02:59:15.355543Z [SSHService b'ssh-userauth' on HoneyPotSSHTransport,5,192.168.x.y] b'root' authenticated with b'password')

- ls -al var/lib/cowrie/downloads
- ls -al var/lib/cowrie/tty
  (## copy the filename for the next command)

- bin/playlog var/lib/cowrie/tty/<filename>

HoneyPi Monitoring

Cowrie, the Self-Healing Honeypot



- Open a second PuTTY screen
- PuTTY → user: root pass: password 22
  (## note root@FServer03:~# (prompt))
- # ls -al (## note: the downloaded file is gone!)
- # ls -al /etc/passwd (## the file is back!)
- Cont + D log off
- Return to Cowrie SSH session
- Cont + D log out *(## cowrie)* | Cont + D log off
- Close open windows and quit all running apps.
- Shutdown the NanoPi NEO2
  (## Menu→System→Shutdown→Confirm)

HoneyPi Monitoring

Common Honeypot Strategies

- Forward ports on routers to honeypots
- Set geoblocking (ex: Brazil, China, Iran, Russia, Thailand, Ukraine) rules
- Setup alerts for honeypot alarms
- Send all logs to a centralized server
- Good list: https://github.com/paralax/awesome-honeypots (open-source)
- Dionaea (HTTPS, MySQL, SMB)
- Allows IT department to become proactive on cyber security

HoneyPi Monitoring

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

HoneyPi Monitoring

Did Someone Mention a Surprise?!?

HoneyPi Monitoring

Questions?

Kevin Capwell
Midwest Educational Technology Association
kcapwell@brainstormk20.com

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