



Global Information Assurance Certification Paper

Copyright SANS Institute
Author Retains Full Rights

This paper is taken from the GIAC directory of certified professionals. Reposting is not permitted without express written permission.

Interested in learning more?

Check out the list of upcoming events offering
"Enterprise Penetration Testing (Security 560)"
at <http://www.giac.org/registration/gpen>

Remote Penetration Testing with Ninja Pi

GIAC (GPEN) Gold Certification

Author: Jeremy Druin, jeremy@ellipsisinfosec.com

Advisor: Rick Wanner

Accepted: September 27, 2020

Abstract

Remote penetration testing can have significant advantages over on-site tests but some types of testing require a physical presence. However, having testers on-premise may increase costs, duration, and difficulty. Penetration testing "drop boxes" can provide the physical connectivity needed while allowing the testing team to work off-site. These drop boxes can be built with readily available hardware such as Raspberry Pi. When paired with Kali Linux and a few helpful scripts, the drop box becomes a viable alternative to onsite testing for many use-cases. Such drop boxes are available for purchase, but a pen tester can build their own in less than a day that connects to a cloud server for maximum flexibility. These custom boxes are less expensive, offer the opportunity to learn new skills, can be customized to get around challenging connectivity issues, and built to fit specific use-cases.

Introduction

Some tasks in a security penetration test require a physical presence on the network. Some networks are logically segmented or physically isolated. In other cases, the type of testing mandates close proximity. For example, Wi-Fi testing typically requires the tester to be within approximately 125 to 235 feet (Mitchell, 2019). Network protocol analysis requires a network adapter connected to the network under study. Remote connections made via Network Address Translation (NAT) can frustrate techniques that require hosts to connect back to the test host (Corelan Team, 2014).

Traditionally, security penetration testers may have traveled to the customer's site. Once onsite, the tester can connect to the appropriate network. However, there are several drawbacks to testing on-premise. Travel increases overhead expense and engagement time without equivalent added value. Also, time away from home and offsite work may burden employees. Extraordinary events like COVID-19 make travel even more unpalatable (National Center for Immunization and Respiratory Diseases (NCIRD), Division of Viral Diseases, 2020).

In many cases, the penetration tester can have the best of both worlds: perform the test remotely while having a physical presence on the customer network. A penetration testing platform deployed to the customer network and accessible by a remote penetration tester could solve this issue (Tomaschik, Raspberry Pi as a Penetration Testing Implant (Dropbox), 2020). Once the "drop box" is placed where needed, the penetration tester can login remotely; potentially from anywhere. There are other benefits as well. The penetration tester can learn new skills in building the rig themselves and increase the flexibility of their service offerings by having the additional options provided by one or more drop boxes.

The pen test equipment needed depends on the type of pen test. Flexibility is important. A drop box must meet several requirements.

- Reasonably inexpensive since the unit may be damaged or lost
- Reliable. Issues during an engagement would be highly disruptive
- Easy to acquire and build

- The software should have a license model that does not frustrate efforts
- The operating system should be designed for penetration testing and let the pen tester install additional software on the fly
- Flexible network connectivity. The pen tester may connect directly to the Raspberry Pi. Alternatively, the drop box might connect out to an intermediate server shared by the penetration tester.
- Powerful enough to run resource intensive processes like vulnerability scanning

Multiple platforms have the potential to meet these requirements (Heath, 2019). One such platform is Raspberry Pi 4 B.

1. Hardware

Hardware is available separately or in kits. The drop box should have the following components.

1.1. Raspberry Pi Board

The Raspberry Pi 4 provides a solid base. The 64-bit processor, optional 8 GB of Random-Access Memory (RAM), and on-board Wi-Fi radio (Raspberry Pi, n.d.) can run Kali 64-bit OS (Kali Linux, n.d.), support wireless penetration testing natively, and run vulnerability scanners (Tenable, Inc., n.d.). The Pi also enjoys significant community support as a penetration testing platform (@dievus, 2020) (Tomaschik, Raspberry Pi as a Penetration Testing Implant (Dropbox), 2020) (EC-Council, 2020) and has several other features that make it a flexible board. Alternatives to the Raspberry PI 4 include ODroid and BeagleBone.

1.2. Case

The additional capabilities of the Raspberry Pi 4 over predecessors generates more heat (Anderson, 2019). An aluminum case helps the Pi run at lower temperatures (Halfacree, Group test: Best Raspberry Pi 4 thermal cases tested and ranked, 2020). Besides temperature control and ventilation, the best cases for penetration tests are inconspicuous so they are less likely to be disturbed during testing. The stark black

aluminum case (Vilros, n.d.) from Vilros was tested in a lab and in the field with good results.

1.3. Power Supply

The Raspberry Pi 4 requires a 3 Amp power supply¹ (Raspberry Pi, n.d.) or Power over Ethernet (PoE). Unlike previous models that use micro-USB, the Pi 4 uses USB-C style power input.

1.4. SD Cards

The read-write speed of SD Cards is measured by class. Classes 2,4,6 and 10 are progressively faster. There are also even faster Ultra-High Speed (UHS) cards rated U1 or U3. The respective symbols look like the letter "U" with a 1 or 3 inside (Carr, 2018).



Figure 1: Symbols for SD Card speed ratings (Hoffman, 2017)

Ultra-High Speed (UHS) cards performed well in testing with the Pi 4. However, testing showed some brands to be more reliable than others. While there are almost certainly other cards that will work as well, the Samsung EVO Select (Amazon.com, n.d.) performed best. Speeds were adequate with no failures in the lab or field. The 64 GB size has a balance of low-price and high-capacity but a 128 GB card was less than \$19.00 at the time of this writing (Amazon.com, n.d.).

Two cards should be purchased; one 64 - 128 GB card for Kali and a card for Raspberry Pi OS² (Raspberry Pi, n.d.). Raspberry Pi OS is only needed to update the EEPROM. No other operating system should be used to perform the update (Chambers, 2020). The SD card for Raspberry PI OS can be smaller. The smallest Samsung EVO

¹ The R-Pi4 can run on 2.5 Amps as long as no peripherals are plugged into the USB or GPIO header (Raspberry Pi, n.d.)

² Formerly Raspbian

Select was 32 GB at the time of this writing. The easiest way to maintain both operating systems is to install each OS on separate SD cards.

1.5. Cooling

The Pi 4 will experience reduced performance via "thermal throttling" if the heat generated by the system is not kept in check (Bate, 2019). A combination of a firmware update, case selection, heat sinks, and an onboard fan can control heat (Cook, 2019).

The Raspberry Pi Foundation has released a firmware update (Raspberry Pi, n.d.) that lowers the amount of electricity used and heat generated by the Pi 4 (Halfacree, 2019). The update is installed by following the EEPROM update process³. Adding heat sinks to the System on Chip (SOC) and Random-Access Memory (RAM) chip further dissipates heat (Cook, 2019). Heat sinks are even more effective when combined with aluminum case equipped with a fan (Wu, 2019).

1.6. Kits

Two kits that included all recommended components except the SD card were tested.

1. Vilros Raspberry Pi 4 Basic Starter Kit (4GB, Black)
 - a. Raspberry Pi 4 Model B 4 GB Ram
 - b. Aluminum Alloy Case
 - c. 2 Heat Sinks and Fan
 - d. 3A USB-C Power Supply with On/Off Switch
2. CanaKit Raspberry Pi 4 4GB Starter Kit (4GB, Black)
 - a. Raspberry Pi 4 Model B 4 GB Ram
 - b. Plastic Case
 - c. 2 Heat Sinks and Fan
 - d. 3.5A USB-C Power Supply with On/Off Switch

Both kits were tested with the Samsung EVO Select 64 GB "Green" (Amazon.com, n.d.) SD cards. Each kit performed well and several factors that affect the buying decision are personal preferences. However, the Vilros kit was chosen because the

³ [How to Update Bootloader EEPROM - Raspberry Pi](#) -

<https://www.youtube.com/watch?v=u6EUplRPF5s>

kit was reasonably priced at \$85.00⁴, came with a nondescript aluminum case and minimized the number of unneeded accessories.

1.7. Optional Accessories

Some types of testing may require additional components. Two USB ports support Wireless, Bluetooth and other adapters. The radio onboard the Pi 4 worked well for basic Wi-Fi monitoring. An external Wi-Fi adapter may provide better range since there will be less interference from the case, the external adapter can use a more powerful radio and use an external antenna (Hackers Grid, n.d.).

The onboard radio works well for basic Wi-Fi monitoring. The *iwconfig* command lists the wireless interface (*Figure 2*).

```
kali@ninja-pi-2:~$ iwconfig
lo                no wireless extensions.

wlan0             IEEE 802.11  ESSID:off/any
                  Mode:Managed  Access Point: Not-Associated  Tx-Power=31 dBm
                  Retry short limit:7   RTS thr:off   Fragment thr:off
                  Power Management:on

leth0             no wireless extensions.
```

Figure 2: Listing the wireless interface

The interface is compatible with the Aircrack-NG suite. Basic Wi-Fi monitoring was tested by placing the interface into monitor mode using the *airmon-ng* tool. Command *sudo airmon-ng check* finds running processes that may interfere (*Figure 3*). These processes are stopped with *sudo airmon-ng check* (*Figure 4*). The interface is placed into monitor mode with *sudo airmon-ng check* (*Figure 5*). The Airodump-NG tool uses the wlan0mon interface to list local Wi-Fi networks with command *sudo airodump-ng wlan0mon* (*Figure 6*).

⁴ July 10, 2020

```
kali@ninja-pi-2:~$ sudo airmon-ng check
[sudo] password for kali:
```

```
Found 3 processes that could cause trouble.
Kill them using 'airmon-ng check kill' before putting
the card in monitor mode, they will interfere by changing channels
and sometimes putting the interface back in managed mode
```

```

PID Name
327 dhclient
355 NetworkManager
[ 435 wpa_supplicant
```

Figure 3: By default, Kali runs processes that can interfere with the operation of Aircrack-NG tools

```
kali@ninja-pi-2:~$ sudo airmon-ng check kill
```

```
Killing these processes:
```

```

PID Name
327 dhclient
[ 435 wpa_supplicant
```

Figure 4: airmon-ng can stop processes that interfere with Aircrack-NG tools

```
kali@ninja-pi-2:~$ sudo airmon-ng start wlan0
```

```

PHY      Interface      Driver      Chipset
phy0     wlan0          brcmfmac   Broadcom 43430

(mac80211 monitor mode vif enabled for [phy0]wlan0 on [phy0]wlan0mon)
command failed: Unknown error 524 (-524)
[ (mac80211 station mode vif disabled for [phy0]wlan0)
```

Figure 5: The wlan0 interface is placed into monitor mode despite a warning message

```
kali@ninja-pi-2:~$ sudo airodump-ng wlan0mon
Warning: Detected you are using a non-UNICODE terminal character encoding.
```

```
CH 9 ][ Elapsed: 0 s ][ 2020-08-21 00:17

BSSID           PWR Beacons  #Data, #/s  CH  MB  ENC CIPHER AUTH ESSID
1E:EC:DA [REDACTED] -73      2         0  0   1  54  WPA2 CCMP PSK [REDACTED] NetN
3A:E8:29 [REDACTED] -46      2         0  0   6  54  WPA2 CCMP PSK [REDACTED] NetN
18:E8:29 [REDACTED] -46      3         0  0   6  54  WPA2 CCMP PSK [REDACTED] NetN
2A:E8:29 [REDACTED] -49      2         2  0   6  54  WPA2 CCMP PSK [REDACTED] NetN
1A:E8:29 [REDACTED] -46      3         0  0   6  54  WPA2 CCMP PSK [REDACTED] NetN
92:2A:A8 [REDACTED] -60      2         0  0  11  54  WPA2 CCMP PSK [REDACTED] NetN
82:2A:A8 [REDACTED] -60      2         0  0  11  54  WPA2 CCMP PSK [REDACTED] NetN
```

Figure 6: The Pi 4 wireless interface monitoring nearby Wi-Fi networks

2. Operating System

2.1. Raspberry Pi OS

As mentioned, Raspberry Pi OS is only needed to update the EEPROM. No other operating system should be used to perform the update (Chambers, 2020). Since SD Cards are relatively inexpensive, install Raspberry Pi OS on a separate SD Card from Kali Linux to avoid complications. Raspberry Pi OS can be set up⁵ with Raspberry Pi Imager (Raspberry Pi Foundation, n.d.).

2.1.1. Updating Raspberry Pi Firmware

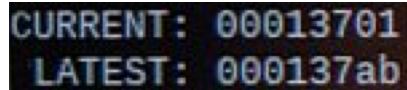
Before using the Pi 4 for penetration testing, update the EEPROM⁶ as needed. Boot the Pi 4 to Raspberry Pi OS then check for available updates with command `sudo rpi-eeprom-update`. If an update is available, the output shows the latest version is higher than the current version (Figure 7). Install the update with `sudo rpi-eeprom-update -a`.

⁵ [How to Install Raspberry OS on Raspberry Pi 4 -](https://www.youtube.com/watch?v=v_myQfxe534)

https://www.youtube.com/watch?v=v_myQfxe534

⁶ [How to Update Bootloader EEPROM - Raspberry Pi -](https://www.youtube.com/watch?v=u6EUplRPF5s)

<https://www.youtube.com/watch?v=u6EUplRPF5s>



```
CURRENT: 00013701
LATEST: 000137ab
```

Figure 7: An update is available

2.2. Kali Linux

Kali Linux is a good choice in a penetration testing operating system. Kali has the most common tools installed and configured (OffSec Services Limited, n.d.). Offensive Security updates packages frequently. Also, Offensive Security directly supports Kali on Pi (OffSec Services Limited, n.d.). Kali Linux 64-bit for Pi can be downloaded⁷ from Offensive Security and installed⁸ by copying the image to the SD card (Druin, 2020).

3. Configuration

3.1. Key-based authentication

Kali is best accessed remotely through Secure Shell (SSH) because SSH provides access to a secure command shell; the best method of interacting with Kali OS. SSH supports multiple types of authentication including passwords and key-based authentication. The Kali Pi image uses password-based authentication by default⁹, but public-key authentication is more secure (SSH Communications Security, Inc.).

With public-key authentication, the user authenticates with a *private-key* which is much harder to guess than an ordinary password. As long as the user passes the "-a" parameter when creating the key, the private-key is encrypted with the owner's password to protect the key sitting in the key file. Unlike a password, neither the private-key nor the key encryption password is transmitted over the network during login (Ylonen, 2006).

Private-keys have a corresponding public-key. The public- and private-key are related so that operations performed with one can only be undone with the other. The public-key is placed on any server that the private-key should authenticate into. This allows the server to challenge the client who is trying to authenticate without requiring

⁷ <https://images.kali.org/arm-images/kali-linux-2020.3-rpi3-nexmon-64.img.xz>

⁸ [How to install Kali Linux on Raspberry Pi 4 -](#)

<https://www.youtube.com/watch?v=xQofad1fTz8>

⁹ Username and password are "kali" (OffSec Services Limited)

the client to share the private key. Instead, the server encrypts a random secret with the public-key then insists the client use its private-key to decrypt that secret. If the client is able, the server knows the client has the private-key.

The key pair are created with the *ssh-keygen* tool using command *ssh-keygen -t <algorithm> -a <rounds key encryption> -f <name of key>*. This command is run on the client. Assuming we are setting up the first Ninja Pi, the key is created with the following command¹⁰.

```
ssh-keygen -t ed25519 -a 125 -f ./kali.ninja-pi-1.id_ed25519
```

The public-key goes onto the system(s) that the user will log into with the private-key in the user's *authorized_keys* file (SSH.com, n.d.). The *ssh-copy-id* utility moves the public-key from the client to the server placing the public-key in the user's *authorized_keys* file. The *authorized_keys* file is located within the *home* directory in the *.ssh* folder. The public-key is moved to Kali with *ssh-copy-id -i <name of key> <username>@<IP address of Kali>*.

```
ssh-copy-id -i kali.ninja-pi-1.id_ed25519 kali@192.168.1.176
```

Key-based authentication may not be enabled by default. To turn on key-based authentication, login to Kali once more and enable private-key authentication with this command which removes the comment symbol "#" from the appropriate line in the */etc/ssh/sshd_config* configuration file.

```
sudo sed -i 's/#PubkeyAuthentication yes/PubkeyAuthentication yes/g' /etc/ssh/sshd_config
```

Now back on the client host, the user can authenticate to Kali with the private-key.

¹⁰ [How to Set Up SSH Key Authentication](#) -

<https://www.youtube.com/watch?v=mZSso17Hal8>

```
ssh -i kali.ninja-pi-1.id_ed25519 kali@192.168.1.176
```

3.2. Enforcing key-based authentication

Once key-based authentication is working well, password-based authentication can be disabled. The following commands disable root login, disable password-based authentication, and restart the SSH service. The commands change the value of the *PermitRootLogin* and *PasswordAuthentication* directives from "yes" to "no", then restart the SSH service to activate the new configuration.

```
sudo sed -i 's/PermitRootLogin yes/PermitRootLogin no/g'
/etc/ssh/sshd_config
sudo sed -i 's/#PasswordAuthentication yes/PasswordAuthentication
no/g' /etc/ssh/sshd_config
sudo service sshd restart
```

3.3. Connecting to the Ninja Pi

The penetration tester can connect to the Ninja Pi directly over SSH since the customer's firewall will block the SSH service (*Figure 8*). This option requires the penetration tester to visit the customer network or have the customer provide a VPN connection. Also, the customer must tell the penetration tester the IP address of the Pi and allow the connection through firewalls. This option works well with the private-key authentication discussed earlier.

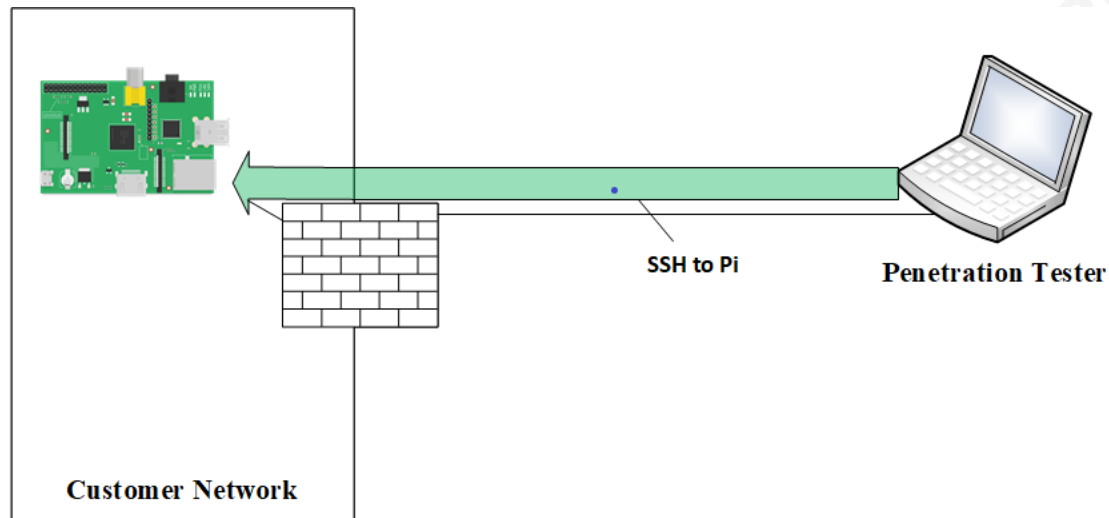


Figure 8: Direct connection to Ninja Pi over SSH

SSH supports reverse connections that allow the server to create an SSH tunnel to the client. From the client's point of view, this tunnel acts as a direct VPN connection. The client connects back to the server through this tunnel.

Using a reverse SSH connection from the Pi back to the penetration tester avoids the need to know the Pi's IP address, the configuration of a VPN, and ingress firewall rules. However, the penetration tester's laptop is most likely behind a router using Network Address Translation (NAT). A direct connection is not practical (Tyson, n.d.). Port forwarding can allow the reverse connection through NAT but does not gracefully support teams of penetration testers working together, a penetration tester working from various locations, or a pen test utilizing multiple Ninja Pi.

An intermediate server solves the remaining issues with the reverse connection (*Figure 9*).

1. The Pi connects to an *intermediate server* (*Figure 9*) that has an IP address on the Internet.
2. The penetration tester(s) log into the intermediate server. Once authenticated, they find the reverse tunnel waiting.
3. The penetration testers open an SSH shell from the intermediate server to the Ninja Pi(s). The remote testers can test from the Pi physically connected to the target network.

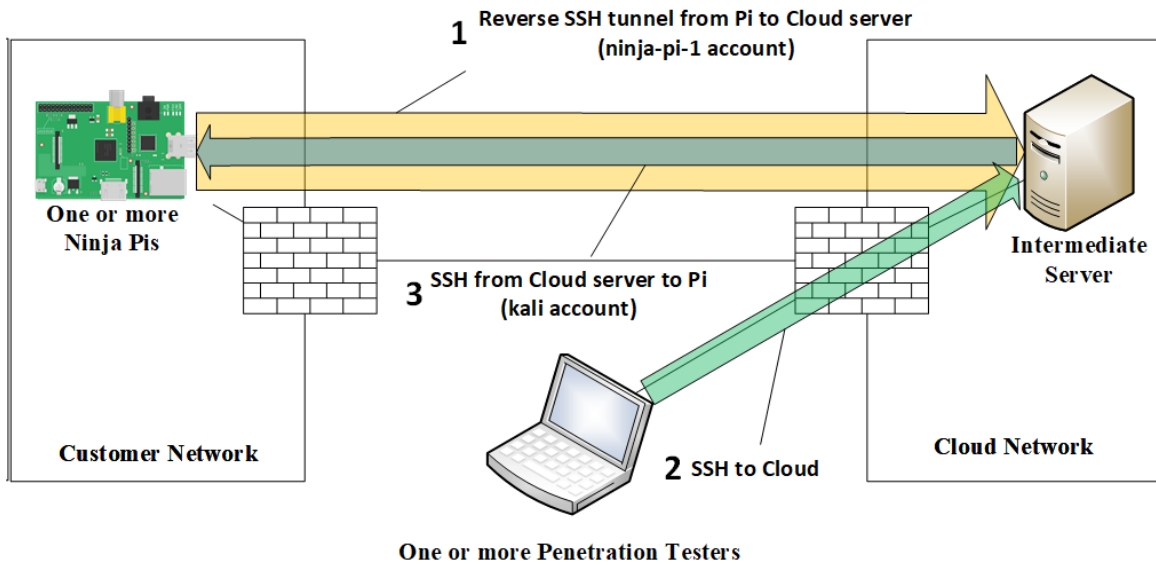


Figure 9: An intermediate server marshals the connection between Ninja Pis and the penetration testing team

3.4. Configuring the tunneled connection

A cloud rental acts as our intermediate server (*Figure 9*). Two accounts are required.

1. An account for the Ninja Pi to authenticate to the cloud to set up the reverse tunnel (*Figure 9 - yellow arrow*)
2. Another allowing the penetration tester sitting on the cloud to authenticate to Ninja Pi (*Figure 9 - blue arrow inside yellow arrow*)

3.5. Connecting Ninja Pi to the cloud server

Some of the following commands are run on the cloud server and others on the Ninja Pi.

- **Red:** Cloud server
- **Blue:** Ninja Pi

On the *cloud server*, these commands create an account for the Ninja Pi to authenticate to the cloud server and set the password.

```
sudo /usr/sbin/useradd -c "Ninja Pi 1 Cloud Account" -m -s
/bin/bash ninja-pi-1
```

```
sudo /usr/bin/passwd ninja-pi-1
```

In a bit of foreshadowing, switch to the *ninja-pi-1* account then create the authentication keys that will allow the *ninja-pi-1* account on the cloud server to log into Ninja Pi 1 device.

```
su -l ninja-pi-1
```

```
ssh-keygen -t ed25519 -a 125 -f kali.ninja-pi-1.id_ed25519
```

The cloud server cannot log into Ninja Pi yet, therefore ssh-copy-id cannot be used to move the cloud account public-key. The public-key created on the cloud server has to be manually copied into the *authorized_keys* file on the Ninja Pi. Copy the key to the *authorized_keys* file for the *kali* account on Ninja Pi. On the *cloud server*, print the public-key.

```
cat /home/ninja-pi-1/.ssh/kali.ninja-pi-1.id_ed25519.pub
```

On the *Ninja Pi*, add this key into the *authorized_keys* file for the *kali* account.

```
echo <value of public key> >> /home/kali/.ssh/authorized_keys
```

A second set of keys is needed to allow the Ninja Pi to log into the *ninja-pi-1* account on the cloud server. While still on the Ninja Pi, create an SSH key pair so the Ninja Pi can log into the *ninja-pi-1* account on the cloud server, then print the public-key.

```
ssh-keygen -t ed25519 -a 125 -f ninja-pi-1.cloud.id_ed25519
```

```
cat /home/kali/.ssh/ninja-pi-1.cloud.id_ed25519.pub
```

On the cloud server, add the public-key to the *authorized_keys* file for the cloud server account *ninja-pi-1*.

```
echo <value of public key> >> /home/ninja-pi-1/.ssh/authorized_keys
```

At this point, keys have been created so that the Ninja Pi can start the SSH reverse tunnel to the cloud server. The reverse tunnel lets the pen tester on the cloud server log into the Ninja Pi. The user *kali* on the Ninja Pi can log into the cloud account *ninja-pi-1* to establish the reverse SSH tunnel (*Figure 9 - Flow 1*). Afterward, the cloud server can leverage the tunnel to log into the Ninja Pi using the *kali* account (*Figure 9 - Flow 3*).

3.6. Automating the reverse tunnel

When using the reverse tunnel connection method, the Ninja Pi has to connect to the cloud automatically. Also, the Ninja Pi needs a more secure and convenient configuration. The following commands are run on the Ninja Pi to change the *kali* user password, set the hostname on the Ninja Pi, and tell the Ninja Pi the hostname of the cloud server.

```
#Change password for the kali account. At this point, the default
password is "kali"
echo kali:<password> | sudo chpasswd -s 50000 -c SHA512
# Set the Ninja Pi hostname to ninja-pi-1
sudo echo "ninja-pi-1" > /etc/hostname
# Overwrite the Ninja Pi hostname in /etc/hosts to ninja-pi-1
sudo sed -i 's/kali/ninja-pi-1/g' /etc/hosts
#Add the cloud server hostname ninja-master in /etc/hosts
sudo echo >> /etc/hosts
sudo echo -e "\<cloud server IP address>\tninja-master" >>
/etc/hosts
```

Kali allows the user to create scheduled jobs to kick off scripts at a time chosen by the user. The following script *connect-to-cloud.sh* automates the connection from Ninja Pi to the cloud server. Notice the script checks if the Ninja Pi is connected to the cloud before setting up the reverse SSH tunnel.

```
#Create file connect-to-cloud.sh to automate connection to Ninja
Master cloud server
cat << EOF > connect-to-cloud.sh
#!/bin/bash

USERNAME="ninja-pi-1"
PRIVATE_KEY="/home/kali/.ssh/ninja-pi-1.cloud.id_ed25519"
LOCAL_PORT="22"
REMOTE_PORT="2222"
```

```

CLOUD_SERVER_NAME="ninja-master"

if [[ \$(ps -ef | grep -c "/usr/bin/ssh -i \${PRIVATE_KEY} -nNT -R
\${REMOTE_PORT}:localhost:\${LOCAL_PORT}
\${USERNAME@\${CLOUD_SERVER_NAME}}) -eq 1 ]]; then

    logger -t \${0} "Connecting to \${CLOUD_SERVER_NAME} as
\${USERNAME} mapping remote port \${REMOTE_PORT} back to here on
local port \${LOCAL_PORT}"

    logger -t \${0} "USERNAME: \${USERNAME}"
    logger -t \${0} "PRIVATE_KEY: \${PRIVATE_KEY}"
    logger -t \${0} "LOCAL_PORT: \${LOCAL_PORT}"
    logger -t \${0} "REMOTE_PORT: \${REMOTE_PORT}"
    logger -t \${0} "CLOUD_SERVER_NAME: \${CLOUD_SERVER_NAME}"

    /usr/bin/ssh -i \${PRIVATE_KEY} -nNT -R
\${REMOTE_PORT}:localhost:\${LOCAL_PORT}
\${USERNAME@\${CLOUD_SERVER_NAME}} -t \${0}
    if [[\${?} -eq 0]]; then
        logger -t \${0} "SSH connection succeeded"
    else
        logger -t \${0} "Connection did not succeed. Will
try again in a minute. SSH connection status code: \${?}"
    fi
else
    logger -t \${0} "Already connected to \${CLOUD_SERVER_NAME}
as \${USERNAME}"
fi
EOF

# Make the connect to cloud script executable

```

```
chmod u+x /home/kali/connect-to-cloud.sh
```

With the *connect-to-cloud.sh* created, the scheduled task is implemented using the Linux *cron* system¹¹. This script will be run every minute.

```
# Allow the kali user to run cron jobs
sudo echo kali > /etc/crontab.allow

# Add a cron job to cron scheduler
sudo echo -e "* * * * *\tkali\t/bin/bash /home/kali/connect-to-
cloud.sh" >> /etc/crontab
```

3.7. Updating Kali OS and installed software

The Advanced Package Toolkit (APT) can keep both the Kali OS and installed software up-to-date (OffSec Services Limited). When an update is requested, APT compares the software installed on the system against a list of available packages then downloads and installs available updates (Debian, 2013).

To update the list of available packages, run command *sudo apt-get update*. Afterward, APT can be instructed to update the OS and installed software with *sudo apt-get dist-upgrade -y*. Once updates are complete, the *sudo apt-get autoremove* and *sudo apt-get clean -y* commands can uninstall unneeded software and remove the installation packages to save space. The following script creates a file, *update-kali.sh*, to automate this process.

```
#Create file update-kali.sh to update Kali OS
cat << EOF > ~/update-kali.sh
#!/bin/bash

sudo apt-get update
sudo apt-get dist-upgrade -y
sudo apt-get autoremove
```

¹¹ <https://opensource.com/article/17/11/how-use-cron-linux>

```
sudo apt-get clean -y
EOF

# Make the update kali script executable
chmod u+x ~/update-kali.sh
```

The file is created in the *kali* user's "home" directory¹². Run the *update-kali.sh* script to update software and operating system.

```
~/update-kali.sh
```

4. Conclusion

Remote penetration testing offers advantages for customers and penetration testers alike, but some penetration testing tasks require a physical presence on the network. A Ninja Pi drop box fills this gap by providing the penetration tester remote access to the customer network.

A few different platforms offer the power and reliability needed in a small, lightweight platform. One such system is the Raspberry Pi 4. The Pi can be partnered with heat-dissipating components, the Kali Linux operating system, and private-key authentication to create a secure, dependable penetration testing system.

The customer may allow direct connection to the Pi through a VPN. If not, a reverse SSH connection to intermediate cloud servers provides a flexible alternative. Multiple penetration testing team members can utilize multiple Ninja Pis over secure SSH tunnels. Short scripts maintain the reverse SSH tunnels even when disconnected.

These Ninja Pis offer the best of both worlds. The boxes are an on-site testing platform that supports many different types of security tests while letting the penetration testing team operate remotely.

¹² /home/kali/update-kali.sh

Appendix A: Video Tutorials

How to Install Raspberry OS on Raspberry Pi 4	https://www.youtube.com/watch?v=v_myQfxe534
How to install Kali Linux on Raspberry Pi 4	https://www.youtube.com/watch?v=xQofad1fTz8
How to Update Bootloader EEPROM - Raspberry Pi	https://www.youtube.com/watch?v=u6EUplRPF5s
How to Set Up SSH Key Authentication	https://www.youtube.com/watch?v=mZSso17Ha18

References

- @dievus. (2020, 1 8). *Pentesting with a Raspberry Pi*. Retrieved from The Ethical Hacker Network: <https://www.ethicalhacker.net/community/pentesting-with-a-raspberry-pi/>
- Aircrack-NG. (n.d.). *Aircrack-NG Home Page*. Retrieved 8 21, 2020, from Aircrack-NG: <http://aircrack-ng.org>
- Amazon.com. (n.d.). *CanaKit Raspberry Pi 4 4GB Starter Kit - 4GB RAM*. Retrieved 8 20, 2020, from Amazon.com: https://www.amazon.com/CanaKit-Raspberry-4GB-Starter-Kit/dp/B07V5JTMV9/ref=sr_1_1?crid=3MR55JOS7LKZK&dchild=1&keywords=canakit+raspberry+pi+4&qid=1597966729&s=electronics&prefix=canakit+%2Celectronics%2C298&sr=1-1
- Amazon.com. (n.d.). *Samsung EVO Select 128 GB*. Retrieved 8 17, 2020, from Amazon.com: https://smile.amazon.com/Samsung-Select-microSDXC-Adapter-MB-ME64HA/dp/B0887GP791/ref=sr_1_16?dchild=1&keywords=Samsung%2BEVO%2BSelect%2B64GB%2BmicroSDXC%2BUHS-I%2BU1%2B100MB%2Fs%2BFull%2BHD&qid=1597714094&s=electronics&sr=1-16&th=1
- Amazon.com. (n.d.). *Samsung EVO Select 64 GB*. Retrieved 8 17, 2020, from Amazon.com: https://smile.amazon.com/Samsung-Select-microSDXC-Adapter-MB-ME64HA/dp/B08879MG33/ref=sr_1_16?dchild=1&keywords=Samsung+EVO+Select+64GB+microSDXC+UHS-I+U1+100MB%2Fs+Full+HD&qid=1597714094&s=electronics&sr=1-16
- Amazon.com. (n.d.). *Vilros Raspberry Pi 4 Basic Starter Kit with Fan-Cooled Heavy-Duty Aluminum Alloy Case (4GB, Black)*. Retrieved 8 20, 2020, from Amazon.com: https://www.amazon.com/Vilros-Raspberry-Fan-Cooled-Heavy-Duty-Aluminum/dp/B07XTRK8D4/ref=sxsts_sxwds-bia-wc-p13n1_0?cv_ct_cx=pi&dchild=1&keywords=pi&pd_rd_i=B07XTRK8D4&pd_rd_r=e03ad84e-dfec-4674-aa78-dbf5c6c0d5c7&pd_rd_w=l7cJy&pd_rd_wg=4tyGg&pf_rd_p=13bf9b
- Anderson, T. (2019, 7 22). *Too hot to handle? Raspberry Pi 4 fans left wondering if kit should come with a heatsink*. Retrieved from The Register: https://www.theregister.com/2019/07/22/raspberry_pi_4_too_hot_to_handle/
- Bate, A. (2019, 11 28). *Thermal testing Raspberry Pi 4*. Retrieved from Raspberry Pi Blog: <https://www.raspberrypi.org/blog/thermal-testing-raspberry-pi-4/>
- BeagleBoard. (n.d.). *Getting Started*. Retrieved 8 16, 2020, from BeagleBoard: <https://beagleboard.org/getting-started>
- Bullock, B. (2016, 8 2). *How to Build Your Own Penetration Testing Drop Box*. Retrieved from Black Hills Information Security: <https://www.blackhillsinfosec.com/how-to-build-your-own-penetration-testing-drop-box/>

- Cannakit. (n.d.). *Raspberry Pi 4 Kits*. Retrieved 8 2020, from Cannakit:
<https://www.cannakit.com/raspberry-pi/pi-4-kits>
- Carr, D. (2018, 7 6). *UNDERSTANDING SD CARD NAMING, SPEEDS AND SYMBOLS*. Retrieved from Shutter Muse: <https://shuttermuse.com/understanding-sd-card-naming-speeds-symbols/>
- Chambers, J. A. (2020, 7 28). *Raspberry Pi 4 Bootloader Firmware Updating / Recovery Guide*. Retrieved from James A Chambers:
<https://jamesachambers.com/raspberry-pi-4-bootloader-firmware-updating-recovery-guide/>
- Cook, J. S. (2019, 12 10). *Raspberry Pi 4 Cooling Solutions Comparison*. Retrieved from Arrow: <https://www.arrow.com/en/research-and-events/articles/raspberry-pi-4-cooling-solutions-comparison>
- Corelan Team. (2014, 1 4). *Metasploit Meterpreter and NAT*. Retrieved from Corelan Team: <https://www.corelan.be/index.php/2014/01/04/metasploit-meterpreter-and-nat/>
- Debian. (2013, 11 25). *apt - command-line interface*. Retrieved 8 21, 2020, from Manpages: <https://manpages.debian.org/jessie/apt/apt.8.en.html>
- Druin, J. (2020, 7 18). *How to install Kali Linux on Raspberry Pi 4*. Retrieved from YouTube: <https://www.youtube.com/watch?v=xQofad1fTz8>
- EC-Council. (2020, 6 4). *HOW TO USE KALI LINUX & RASPBERRY PI FOR WIRELESS PENETRATION TESTING*. Retrieved from EC-Council Blog:
<https://blog.eccouncil.org/how-to-use-kali-linux-raspberry-pi-for-wireless-penetration-testing/>
- F.Sikos, L. (2020, 3). Packet analysis for network forensics: A comprehensive survey. *Forensic Science International: Digital Investigation, 200892, 32*. Retrieved from
<https://www.sciencedirect.com/science/article/pii/S1742287619302002>
- Hackers Grid. (n.d.). *Ultra Long Range WiFi Receiver at Home*. Retrieved 8 21, 2020, from Hackers Grid: <https://hackersgrid.com/2017/09/ultralong-ranged-wifi-station.html>
- Halfacree, G. (2019, 7 3). *Raspberry Pi 4 Firmware Update Available: Less Heat, Same Speeds*. Retrieved from Tom's Hardware Guide:
<https://www.tomshardware.com/news/raspberry-pi-4-firmware-update-tested,39791.html>
- Halfacree, G. (2020, 2 1). *Group test: Best Raspberry Pi 4 thermal cases tested and ranked*. Retrieved from MagPi Magazine:
<https://magpi.raspberrypi.org/articles/group-test-best-raspberry-pi-4-thermal-cases-tested-and-ranked>
- HardKernel. (n.d.). *Products*. Retrieved 8 16, 2020, from HardKernel:
<https://www.hardkernel.com/product/>
- Heath, N. (2019, 7 9). *What are the best Raspberry Pi alternatives? Everything you need to know about Pi rivals*. Retrieved from ZDNet:
<https://www.zdnet.com/article/what-are-the-best-raspberry-pi-alternatives-everything-you-need-to-know-about-pi-rivals/>
- Hoffman, C. (2017, 7 3). *How to Buy an SD Card: Speed Classes, Sizes, and Capacities Explained*. Retrieved from How-To Geek:

- <https://www.howtogeek.com/189897/how-to-buy-an-sd-card-speed-classes-sizes-and-capacities-explained/>
- Kali Linux. (n.d.). *Kali Linux Arm Images - Kali Linux RaspberryPi 2 (v1.2), 3 and 4 (64-Bit)*. Retrieved 8 15, 2020, from Kali Linux: <https://www.offensive-security.com/kali-linux-arm-images/>
- Kali Linux. (n.d.). *Raspberry Pi 4 and Kali*. Retrieved 8 15, 2020, from Kali Linux: <https://www.kali.org/news/raspberry-pi-4-and-kali/>
- Kinzie, K. (2018, 12 10). *Check if Your Wireless Network Adapter Supports Monitor Mode & Packet Injection*. Retrieved from Null Byte: <https://null-byte.wonderhowto.com/how-to/check-if-your-wireless-network-adapter-supports-monitor-mode-packet-injection-0191221/>
- McKay, D. (2019, 7 18). *What Is Reverse SSH Tunneling? (and How to Use It)*. Retrieved from How-To Geek: <https://www.howtogeek.com/428413/what-is-reverse-ssh-tunneling-and-how-to-use-it/>
- Mitchell, B. (2019, 11 10). *What Is the Range of a Typical Wi-Fi Network?* Retrieved from Lifewire: <https://www.lifewire.com/range-of-typical-wifi-network-816564>
- Mullen, C. (2019, 12 20). *The burden of business travel: 1 in 5 who travel for work get stressed about upcoming trips*. Retrieved from BizWomen: <https://www.bizjournals.com/bizwomen/news/latest-news/2019/12/the-burden-of-business-travel-1-in-5-who-travel.html?page=all>
- National Center for Immunization and Respiratory Diseases (NCIRD), Division of Viral Diseases. (2020, 8 6). *Travel during the COVID-19 Pandemic*. Retrieved from Centers for Disease Control and Prevention: <https://www.cdc.gov/coronavirus/2019-ncov/travelers/travel-during-covid19.html>
- Network Lessons. (n.d.). *Cisco IOS NAT Port Forwarding*. Retrieved 8 21, 2020, from Network Lessons: <https://networklessons.com/cisco/ccie-routing-switching/cisco-ios-nat-port-forwarding>
- Offensive Security. (n.d.). *Kali Linux Arm Images*. Retrieved 8 21, 2020, from Offensive Security: <https://www.offensive-security.com/kali-linux-arm-images/>
- OffSec Services Limited. (n.d.). *Advanced Package Management in Kali Linux*. Retrieved 8 21, 2020, from Kali Linux: <https://www.kali.org/tutorials/advanced-package-management-in-kali-linux/>
- OffSec Services Limited. (n.d.). *Kali Linux*. Retrieved 8 21, 2020, from Official Kali Linux Releases: <https://www.kali.org/kali-linux-releases/>
- OffSec Services Limited. (n.d.). *Kali Linux Tools Listing*. Retrieved 8 21, 2020, from Kali Tools: <https://tools.kali.org/tools-listing>
- OffSec Services Limited. (n.d.). *Kali's Default Credentials*. Retrieved 8 21, 2020, from Kali Linux: <https://www.kali.org/docs/introduction/default-credentials/>
- OffSec Services Limited. (n.d.). *Raspberry Pi*. Retrieved 8 21, 2020, from Kali Linux: <https://www.kali.org/docs/arm/kali-linux-raspberry-pi/>
- Raspberry Pi Foundation. (n.d.). *Downloads*. Retrieved 8 21, 2020, from Raspberry Pi Foundation: <https://www.raspberrypi.org/downloads/>

- Raspberry Pi Foundation. (n.d.). *Raspberry Pi 4*. Retrieved 8 2020, from Raspberry Pi: <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/>
- Raspberry Pi. (n.d.). *Raspberry Pi*. Retrieved 8 19, 2020, from Downloads: <https://www.raspberrypi.org/downloads/>
- Raspberry Pi. (n.d.). *Raspberry Pi 4 boot EEPROM*. Retrieved 8 19, 2020, from Raspberry Pi: <https://www.raspberrypi.org/documentation/hardware/raspberrypi/boot-eprom.md>
- Raspberry Pi. (n.d.). *Raspberry Pi 4 Tech Specs*. Retrieved 8 2020, from Raspberry Pi: <https://www.raspberrypi.org/products/raspberry-pi-4-model-b/specifications/>
- Raspberry Pi. (n.d.). *Raspberry Pi PoE HAT*. Retrieved 8 16, 2020, from Raspberry Pi: <https://www.raspberrypi.org/products/poe-hat/>
- Raspberry Pi. (n.d.). *Schematics - USB-C POWER IN*. Retrieved 8 16, 2020, from Raspberry Pi: https://www.raspberrypi.org/documentation/hardware/raspberrypi/schematics/rpi_SCH_4b_4p0_reduced.pdf
- SSH Communications Security, Inc. (n.d.). *Public Key authentication for SSH*. Retrieved 8 21, 2020, from SSH.com: <https://www.ssh.com/ssh/public-key-authentication>
- SSH.com. (n.d.). *Authorized_keys File in SSH*. Retrieved 8 22, 2020, from SSH.com: https://www.ssh.com/ssh/authorized_keys/
- Tenable, Inc. (n.d.). *Software Requirements*. Retrieved 8 15, 2020, from Nessus: <https://docs.tenable.com/nessus/Content/HardwareRequirements.htm>
- Tomaschik, D. (2020, 7 14). *Raspberry Pi as a Penetration Testing Implant (Dropbox)*. Retrieved from System Overlord: <https://systemoverlord.com/2020/07/14/raspberry-pi-as-a-penetration-testing-implant.html>
- Tomaschik, D. (2020, 7 14). *Raspberry Pi as a Penetration Testing Implant (Dropbox)*. Retrieved from System Overlord: <https://systemoverlord.com/2020/07/14/raspberry-pi-as-a-penetration-testing-implant.html>
- Tyson, J. (n.d.). *How Network Address Translation Works*. Retrieved 8 21, 2020, from How Stuff Works: <https://computer.howstuffworks.com/nat.htm>
- Vilros. (n.d.). *Vilros Raspberry Pi 4 Compatible Self Cooling Heavy Duty Aluminum Alloy Case*. Retrieved 8 16, 2020, from Vilros: https://cdn.shopify.com/s/files/1/0195/1344/2404/products/Aluminum_case_for_pi_4_without_fan-1_2000x.jpg?v=1581159277
- Vilros.com. (n.d.). *Raspberry Pi*. Retrieved 8 2020, from Vilros: <https://vilros.com/collections/raspberry-pi-kits>
- Wu, E. (2019, 9). *Compare Raspberry Pi 4 Heatsink and Cooling Fan*. Retrieved from Seeed Studio: <https://www.seeedstudio.com/blog/2019/09/18/raspberry-pi-4-cooling-test-compare-heat-sinks-and-cooling-fan/>
- Ylonen, T. (2006, 1). *The Secure Shell (SSH) Authentication Protocol*. Retrieved from Internet Engineering Task Force (IETF): <https://tools.ietf.org/html/rfc4252>