Pwnie Express User Manual

“Pwn Pro” Fixed Sensor

The latest version of this manual is maintained here:

https://www.pwnieexpress.com/support/

Contents

Legal Disclaimers

- All Pwnie Express products are for legally authorized use only.
- By using this product you agree to the terms of the Rapid Focus Security, Inc. EULA: (http://www.pwnieexpress.com/wp-content/uploads/2014/12/Pwnie-Express-EULA-10-13-14-.pdf)
- This product contains both open source and proprietary software:
    - Proprietary software is distributed under the terms of the Rapid Focus Security, Inc. EULA: (http://www.pwnieexpress.com/wp-content/uploads/2014/12/Pwnie-Express-EULA-10-13-14-.pdf)
    - Open source software is distributed under one or more of the following licenses:
        - GNU PUBLIC LICENSE (https://www.gnu.org/licenses/gpl.html)
        - BSD-3-CLAUSE LICENSE (http://opensource.org/LICENSES/BSD-3-CLAUSE)
        - OPENSSL TOOLKIT DUAL LICENSE (https://www.openssl.org/source/license.html)
        - APACHE LICENSE, VERSION 2.0 (https://www.apache.org/licenses/LICENSE-2.0.html)
- As with any software application, any downloads/transfers of this software are subject to export controls under the U.S. Commerce Department's Export Administration Regulations (EAR): (http://www.bis.doc.gov/index.php/regulations/export-administration-regulations-ear). By using this software, you certify your complete understanding of and compliance with these regulations.

Specifications

- **Hardware**
    - Processor: 1.8GHz Intel i3 64bit (Dual Core – Four Threads)
    - Memory: 4GB 1600MHz DDR3
    - Disk Storage: 32GB mSATA SSD
    - Onboard I/O: 1x Gigabit Ethernet, 2x USB ports, 1x HDMI
    - Dimensions: 7.7" x 5.2" x 1.5"
- **Wireless**
    - Onboard high-gain dual-band 802.11 a/b/g/n/ac wireless supporting packet injection & monitor mode (with detachable antennas); poles #2 & #4
    - Onboard high-gain Bluetooth (up to 300ft. range) supporting packet injection & monitor mode (with detachable antennas); pole #1
- **4G GSM LTE USB Adapter**
    - External 6-band (worldwide) 4G GSM cellular USB adapter
    - Compatible with SIM cards from AT&T, T-Mobile, Vodafone, Orange, and GSM carriers in over 160 countries (SIM card not included)
    - 4G/LTE DD (up to 150Mbps): 800/900/1800/2100/2600MHz
    - 3G/DC-HSPA+/HSPA+/UMTS (up to 84Mbps): 900/1700(AWS)/1900/2100MHz
    - SIM/USIM card: standard 6-pin SIM card interface

Getting Started

1. Connect the onboard Ethernet port (eth0) to the local area network (LAN).
2. Connect the AC adapter to a power source.
3. Press the square power button on the front of the sensor.

4. Use a web browser to login to the Pwnix UI, acquire all available updates and configure the sensor.
5. After the sensor has been configured for use, the sensor is accessed either through an SSH connection established to the sensor or through a Reverse SSH connection established from the sensor.

Note: By default, the Ethernet port (eth0) on the sensor uses DHCP to acquire its IP address. Whereas this will initially be unknown, it is recommended to run nmap (e.g. **`nmap –p 22,1443 --open x.x.x.x/x`**) to scan the network subnet from a laptop (or desktop) to discover the IP assigned to the eth0 interface. Once learned, open a web browser and go to https://*x.x.x.x*:1443 (where x.x.x.x is the IP address assigned) to access the Pwnix UI.

If DHCP is not used within the LAN or if the sensor is unable to acquire an IP address via DHCP, the sensor has a virtual Ethernet interface (eth0:1) with IP address 192.168.9.10/24 already assigned. To access the sensor via its virtual Ethernet interface will require IP address 192.168.9.11/24 assigned to a laptop (or desktop). Then, from this system plugged into the same LAN the sensor is located within and assuming the network allows the simultaneous broadcast of the additional 192.168.9.0/24 network, the laptop or desktop should be able to PING the IP address assigned to the sensor, (i.e. **ping 192.168.9.10**). If successful, open a web browser and go to https://192.168.9.10:1443 to access the Pwnix UI.

Tip: If unsuccessful with accessing the virtual Ethernet interface in this manner, the most likely cause would be the network not allowing the simultaneous broadcast of the 192.168.9.0/24 network. To resolve, simply connect the Ethernet interface on the sensor directly to the Ethernet interface on the laptop (or desktop), then try to ping the IP address assigned to the sensor's virtual Ethernet interface, (e.g. **ping 192.168.9.10**). If successful, open a web browser and go to https://192.168.9.10:1443 to access the Pwnix UI.

Using the Pwnix UI

Note: When accessing the Pwnix UI for the first time, you will need to accept/acknowledge the warning about a self-signed certificate to continue.

1. At the login prompt, enter **pwnie** for the username and enter **pwnplug8000** (the default) for the password.
2. Next, the "Setup" page appears.

Setup page

System Authentication

1. From the "Setup" page, under "General Configuration", click "System Authentication".
2. Enter the current password for the "pwnie" user account, then enter a new password and provide its confirmation.

   IMPORTANT: We strongly recommend changing the default password for the "pwnie" user account as soon as possible.

3. Next, click "Change Password" button.

   Note: This will change the password for the "pwnie" UI user and the "pwnie" system (Linux/SSH) account. Pwnix UI authentication is integrated with Linux PAM, allowing the UI and system passwords to be synchronized for the "pwnie" user.

4. Afterward, click "Logout" on the top menu and then re-login to re-authenticate with your new credentials.

Tip: After the sensor is deployed, and if accessing the sensor through an SSH connection, you can also set the "pwnie" user's password via the command line, as shown:

```
# passwd pwnie
```

Note: Please note that if you change the password from the command line it will change the Pwnix UI password as well.

Network Config

Note: If the sensor is being accessed via the virtual Ethernet interface (eth0:1), then it is recommended to manually set an IP address to the Ethernet interface (eth0), restart the sensor, then access the sensor via the Ethernet interface (eth0) and the IP address assigned from this point forward.

1. From the "Setup" page, under "General Configuration", click "Network Config".
2. Under "Current Network Settings", the Ethernet network interface(s) will be displayed reflecting the IP assigned and their state.

   Depending on configuration, the following interfaces may be visible:
   eth0 - This Ethernet interface is located on the rear of the sensor and is configured for DHCP by default
   eth0:1 - This virtual Ethernet interface is used if DHCP is not available
   wlan0 - The onboard 802.11 wireless adapter (DOWN by default)
   wlan0mon - The onboard 802.11 wireless adapter in monitor mode (DOWN by default)

3. To change the IP configuration for eth0, click the "eth0" link in the adapter table.
4. Under "Configuration for eth0", the Ethernet interface eth0 is configured to use DHCP by default. To set a static IP for eth0, click "Static Config" under "Configure eth0 Settings", then enter a new IP Address, Netmask, Default gateway IP, and Primary DNS server IP.
5. Afterward, click the "Apply Static IP Settings" button.

Note: After the sensor's IP address is changed for the Ethernet interface (eth0), logout and reconnect to the Pwnix UI using the newly assigned IP address.

If the IP address to the Ethernet interface eth0 has been set to a Static IP, to set eth0 to acquire its network settings from a DHCP server instead, select "DHCP", then click the "Enable" button.

- Under "Change eth0 MAC Address", the current MAC Address assigned to the Ethernet interface (eth0) is displayed. To change the MAC Address, enter a new MAC Address and click the "Change MAC" button.

  Note: The eth0 MAC address will always revert back to the hardware default if the sensor is rebooted.

Tip: If it is desired to shut down the virtual Ethernet interface (eth0:1), this can be performed by running the following command with an SSH connection with the sensor.

# ifdown eth0:1

- Under "Change Hostname", the current hostname assigned to the sensor is displayed. To change the hostname, provide a new hostname and click the "Change Hostname" button.

Tip: After changing the hostname, log out of any active SSH terminal sessions, then re-login to update the prompt.

- Under "Configure NTP Servers", the current NTP servers are displayed. To change the NTP Servers for use, enter two or more NTP Servers and click the "Configure NTP" button. Afterward, go to the "Services" page and under "Service Management", select "Manage Service" for NTP and ensure it is set to enabled.

Reverse Shell Key

1. From the "Setup" page, under "General Configuration", click "Reverse Shell Key".
2. Under "Current Key", if it has already been generated, the current SSH Public Key for the sensor is displayed, which is used establish the Reverse Shells.

   Tip: If no key pair is displayed a key pair will be generated automatically after enabling one or more reverse shells on the "Reverse Shells" page.

   Optional: If desired to create or change the key pair, click the "Generate" button (found under "Generate New Key") to generate a new key pair for the Reverse Shells.

Register to Pwn Pulse

Pwn Pulse is a distributed security assessment service providing unparalleled visibility, vulnerability scanning, and insight into your remote branch offices and networks. By aggregating data from multiple sensors, Pwn Pulse is able to combine real time Wired, Wireless, and Bluetooth asset discovery with local, on-demand vulnerability scanning to provide unprecedented risk awareness of your organization's hardest to reach areas.

If you have purchased a subscription to Pwn Pulse to allow a specific number of sensors and you have completed the registration process to successfully login to Pwn Pulse, you are eligible to "join" that number of sensors to Pwn Pulse by performing the steps below.

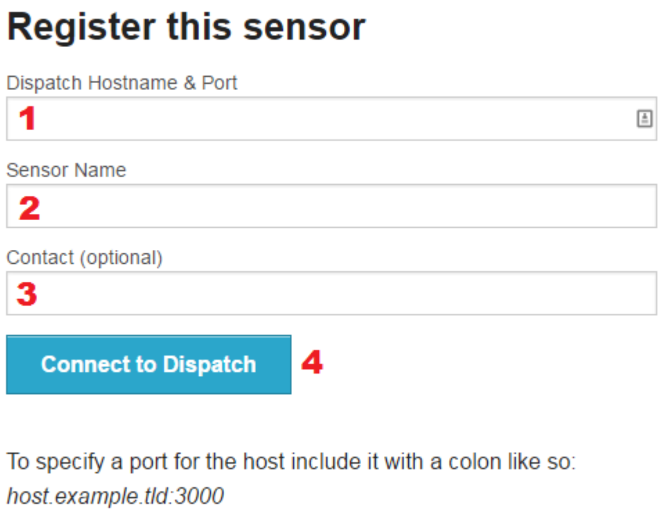
IMPORTANT: Do not register (join) the sensor to Pwn Pulse unless the sensor is already physically located within the network it is to report data upon. Additionally note the connectivity from the sensor requires TCP port 443 to the Internet and because of bidirectional certificates used to secure the connection, the use of a proxy server is unsupported.

1. From the "Setup" page, under "General Configuration", click "Register to Pwn Pulse"
2. When a sensor has yet to be registered, three fields will be visible; see screenshot below.
   a. Provide the Dispatch Hostname (i.e. Pwn Pulse) required for use (#1). Specifying the port is not required.
   b. Provide the Sensor Name required for use (#2). It is encouraged the name of the sensor is indicative of the geolocation location or branch office where the sensor is located.

   c. Provide the Contact Name (#3) of an individual who might be responsible for the administration of the sensor

3. After providing the required fields, click the "Connect to Dispatch" button (#4) to join the sensor to Pwn Pulse.
4. Wait five minutes, then login to Pwn Pulse and go to the "Unapproved Sensors" page, located under the "Wrench" in the upper right corner.
5. When the sensor appears on the list, click the Approve button to complete the process of joining the sensor to Pwn Pulse.
6. In approximately 10-15 minutes, the main Dashboard will begin reflecting statistics based upon the default tasks configured to run by the sensor.
7. Repeat the above steps to join additional sensors to Pwn Pulse.

   Tip: For further information on configuring Pwn Pulse, please review the "*Pwn Pulse User Manual*" available from the Support site (http://www.pwniexpress.com/support")



To specify a port for the host include it with a colon like so:
host.example.tld:3000

Clean up History and Logs

From the "Setup" page, under "General Configuration", click the "Cleanup now" button.

This clears the root user's bash history, UI logs, and all logs in /var/log.

Note: The bash history for any currently active root user sessions will be cleared upon the next logout.

Tip: If accessing the sensor through an SSH connection, the cleanup script can also be run from the command line, as shown:

```
# /opt/pwnix/pwnix-scripts/cleanup.sh
```

Update Device

IMPORTANT: To acquire updates, the following external web sites ***must*** be accessible from the sensor via the ports specified below. If access to one or more of the sites is blocked the result of a firewall, proxy server or web filtering solution in use, the update process will fail rendering the Pwnix UI to become inaccessible. If it is necessary, add the exclusion(s) below to the firewall, proxy server or web filtering solution ***before*** attempting to update the sensor:

> Allow TCP port 443 to updates.pwnieexpress.com
> Allow TCP port 443 to kalirepo.pxinfra.net
> Allow TCP port 443 to gemrepo.pxinfra.net
>
> **Note**: If the sensor is registered with Pwn Pulse, TCP port 873 will be additionally required to openvas-feed.intevation.org

1. From the “Setup” page, under “Update Device”, click the “Update Now” button.

   Note: Depending upon connection speed and the number of updates to install, please allow 3-5 minutes to allow the process to complete. While updating the sensor, the Pwnix UI may become temporarily unavailable. To determine if/when the process has completed, please review the “*Pwnix Update Log*” found under “System Logs”, on the “System Details” page. When the update process has completed, the last line in the log will show “*System update completed*”.

   Tip: If accessing the sensor through an SSH connection, you can also update the sensor as follows:

   ```
   # /opt/pwnix/chef/update.sh
   ```

   Note: After the update has completed it is recommended to restart the sensor.

Restart Device

1. From the “Setup” page, under "Restart Device", click the "Reboot Now" button and the sensor will reboot immediately.

   Tip: If accessing the sensor through an SSH connection, you may also restart the sensor as follows:

   ```
   # reboot
   ```

Services page

From the Services page, the Passive Recon, Evil AP, and NTP services can be managed (i.e. enabled or disabled)

Passive Recon

From the “Services” page, under “Service Management”, click “Manage Service” to change the status of the Passive Recon service. Depending upon its status, click “Enable Service” to start the passive recon service or “Disable Service” to stop the passive recon service.

Note: When enabled, the sensor will passively listen on the Ethernet interface eth0, recording HTTP requests, user-agents, cookies, OS guesses, and clear-text passwords to the following logs:

HTTP requests: /var/log/pwnix/passive_recon/http.log
OS guesses: /var/log/pwnix/passive_recon/p0f.log
Clear-text passwords: /var/log/pwnix/passive_recon/dsniff.log

Note: Passive Recon is most effective when the sensor is in NAC Bypass / transparent bridging mode, or when connected to a switch monitor/SPAN port or network tap.

Tip: If accessing the sensor through an SSH connection, you may also enable or disable the Passive Recon service from the command line, as follows:

To enable the Passive Recon service, run the following commands:

```
# service pwnix_passive_recon start
# update-rc.d pwnix_passive_recon defaults
```

To disable the Passive Recon service, run the following commands:

```
# service pwnix_passive_recon stop
# update-rc.d -f pwnix_passive_recon remove
```

Evil AP

From the "Services" page, under "Service Management", click "Manage Service" to change the status of the "Evil AP" service. Depending upon its status, click "Start Service" to start the Evil AP service or "Disable Service" to stop the Evil AP service.

Note: When enabled, the Evil AP will perform a "Karma" attack against **all** wireless client devices, which results with wireless client devices to connect automatically to the Evil AP created by the sensor, based upon the broadcasted probe requests learned from the client devices. When wireless client devices connect to the Evil AP, the traffic from the wireless client devices is routed through the Ethernet interface (eth0).

Tip: If accessing the sensor through an SSH connection, to view real-time Evil AP activity from the command line, type the following:

```
# tail -f /var/log/pwnix/evilap.log
```

Tip: If accessing the sensor through an SSH connection, you may also enable or disable the Evil AP service from the command, as follows:

To enable Evil AP, run the following commands:

```
# service pwnix_evil_ap start
# update-rc.d pwnix_evil_ap defaults
```

To disable Evil AP, run the following commands:

```
# service pwnix_evil_ap stop
# update-rc.d -f pwnix_evil_ap remove
```

Tip: The configuration of the Evil AP can customized in /opt/pwnix/Pwnix config/services/evil_ap.conf file.

NTP

From the “Services” page, under “Service Management”, click “Manage Service” to change the status of the “NTP” service. Depending upon its status, click “Start Service” to start the NTP service or “Disable Service” to stop the NTP service.

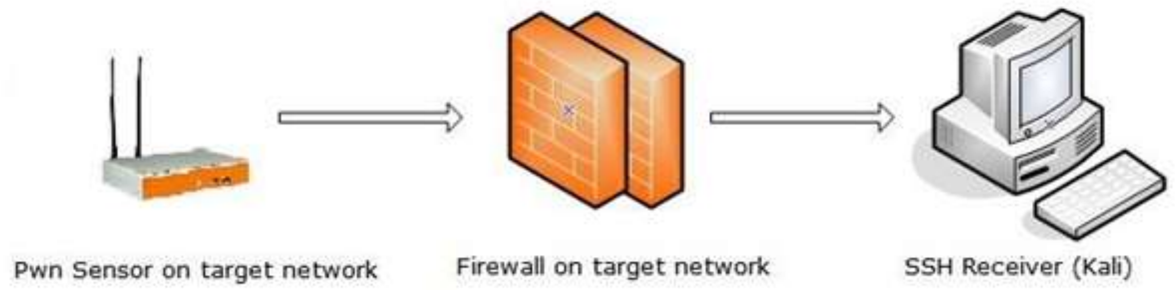
Tip: If the NTP servers have been modified on “Setup” page, it is required to stop/start the NTP server for the changes to take effect.

Reverse Shells page

- All Pwnie sensors include aggressive, reverse tunneling capabilities to allow persistent, remote SSH access.
- The use of reverse SSH access provides the ability to use the sensor as a “beach-head” or “pivot point” to perform testing of the remote network.
- SSH over HTTPS/SSL, SSH over DNS, SSH over ICMP, and other covert tunneling options are available for traversing strict firewall rules, web filters, & application-aware IPS solutions.
- All tunnels are encrypted and will maintain access wherever the sensor has an Internet connection.

Typical deployment overview

1. On a staging/lab network, enable the desired reverse shells (see “Activating the reverse shells”)
2. Next, configure a Kali Linux system (i.e. “SSH Receiver”) to receive the reverse shells (see “Configuring Kali to receive the reverse shells”)
3. Test the reverse shells to confirm all enabled shells are working as expected (see “Connecting to the reverse shells”)
4. Deploy the sensor to your target network and watch your SSH receiver for incoming shells (see “Deploying to target network”)



Activating the reverse shells

1. Open a browser and log into the Pwnix UI.
2. Click “Reverse Shells” on the top menu.
3. Choose the shell you wish to configure from the drop-down list.

Tip: To best maintain persistent remote access, enable all of the reverse shells.

4. Enter the SSH shell receiver IP address or DNS name for each selected reverse shell, as well as any other configuration settings as necessary. The device will connect to this shell receiver system to establish the reverse shell connections.
5. Click "Submit" to apply your changes for the reverse shell(s) you are configuring.

   Note: The following SSH client config directives (/etc/ssh/ssh_config) are set on all devices to allow for automation of reverse shell connections. Be sure you understand the security implications of these settings before connecting to other SSH servers from the sensor.

   ```
   StrictHostKeyChecking no
   UserKnownHostsFile /dev/null
   ```

6. After the reverse shell(s) have been configured, proceed to configure Kali to receive the reverse shells.

Configuring Kali to receive the reverse shells

A system running Kali Linux (1.0.6 or later) will serve as the "SSH Receiver" system. The SSH Receiver will "listen" for incoming connections from the sensor, which will be configured to use one or more reverse shells.

1. Place the sensor and the SSH Receiver system on the same local network/subnet
2. Login to the SSH Receiver system, open a web browser and connect to the sensor via the Pwnix UI (e.g. https://x.x.x.x:1443)
3. Login to the Pwnix UI when prompted.
4. From the top menu, click "Reverse Shells".
5. If they haven't already been configured, configure desired shells (*see "Activating the Reverse Shells" section*)
6. Next, click the "Download Shell Receiver script for Kali Linux" link at the top of the page to download the "pwnix_ssh_receiver.sh" script.
7. Save the script file (pwnix_ssh_receiver.sh) into the user's home directory (selected by default)
8. Open a Terminal window and enter the following commands:

   ```
   # chmod +x pwnix_ssh_receiver.sh
   #  /pwnix_ssh_receiver.sh
   ```

   Note: The first time the script is run on the SSH Receiver system, it will download and install various packages required to support the functionality.

9. When prompted, enter the desired certificate information for the stunnel SSL certificate (or just press ENTER to accept the defaults)
10. Once the auto-config script completes you will see:

    ```
    [+] Setup Complete.
    [+] Press ENTER to listen for incoming connections..
    ```

11. Press ENTER to watch for incoming device connections. This may take up to two minutes.

    Note: When an incoming connection from the sensor is successfully established to the SSH Receiver, the Reverse SSH method being used can be determined based upon the port number displayed amongst the listening connections.

    Port 3333 is associated with **Standard SSH**
    Port 3334 is associated with **Egress Buster**
    Port 3335 is associated with **SSH over DNS**
    Port 3336 is associated with **SSH over SSL**
    Port 3337 is associated with **SSH over 4G/GSM**
    Port 3339 is associated with **SSH over ICMP**

    Tip: If two minutes have passed and nothing is displayed under listening connections, this indicates the sensor is unable to establish a connection with the SSH Receiver. The most likely cause can be attributed to a firewall protecting the SSH Receiver system from being able to receive the incoming connection(s). If a firewall is protecting the SSH Receiver, ensure it is configured to allow the incoming port(s) relevant to the methods enabled.

    If **Standard Reverse SSH** is enabled, then forward the port specified in the UI (default is 22) to the same on the SSH Receiver system.
    If **SSH over SSL** is enabled, then forward/allow port 443 to the SSH Receiver system.
    If **SSH over DNS**, is enabled, then forward/allow UDP port 53 to the same on the SSH Receiver system.
    If **SSH over ICMP** is enabled, the ICMP protocol must be allowed to pass through the firewall. Generally, this will not be allowed. Hence SSH over ICMP may not be expected to work successfully.
    If **SSH over 4G** is enabled, then forward/allow the port specified in the UI to port 22 on the SSH Receiver system.
    If **SSH Egress Buster**, is enabled, then forward all of the ports specified in the UI (defaults are 21, 22, 23, 25, 110, 123, 161, 500, 1723, and 4500) to the SSH Receiver system.

12. Proceed to “Connecting to the reverse shells”.

Connecting to the reverse shells

1. Open a second terminal window on the SSH Receiver system and use the SSH command (with the –p switch) to establish a reverse connection using the port number displayed.

   For example, if ports 3333 and 3336 are the ports displayed as “Listening” when viewing the incoming connections, type either of the following to establish a reverse SSH connection to the sensor:

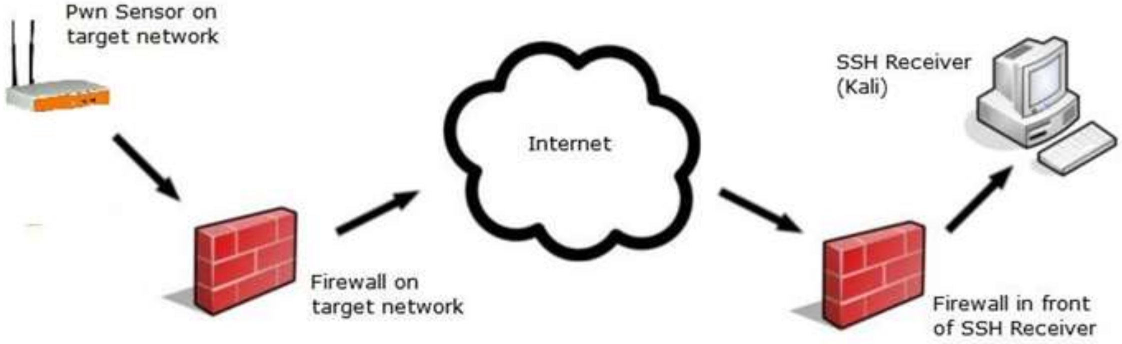
   ```
   # ssh pwnie@localhost –p 3333
   # ssh pwnie@localhost –p 3336
   ```

2. When prompted to login, enter “pwnie” for the username, the password and voila! You are now remotely connected to the sensor through the reverse shell.
3. Proceed to “Deploying to target network”

Tip: The SSH receiver address can be anonymized using the "Tor Hidden Service" feature as described here http://www.securitygeneration.com/security/reverse-ssh-over-tor-on-the-pwnie-express/

Special thanks to Sebastien J. of Security Generation for streamlining the SSH receiver setup process, and to Lance Honer for his resilient autossh script improvements.

Deploying to target network


Pwn Sensor on target network
Internet
SSH Receiver (Kali)
Firewall on target network
Firewall in front of SSH Receiver

1. Place the SSH Receiver system behind an Internet facing firewall.
2. Configure the appropriate rules on the firewall to allow:

   If enabling **Standard Reverse SSH**, then forward the port specified in the UI (default is 22) to port 22 on the SSH Receiver system.
   If enabling **SSH over SSL**, then forward port 443 to port 443 on the SSH Receiver system.
   If enabling **SSH over DNS**, then forward UDP port 53 to UDP port 53 on the SSH Receiver system.
   If enabling **SSH over ICMP**, the ICMP protocol must be allowed to pass through the firewall. Generally, this may not be allowed.
   If enabling **SSH over 4G**, then forward the port selected in the UI to port 22 on the SSH Receiver system.
   If enabling **SSH Egress Buster**, then forward all ports specified in the UI (defaults are) to port 22 on the SSH Receiver system.

3. In the Pwnix UI ("Reverse Shells" page), configure the reverse shells to connect to your firewall's public IP address (or DNS name if available).
4. You can now deploy your Pwnie sensor to your target network. The device will automatically "phone home" to the designated SSH Receiver system, providing encrypted remote access to your target network.

Tip: In some environments, you may wish to schedule a nightly reboot of the sensor to re-initiate all connections from the sensor side. This way, if some part of the connection process crashes on the sensor side (for example, sshd), the connection process will start "fresh" again after the reboot.

Using SSH port forwarders on Kali

Connecting to remote RDP servers

Perform the following steps to use the sensor as a “pivot point”, to connect to remote RDP servers, through a Reverse SSH connection.

1. First, run the Shell Receiver script on the system running Kali Linux.
2. Once the system is waiting for incoming connections, open a new Terminal window and run the following command:

   ```
   # ssh pwnie@localhost -p XXXX -NL 3389:xxx.xxx.xxx.xxx:3389
   ```

   .. where "*XXXX*" is the local listening port of an active reverse shell (such as 3333 for standard reverse SSH), and where "*xxx.xxx.xxx.xxx*" is the IP address of an RDP target system on the remote network your Pwnie sensor is physically connected to.

3. When prompted, login to the sensor using the “pwnie” user account.
4. Next, connect to the remote RDP server through the SSH tunnel by using the rdesktop command and "localhost" as the target server.

   ```
   # rdesktop localhost
   ```

Connecting to remote web servers

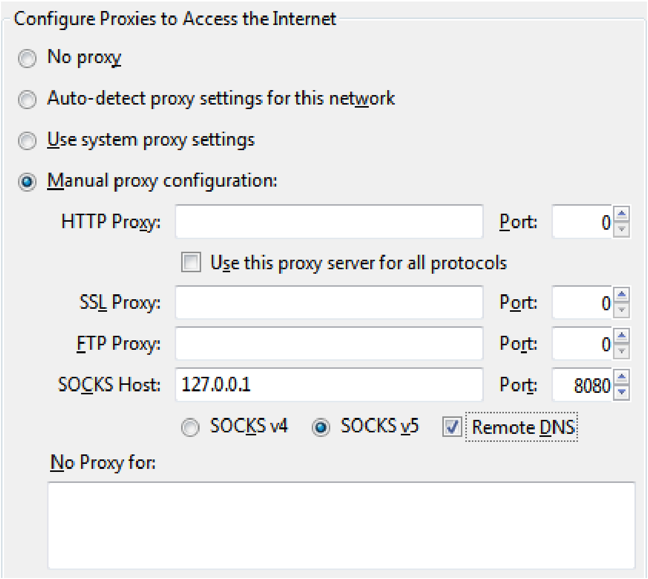
Perform the following steps to use the sensor as a “pivot point”, to connect to a remote web server, through a Reverse SSH connection.

1. First, run the Shell Receiver script on the system running Kali Linix.
2. Once the system is waiting for incoming connections, open a new Terminal window and run the following command:

   ```
   # ssh pwnie@localhost -p XXXX -ND 8080
   ```

   .. where "*XXXX*" is the local listening port of an active reverse shell (such as 3333 for standard reverse SSH).

3. When prompted, login to the sensor using the “pwnie” user account.
4. Next, open Firefox (install Firefox if necessary), and go to Options | Advanced | Network | Connections and click Settings. Configure the options as follows:



Configure Proxies to Access the Internet

○ No proxy

○ Auto-detect proxy settings for this network

○ Use system proxy settings

● Manual proxy configuration:

HTTP Proxy: Port: 0

□ Use this proxy server for all protocols

SSL Proxy: Port: 0

FTP Proxy: Port: 0

SOCKS Host: 127.0.0.1 Port: 8080

○ SOCKS v4 ● SOCKS v5 ☑ Remote DNS

No Proxy for:

5. Next, save the changes, then exit and re-open Firefox.
6. At this time, you can now connect to any web server on the remote network by entering the IP address or URL into address bar within Firefox.

Creating an SSH VPN

The OpenSSH server on the Pwn Pro supports SSH-based VPN tunneling through any active reverse shell, allowing transparent (albeit slow) access to your target network from the designated SSH Receiver system running Kali Linux. This is mainly useful when the need arises to access a GUI-based or third party, penetration testing tool, such as BurpSuite, Nessus, Remote Desktop client, etc.

Sample environment

The steps below assume the following IP addresses/ranges. Substitute the addresses/ranges for your target and local networks where appropriate.

- Target network (where the Pwn Pro is deployed): 172.16.1.0/24
- Local network (where the SSH Receiver system is located): 192.168.1.0/24
- VPN network: 10.1.1.0/30
- Kali VPN address (tun0 interface): 10.1.1.1
- Pwn Pro VPN address (tun0 interface): 10.1.1.2
- Assumes a reverse shell is currently established and listening on localhost:3333 (Standard Reverse SSH). Any active reverse shell can be used to carry the VPN tunnel (change "3333" where appropriate).

Activating the SSH VPN tunnel

1. Ensure that the root user on your Kali receiver has an SSH key generated. To do so, type the command below and follow the prompts.

```
#  ssh-keygen
```

2. Open Firefox on your Kali SSH receiver. From the reverse shells UI (https://*[device_ip_address]*:1443/reverse_shells), click "Download SSH VPN script for Kali Linux" to download the "ssh_vpn.sh" script.
3. Save the script file (ssh_vpn.sh) into the user's home directory (selected by default)
4. Open a terminal window and enter the following commands:

```
# cd
# chmod +x ssh_vpn.sh
#  ./ssh_vpn.sh
```

5. Follow the on-screen prompts to configure the SSH VPN. The default values assume a standard reverse shell to the target machine is open on port 3333 (see 'Configuring Kali to use the reverse shells' above).
6. When the script completes, the Kali receiver should have an interface 'tun0' that tunnels directly to the Pwn Pro. Type

```
# ping 10.1.1.2
```

to test the SSH VPN tunnel.

7. Type

```
# route add 172.16.1.0/24 gw 10.1.1.1
```

to route any traffic from the receiver to the 172.16.1.0/24 subnet over the VPN connection. (Substitute the subnet of the Pwn Pug R3 for 172.16.1.0/24.)

To disable the VPN tunnel on the Kali side:

```
# ifconfig tun0 down
```

To disable the VPN tunnel on the Pwn Pro side:

```
# service pwnix_ssh_vpn stop
```

Using the wireless hardware

802.11a/b/g/n/ac wireless

The wireless interface is internal and will appear as "wlan0" and "wlan0mon". Facing the rear of the Pwn Pro, the wireless interface provides dual antennas, identified as poles "2" & "4".

Connecting to an open Wi-Fi network

1. Set the wireless interface to managed mode by typing the following command and press Enter:

```
# iwconfig wlan0 mode managed
```

2. Bring the wireless interface up by typing the following command and press Enter:

```
# ifconfig wlan0 up
```

3. Scan for access points in the area by typing the following command and press Enter:

```
# iwlist scan
```

4. Associate with an access point with SSID “example” on channel 6 by typing the following commands and press Enter after each:

```
# iwconfig wlan0 essid “example”
# iwconfig wlan0 channel 6
```

5. Restart the interface by typing the following commands and press Enter after each:

```
# ifconfig wlan0 down
# ifconfig wlan0 up
```

6. Acquire a DHCP address by typing the following command and press Enter:

```
# dhclient wlan0
```

Running Airodump-ng & Kismet

1. Bring the wireless interface down by typing the following command and press Enter:

```
# ifconfig wlan0 down
```

2. Run airodump-ng by typing the following command and press Enter:

```
# airodump-ng wlan0
```

3. When finished, press CTRL+C to exit
4. Run Kismet by typing the following command and press Enter:

```
# kismet
```

5. Press ENTER 3 times, then press TAB, then press ENTER to display the main window.
6. When finished, press CTRL+C to exit

Tip: Certain wireless tools may leave the wireless adapter in a mode that is not compatible with other wireless tools. It is strongly recommended to set the interface to a “down” state before running most wireless tools by typing the following command and press Enter:

```
# ifconfig wlan0 down
```

Packet injection

1. To run a simple packet injection test, execute the following commands. This example assumes a WEP-enabled access point on channel 6 with SSID "*example*" is within range of the sensor.

   ```
   # ifconfig wlan0 up
   # iwconfig wlan0 channel 6
   # ifconfig wlan0 down
   # aireplay-ng -e example --test wlan0
   ```

2. Next, Look for output similar to the following::

   ```
   17:19:45  Waiting for beacon frame (ESSID: example) on channel 6
   Found BSSID "00:13:10:9E:52:3D" to given ESSID "example".
   17:19:45  Trying broadcast probe requests...
   17:19:45  Injection is working!
   17:19:46  Found 1 AP
   ```

Wireless client de-authentication

1. This example assumes the target access point is on channel 6:

   ```
   # iwconfig wlan0 channel 6
   ```

2. In one terminal, start airodump-ng:

   ```
   # airodump-ng --bssid [MAC of target AP] -c 6 wlan0
   ```

3. Then, in a second terminal, start the client de-authentication:

   ```
   # aireplay-ng -0 0 -a [MAC of target AP] -c [MAC of target client] wlan0
   ```

Bluetooth

Using the Bluetooth adapter

The Bluetooth interface is internal and will appear as "hci0". Facing the rear of the Pwn Pro, the antenna is closest to the left side, i.e. pole "1".

1. Confirm the output of the following commands:

   ```
   # lsusb
   ```

   Look for output similar to the following::

   ```
   Bus 001 Device 002: ID 0a12:0001 Cambridge Silicon Radio, Ltd Bluetooth
   Dongle (HCI mode)
   ```

   ```
   # hciconfig hci0
   ```

Look for output similar to the following::

```
hci0:   Type: BR/EDR  Bus: USB
BD Address: XX:XX:XX:XX:XX:XX  ACL MTU: 310:10  SCO MTU: 64:8
DOWN
RX bytes:466 acl:0 sco:0 events:18 errors:0
TX bytes:73 acl:0 sco:0 commands:17 errors:0
```

2. Enable the Bluetooth interface and set it to "Non-Discoverable":

   # hciconfig hci0 up
   # hciconfig hci0 noscan

3. To scan for remote Bluetooth devices

   # hcitool -i hci0 scan --flush --info –class

4. To ping the address of a remote Bluetooth device

   # l2ping -i hci0 *XX:XX:XX:XX:XX:XX*

5. To dump Bluetooth packets:

   # hcidump -i hci0 -t –X

6. To pair with a remote Bluetooth device:

   # bluez-simple-agent hci0 *XX:XX:XX:XX:XX:XX*

   IMPORTANT: Before disconnecting the USB Bluetooth adapter, always set the interface to a DOWN state first by running the command below.

   # hciconfig hci0 down

4G/GSM Cellular (Optional Accessory)

Using the unlocked GSM adapter

Either of the unlocked GSM adapters will support five GSM cell bands and is compatible with AT&T, T-Mobile, Vodafone, Orange, and GSM carriers in over 160 countries.

E369 2G/3G USB Modem Adapter

E3276 4G LTE Modem Adapter

List of mobile network operators of the Americas:
http://en.wikipedia.org/wiki/List_of_mobile_network_operators_of_the_Americas

List of mobile network operators of Europe:
http://en.wikipedia.org/wiki/List_of_mobile_network_operators_of_Europe

List of mobile network operators of the Asia Pacific region:
http://en.wikipedia.org/wiki/List_of_mobile_network_operators_of_the_Asia_Pacific_region

List of mobile network operators of the Middle East and Africa:
https://en.wikipedia.org/wiki/List_of_mobile_network_operators_of_the_Middle_East_and_Africa

IMPORTANT: Verizon, Sprint, Virgin Mobile, and other CDMA carrier SIMs will **not** work in the unlocked GSM adapter.

1. First, obtain a SIM card from the GSM cell provider of your choice.

   Note: The mobile service attached to the SIM card must have mobile broadband data service. Verify you can access the Internet from a tablet or other device using the SIM card before attempting to use the SIM card with the adapter. In addition, the carrier must allow "unauthorized" or "unapproved" devices to connect to their network.

2. Slide open the plastic cover on the GSM adapter.
3. Insert your SIM card into the adapter with the notch positioned as shown by the line drawing on the SIM slot, with the SIM card contacts facing down.

   Note: Many GSM phones, including the iPhone4+, use a micro-SIM instead of a standard-sized SIM card. To fit these SIM cards into the GSM adapter, use the included micro-SIM card adapter.

4. Slide the plastic cover back onto the adapter.
5. Connect the GSM adapter to either of the two rear USB ports.
6. Initially, the adapter will begin blinking green every three seconds. After 15-20 seconds have passed, the adapter will begin blinking either green or blue every three seconds.

   If using the E369 adapter, refer to the following:
   Green, blinking once every 3 seconds: The adapter is registering with a 2G network
   Blue, blinking once every 3 seconds: The adapter is registering with a 3G/3G+ network
   Green, solid: The adapter is connected to a 2G network
   Blue, solid: The adapter is connected to a 3G network
   Cyan, solid: The adapter is connected to a 3G+ network
   Off: The adapter is removed, i.e. not receiving power

   If using the E3276 adapter, refer to the following:
   Blue, blinking: 4G network available
   Cyan, blinking: 4G LTE network available
   Blue, solid: Connected to a 4G network

Cyan, solid: Connected to a 4G LTE network

7. Next, confirm the GSM adapter is detected properly by running the following command:

```
# gsmctl -d /dev/ttyUSB0 me
```

Depending upon which adapter is used, look for output similar to the following:

```
<ME0>  Manufacturer: Huawei Technologies Co., Ltd.
<ME1>  Model: E369
<ME2>  Revision: 41.102.18.00.00
<ME3>  Serial Number: 868414002759466
```

or

```
<ME0>  Manufacturer: Huawei
<ME1>  Model: E3276
<ME2>  Revision: 21.436.03.00.00
<ME3>  Serial Number: 863781018670399
```

Note: If the command returns “SIM failure”, the SIM card is either missing or not inserted properly. Remove the adapter, then remove the cover and check the SIM card.

8. To list cellular operators in range of the adapter, run the following command:

```
# gsmctl -d /dev/ttyUSB0 op
```

Look for output similar to the following:

```
<OP0>  Status: current  Long name: 'AT&T'   Short name: 'AT&T'   Numeric
name: 1
<OP1>  Status: available  Long name: 'AT&T'   Short name: 'AT&T'   Numeric
name: 1
<OP2>  Status: available  Long name: 'T-Mobile'   Short name: 'TMO'   Numeric
name: 1
<OP3>  Status: available  Long name: ''   Short name: ''   Numeric name: 1
<OP4>  Status: available  Long name: 'T-Mobile'   Short name: 'TMO'   Numeric
name: 1
<OP5>  Status: available  Long name: 'USA Verizon'   Short name: 'Verizon'
Numeric name: 1
<OP6>  Status: available  Long name: 'T-Mobile'   Short name: 'TMO'   Numeric
name: 1
```

9. To show the current attached operator (i.e. cellular carrier) the SIM card is associated with, run the following command:

```
# gsmctl -d /dev/ttyUSB0 currop
```

Look for output similar to the following:

```
<CURROP0>  Long name: 'AT&T'   Short name: 'AT&T'   Numeric name: 310410
```

```
Mode: automatic
```

10. To show signal strength of current operator connection:

    # gsmctl -d /dev/ttyUSB0 sig

    Look for output similar to the following:
    ```
    <SIG0>  19
    ```

    Note: If the value returned is 99, it indicates the signal is not detectable.

11. To check PIN status (READY = No PIN set):

    # gsmctl -d /dev/ttyUSB0 pin

    Look for output similar to the following:
    ```
    <PIN0> READY
    ```

Connecting to the Internet using the adapter

1. Run the pppd dialup script to establish a data connection with the associated carrier:

   # pppd nodetach call e160 &

   Look for output similar to the following:
   ```
   [1] 3609
   root@pwnix-c03fd56cc0f6:/home/pwnie# Script /usr/sbin/chat -vf
   /etc/ppp/peers/e160_chat finished (pid 3610), status = 0x0
   Serial connection established.
   using channel 2
   Using interface ppp0
   Connect: ppp0 <--> /dev/ttyUSB0
   rcvd [LCP ConfReq id=0x1 <asyncmap 0x0> <auth chap MD5> <magic 0x88bcbdf1>
   <pcomp> <accomp>]
   sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x8fa1f84e> <pcomp> <accomp>]
   sent [LCP ConfAck id=0x1 <asyncmap 0x0> <auth chap MD5> <magic 0x88bcbdf1>
   <pcomp> <accomp>]
   rcvd [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0x8fa1f84e> <pcomp> <accomp>]
   rcvd [CHAP Challenge id=0x1 <50f8e67f0abecd00>, name = ""]
   sent [CHAP Response id=0x1 <2da7d5a5cf5f182295846ad8fb5d240d>, name = "att"]
   rcvd [CHAP Success id=0x1 "Welcome!"]
   CHAP authentication succeeded: Welcome!
   CHAP authentication succeeded
   sent [IPCP ConfReq id=0x1 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
   rcvd [IPCP ConfNak id=0x1 <ms-dns1 10.11.12.13> <ms-dns2 10.11.12.14>]
   sent [IPCP ConfReq id=0x2 <addr 0.0.0.0> <ms-dns1 10.11.12.13> <ms-dns2
   10.11.12.14>]
   rcvd [IPCP ConfNak id=0x2 <ms-dns1 10.11.12.13> <ms-dns2 10.11.12.14>]
   ```

```
sent [IPCP ConfReq id=0x3 <addr 0.0.0.0> <ms-dns1 10.11.12.13> <ms-dns2
10.11.12.14>]
rcvd [IPCP ConfNak id=0x3 <ms-dns1 10.11.12.13> <ms-dns2 10.11.12.14>]
sent [IPCP ConfReq id=0x4 <addr 0.0.0.0> <ms-dns1 10.11.12.13> <ms-dns2
10.11.12.14>]
rcvd [IPCP ConfNak id=0x4 <ms-dns1 10.11.12.13> <ms-dns2 10.11.12.14>]
sent [IPCP ConfReq id=0x5 <addr 0.0.0.0> <ms-dns1 10.11.12.13> <ms-dns2
10.11.12.14>]
rcvd [IPCP ConfReq id=0x1]
sent [IPCP ConfNak id=0x1 <addr 0.0.0.0>]
rcvd [IPCP ConfNak id=0x5 <addr 10.184.32.71> <ms-dns1 172.26.38.1> <ms-dns2
172.26.38.2>]
sent [IPCP ConfReq id=0x6 <addr 10.184.32.71> <ms-dns1 172.26.38.1> <ms-dns2
172.26.38.2>]
rcvd [IPCP ConfReq id=0x2 <addr 10.184.32.71>]
sent [IPCP ConfAck id=0x2 <addr 10.184.32.71>]
rcvd [IPCP ConfAck id=0x6 <addr 10.184.32.71> <ms-dns1 172.26.38.1> <ms-dns2
172.26.38.2>]
not replacing existing default route via 192.168.11.1
local  IP address 10.184.32.71
remote IP address 10.184.32.71
primary   DNS address 172.26.38.1
secondary DNS address 172.26.38.2
Script /etc/ppp/ip-up started (pid 3617)
Script /etc/ppp/ip-up finished (pid 3617), status = 0x0
```

2. The adapter will establish an Internet connection within 10-20 seconds, assuming a cellular signal is available. Once connected and depending upon which adapter is used, you will see a solid green, blue or cyan LED on the adapter.   Next, press CTRL-C to exit the script and return to a prompt.
3. To confirm an IP address has been assigned to the adapter, run the following command:

   # ifconfig ppp0

   Look for output similar to the following:

   ```
   ppp0  Link encap:Point-to-Point Protocol
         inet addr:10.184.32.71  P-t-P:10.184.32.71  Mask:255.255.255.255
         UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
         RX packets:8 errors:0 dropped:0 overruns:0 frame:0
         TX packets:8 errors:0 dropped:0 overruns:0 carrier:0
         collisions:0 txqueuelen:3
         RX bytes:122 (122.0 B)  TX bytes:152 (152.0 B)
   ```

4. Next, to route traffic to the adapter for connectivity through the data carrier to the Internet, set the default route to use the Point-to-Point interface (ppp0) using the following commands:

   # route del default
   # route add default ppp0

5. Next, test connectivity using ping and traceroute commands:

# ping google.com

Look for output similar to the following:

```
PING google.com (173.194.123.40) 56(84) bytes of data.
64 bytes from lga15s47-in-f8.1e100.net (173.194.123.40): icmp_req=1 ttl=55
time=119 ms
64 bytes from lga15s47-in-f8.1e100.net (173.194.123.40): icmp_req=2 ttl=55
time=77.8 ms
64 bytes from lga15s47-in-f8.1e100.net (173.194.123.40): icmp_req=3 ttl=55
time=106 ms
```

# traceroute google.com

Look for output similar to the following:

```
traceroute to google.com (74.125.226.34), 30 hops max, 60 byte packets
 1  172.26.96.169 (172.26.96.169)  109.843 ms  129.739 ms  229.653 ms
 2  172.26.96.9 (172.26.96.9)  229.635 ms  229.612 ms  229.594 ms
 3  172.18.112.164 (172.18.112.164)  229.579 ms  229.564 ms  229.548 ms
 4  12.249.2.33 (12.249.2.33)  229.531 ms  53.747 ms  51.593 ms
 5  12.83.172.162 (12.83.172.162)  109.451 ms  109.429 ms  109.407 ms
 6  gar1.chsct.ip.att.net (12.122.105.57)  109.382 ms  109.362 ms  57.851 ms
 7  12.249.88.6 (12.249.88.6)  73.047 ms  73.027 ms  73.008 ms
 8  216.239.50.139 (216.239.50.139)  82.997 ms  82.978 ms  82.963 ms
 9  209.85.245.179 (209.85.245.179)  103.580 ms  87.549 ms  97.376 ms
10  lga15s43-in-f2.1e100.net (74.125.226.34)  97.314 ms  97.295 ms  97.277 ms
```

6. To close the connection and restore Internet connectivity through the Ethernet interface (eth0), run the following commands:

   # killall -s SIGHUP pppd
   # ifdown eth0 && ifup eth0

Accessing the pentesting tools

Accessing Metasploit

The Metasploit binaries (msfconsole, msfcli, etc.) can be run from any directory. Simply type 'msfconsole' to launch the local Metasploit Console.

Note: For information on how to use Metasploit, go to http://www.offensive-security.com/metasploit-unleashed/Main_Page

Running additional pentesting tools

Thanks to the rock stars at the Kali Linux project (kali.org), the penetration tools listed below are preinstalled as Debian packages and can be run from any path on the system:

aircrack-ng	Gpsd	p0f	Sslstrip
Amap	Grabber	Pingtunnel	Stunnel
arp-scan	hping3	Plecost	Tcpflow
Arping	Httptunnel	Proxychains	thc-ipv6
Bed	Hydra	Proxytunnel	theharvester
Bluelog	Iodine	Redfang	tinyproxy
Bluez	John	Scapy	ubertooth
cisco-auditing-tool	Kismet	Setoolkit	udptunnel
cisco-global-exploiter	Lbd	sendEmail	ussp-push
cryptcat	mdk3	Sipcrack	waffit
darkstat	Metagoofil	Sipsak	wapiti
Dmitry	Miranda	Skipfish	weevely
dns2tcp	Miredo	smtp-user-enum	
dnsenum	Nbtscan	Snmpcheck	wifitap
dnstracer	Netcat	Socat	wifite
Dsniff	Netdiscover	Sqlmap	xprobe2
ettercap	Ngrep	Sqlninja	
Fierce	Nikto	Ssldump	
Fimap	Nmap	Sslscan	
Fping	Onesixtyone	Sslsniff	

Pentesting Resources

Provided below are a few recommended resources.

Kali Linux Tools Listing:
http://tools.kali.org/tools-listing

PTES Technical Guidelines:
http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines

Metasploit Unleashed:
http://www.offensive-security.com/metasploit-unleashed/Main_Page

Packt Publishing:
https://www.packtpub.com/networking-and-servers?search=kali

Maintaining your Pwnie sensor

Updating the Pwnix software

Updates to the applications installed within the Pwnix software platform are frequent. As a result, keep the sensor up-to-date as often as possible, or at least before beginning any engagements. Updating the sensor takes minutes and if desired, automated via a cron job.

To update the Pwnix software platform to the latest release (including security updates), follow the steps shown in section "Using the Pwnix UI → Setup page → Update device".

Reviewing the Pwnix environment

Show device software revision:

```
# grep Release /etc/motd
```

Show kernel version:

```
# uname –r
```

Show date/time:

```
# date
```

Show filesystem disk usage (note your disk usage may vary):

```
# df –h
```

Show CPU details:

```
# cat /proc/cpuinfo
```

Show total memory:

```
# grep MemTotal /proc/meminfo
```

Show current eth0 config:

```
# ifconfig eth0
```

Show currently listening TCP/UDP services (note dhclient won't be present if not using DHCP):

```
# netstat –lntup
```

Check syslog for errors, warnings, etc:

```
# egrep -i "warn|fail|crit|error|bad|unable" /var/log/messages
```

Show Ruby version:

```
# ruby –v
```

Show Perl version:

```
# perl –v
```

Show Python version:

```
# python –V
```

How to obtain support

All Pwn products come with FREE technical support during the first thirty-days from the initial date of purchase. After thirty-days, the ability to obtain technical support requires a subscription to "*Pwnie Care*".

- *What is the URL to the Support Center or to read product-related support articles?*

  The Support Center is available at http://support.pwnieexpress.com

- *What is the URL to the Support Portal?*

  The Pwnie Express Support Portal is available at http://www.pwnieexpress.com/pages/support

- *What is the URL to visit the Support Forum?*

  The Pwnie Express Support Forum is available at http://forum.pwnieexpress.com

- *What is the email address to submit technical support requests?*

  You can submit an email to request support via the Support Portal at http://support.pwnieexpress.com/customer/portal/emails/new or you can send an email to support@pwnieexpress.com

- *What is the phone number to call to request technical support?*

  Call our main number at 855-793-1337, then select option 3