Pwnie Express User Manual

Fixed sensor product line

Pwn Plug R2

(http://pwnieexpress.com/products/pwn-plug-r2)

Pentesting Appliance

(http://pwnieexpress.com/products/enterprise-pentesting-appliance)

The latest version of this manual is maintained here:
http://www.pwnieexpress.com/pages/documentation

Table of Contents:

- Legal disclaimers
- Getting started
- Using the Pwnix UI
  - Accessing Pwnix UI
  - Setup page
    - System Authentication
    - Network Config
    - Reverse Shell Key
    - Clean up History and Logs
    - Update Device
    - Restart Device
  - Services page
    - Passive Recon
    - Evil AP
  - Reverse Shells page
  - System Details page
  - Help page
- Using the reverse shells
  - Reverse shell overview
  - Typical deployment overview
  - Activating the reverse shells
  - Configuring Backtrack to receive the reverse shells
  - Connecting to the reverse shells
  - Deploying to target network
  - Using SSH port forwarders on Backtrack
    - Example 1: Connecting to remote RDP servers
    - Example 2: Connecting to remote web servers
- Using the wireless hardware
  - 802.11 wireless
    - Connecting to an open wifi network
    - Running Airodump-ng & Kismet
    - Packet injection & WEP cracking
    - Wireless client de-authentication
  - Bluetooth
    - Using the Bluetooth adapter
  - 4G/GSM cellular
    - Using the unlocked GSM adapter
    - Connecting to the Internet via 3G/4G
    - Using the SSH-over-3G shell
- Accessing the pentesting tools
  - Accessing Metasploit
    - Accessing Metasploit via msfrpcd
  - Running additional pentesting tools

- Pentesting Resources
- Using advanced features
  - Stealth Mode
  - NAC/802.1x Bypass (Pwn Plugs only)
    - NAC Bypass overview
    - Enabling NAC Bypass mode
    - NAC Bypass troubleshooting
    - Disabling NAC Bypass mode
  - Setting up an encrypted partition (Pentesting Appliance only)
- Maintaining your Pwnie device
  - Updating the Pwnix software
  - Accessing the local serial console (Pwn Plugs only)
  - Reviewing the Pwnix environment
- How to get support

Legal disclaimers

- All Pwnie Express products are for legally authorized uses only. By using this product you agree to the terms of the Rapid Focus Security, Inc. EULA: (http://pwnieexpress.com/pdfs/RFSEULA.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://pwnieexpress.com/pdfs/RFSEULA.pdf). Open source software is distributed under one or more of the following licenses:

    - GNU PUBLIC LICENSE (HTTP://WWW.GNU.ORG/LICENSES/GPL.HTML).
    - BSD-3-CLAUSE LICENSE (HTTP://WWW.OPENSOURCE.ORG/LICENSES/BSD-3-CLAUSE):
    - OPENSSL TOOLKIT DUAL LICENSE (HTTP://WWW.OPENSSL.ORG/SOURCE/LICENSE.HTML)
    - APACHE LICENSE, VERSION 2.0 (HTTP://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). By using this software you certify your complete understanding of and compliance with these regulations.

Getting started

1. Connect the provided wireless antenna to the device:

    - **Pwn Plug R2:** Connect antenna to SMA jack on the side of the device.
    - **Pentesting Appliance:** Connect antenna to SMA jack on the rear of the device.

2. Connect the onboard Ethernet jack to a local network or switch:

    - **Pwn Plug R2:** Use the right-most Ethernet jack (eth0) on the rear of the unit.
    - **Pentesting Appliance:** Use the single Ethernet jack (eth0) on the rear of the unit.

3. Connect the AC adapter to a power source:

    - **Pwn Plug R2:** The device will power on automatically.
    - **Pentesting Appliance:** Press the oval button on front-panel to power up the appliance.

4. The default device IP address is 192.168.9.10 (netmask 255.255.255.0).
5. To access the device for the first time, configure your Linux/Mac/Windows system with the following IP settings:

    IP address: 192.168.9.11
    Netmask: 255.255.255.0

    Tip: On Linux hosts you can configure a virtual interface as shown:

```
# ifconfig eth0:1 192.168.9.11/24
```

6. Confirm connectivity to the device by pinging it:

   ```
   $ ping 192.168.9.10
   ```

7. You can now access the device through the Pwnix UI. Proceed to "Using the Pwnix UI" below.

   Tip: You can now also connect to the device via SSH (default login **pwnie : pwnplug8000**). From a Linux/Mac host, run the following command (For Windows users, we recommend the PuTTY SSH client):

   ```
   $ ssh pwnie@192.168.9.10
   ```

   Note: The "pwnie" system account is a standard user with sudo privileges. Most of the system commands and pentesting tools referenced in this manual must be run as root, as indicated by a hash tag (#) proceeding the command. Once logged in as "pwnie", you can sudo to root as follows:

   ```
   $ sudo su -
   ```

Using the Pwnix UI

Accessing Pwnix UI

1. Open a web browser and access the UI: https://*[device_ip_address]*:1443

   Tip: If accessing for the first time, the default URL is https://192.168.9.10:1443

2. The UI is SSL-enabled, but you will receive a warning as the certificate is self-signed.
3. At the login prompt, enter your username/password (default is **pwnie : pwnplug8000**)
4. The "Setup" page appears.

   Important: We recommend changing the default "pwnie" user password as soon as possible. Proceed to "System Authentication" below.

Setup page

System Authentication

1. Click "Setup" on the top menu.
2. Click "System Authentication"
3. Enter a new password for the "pwnie" user into both fields and click "Change password".

   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 synced for the "pwnie" user.

4. Click "Logout" on the top menu to re-authenticate with your new credentials.

   Tip: You can also set the "pwnie" user's password via the command line, as shown:

   ```
   # passwd pwnie
   ```

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

Network Config

1. Click "Setup" on the top menu.
2. Click "Network Config".
3. The device's onboard network interfaces are displayed under "Current Network Settings".

   Pwn Plug R2: By default, the Pwn Plug R2 ships with the following interfaces:

   - eth0 - The right-most Ethernet jack on the rear of the unit (configured for DHCP by default)
   - eth0:1 - The virtual default interface for initial access (192.168.9.10/24 by default)
   - eth1 - The left-most Ethernet jack on the rear of the unit
   - wlan0 - The onboard 802.11 wireless adapter (DOWN by default)

   Pentesting Appliance: By default, the Pentesting Appliance ships with the following interfaces:

   - eth0 - The Ethernet jack on the rear of the unit (configured for DHCP by default)
   - eth0:1 - The virtual default interface for initial access (192.168.9.10/24 by default)
   - wlan0 - The onboard 802.11 wireless adapter (DOWN by default)

4. To change the device's host name, enter a new host name and click "Change hostname".

   Tip: After changing the hostname, log out of any active terminal sessions to update your terminal prompt.

5. To configure NTP Servers enter 3-7 NTP Servers and click "Configure NTP".
6. To change the IP configuration for eth0, click the "eth0" link in the adapter table.

   - eth0 is configured for DHCP by default. To set a static IP for eth0, Select "Static Config", enter a new IP address, network mask, default gateway, and primary DNS server and click "Apply static IP settings".

     Note: After the device's IP address is changed, reconnect to the UI using the newly assigned

IP address.

- To set eth0 to acquire network settings from a DHCP server instead (recommended), click "Enable" on the DHCP Tab.

  Note: After switching to DHCP, you'll need to access the device via the virtual default interface (192.168.9.10) or local serial/KVM console to determine the new IP address assigned by DHCP. Once the new DHCP-assigned IP address is known, reconnect to the UI using the newly assigned IP address.

- To change the eth0 interface MAC address, enter a new MAC and click "Change MAC". Note that the eth0 MAC address will always revert back to the hardware default after a reboot.

  Tip: The virtual default interface (192.168.9.10) can be shut down by running the following at the SSH or serial console command line.

  ```
  # ifdown eth0:1
  ```

Reverse Shell Key

1. Click "Setup" on the top menu.
2. Click "Reverse Shell Key"
3. This section shows the current root user SSH key used to establish the reverse shells.
4. (Optional) To generate a new key pair for the reverse shells, click "Generate".

   Tip: If a key pair doesn't already exist, a new one will be automated generated after enabling one or more reverse shells on the "Reverse Shells" page.

Clean up History and Logs

1. Click "Setup" on the top menu.
2. Under "Clean up History and Logs", click the "Cleanup now" button.
3. 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 at next logout.

   Tip: The cleanup script can also be invoked from the command line as follows:

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

Update Device

1. Click "Setup" on the top menu.
2. Under "Update Device", click the "Update Now" button.

Note: The device must have Internet access via ports 80 and 443 for the update to succeed.

3. The latest stable Pwnix release is downloaded and applied (typically 3-5 minutes).

    - The current Pwnix version can be viewed under the "System Details" on the top menu
    - The Pwnix Update log can be view under the "System Details" on the top menu

Restart Device

1. Click "Setup" on the top menu.
2. Under "Restart Device", click the "Reboot Now" button.
3. The device will reboot immediately.

Services page

Passive Recon

1. Click "Services" on the top menu.
2. Click "Passive Recon".
3. Click "Enable Service" to start the passive recon service.
4. While enabled, the device will passively listen on 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

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

    Tip: The Passive recon service can also be enabled/disabled from the command line as follows:

    To enable:

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

    To disable:

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

Evil AP

1. Ensure the wireless antenna is connected to the device (see "Getting Started").
2. Click "Services" on the top menu.
3. Click "Evil AP".
4. Click "Start Service"
5. Wireless clients will begin connecting to the AP, either automatically via preferred network lists or by direct AP association.

   Tip: To view realtime Evil AP activity from the command line:

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

6. By default the device will function as a standard AP, transparently routing all client Internet requests through the wired interface (eth0).

   Tip: The Evil AP service can also be enabled/disabled from the command line as follows:

   To enable:

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

   To disable:

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

   Tip: The Evil AP interface and SSID can customized in /opt/pwnix/pwnix-config/services/evil_ap.conf

Reverse Shells page

See section "Using the reverse shells" for details on this feature.

System Details page

This section displays the device's software release level, system logs, disk usage, etc.

Help page

This section covers basic UI interface usage.

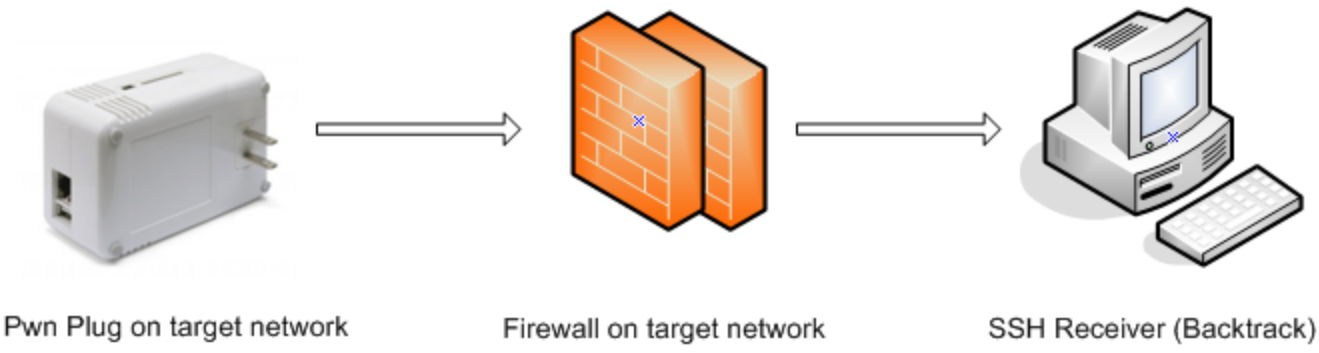
Using the reverse shells

Reverse shell overview

- All Pwnie devices include aggressive reverse tunneling capabilities for persistent remote SSH access.
- SSH over HTTP, HTTPS/SSL, DNS, ICMP, and other covert tunneling options are available for traversing strict firewall rules, web filters, & application-aware IPS.
- All tunnels are encrypted via SSH and will maintain access wherever the device has an Internet connection - including wired, wireless, and 4G/GSM where available.

Typical deployment overview

1. On a staging/lab network, enable the desired reverse shells (see "Activating the reverse shells")
2. Configure a Backtrack 5 system to receive the reverse shells (see "Configuring Backtrack to receive the reverse shells")
3. Test the reverse shells in a lab / local LAN to confirm all shells are working as expected (see "Connecting to the reverse shells")
4. [Optional] Enable Stealth Mode (see "Using the Pwnix UI")
5. Deploy the device to your target network and watch your SSH receiver for incoming shells (see "Deploying to target network")



Activating the reverse shells

1. Log into the Pwnix UI.
2. Click "Reverse Shells" on the top menu.

3. Click the name of the shell you wish to configure.

   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. The device will connect to this shell receiver system to establish the reverse shell connections.
5. Choose how often the reverse shell connection should be attempted. By default, a shell connection will be attempted every minute (recommended).

   Tip: To use an HTTP proxy for the "SSH over HTTP Tunnel", enable the "Use HTTP Proxy" checkbox and enter the proxy server address and port (and optionally, proxy server credentials).

   Note: The HTTP proxy auth password is stored in clear text in /opt/pwnix/pwnix-scripts/script_configs

6. Click "Configure" to apply your changes for the reverse shell you're 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 device.

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

7. Proceed to configure Backtrack to receive the reverse shells.

Configuring Backtrack to receive the reverse shells

A Backtrack 5 system (Backtrack 5 R3 recommended) can serve as the SSH tunnel "receiver". Your Pwnie device will connect to this system when initiating the reverse shell connections.

Note: These steps assume you're using Backtrack 5 R3 as your SSH receiver. Older Backtrack distributions may be used, but different steps may apply.

1. Place your Pwnie device and the Backtrack system on the same local network/subnet
2. Login to the Backtrack system and open Firefox
3. Connect to the UI: https://*[device_ip_address]*:1443
4. Login to the UI when prompted.
5. Click "Reverse Shells" on the top menu.
6. Click the "Generate Backtrack config" link at the top of the page (under step 5) to download the "backtrack_receiver.sh" script.
7. Save the script file (backtrack_receiver.sh) into the root user's home directory (selected by default)
8. Open a terminal window and enter the following commands:

   ```
   # cd

   # chmod +x backtrack_receiver.sh

   # ./backtrack_receiver.sh
   ```

9. The script auto-configures and starts the reverse shell listeners on Backtrack.
10. When prompted, enter the desired certificate information for the stunnel SSL certificate (or just press ENTER to accept the defaults)
11. Once the auto-config script completes you will see:

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

12. Press ENTER to watch for incoming device connections. Each reverse shell will attempt to connect using the interval you specified in the UI.

    Tip: You can list all active device connections at any time by typing:

    ```
    # netstat -lntup4 | grep 333
    ```

13. Proceed to "Connecting to the reverse shells".

Connecting to the reverse shells

1. Open a terminal window on your Backtrack shell receiver system and connect to any available "listening" Pwnie device shell as follows:

    - **Standard SSH:** ssh pwnie@localhost -p 3333
    - **SSH Egress Buster:** ssh pwnie@localhost -p 3334
    - **SSH over DNS:** ssh pwnie@localhost -p 3335
    - **SSH over SSL:** ssh pwnie@localhost -p 3336
    - **SSH over 4G/GSM:** ssh pwnie@localhost -p 3337
    - **SSH over HTTP:** ssh pwnie@localhost -p 3338
    - **SSH over ICMP:** ssh pwnie@localhost -p 3339

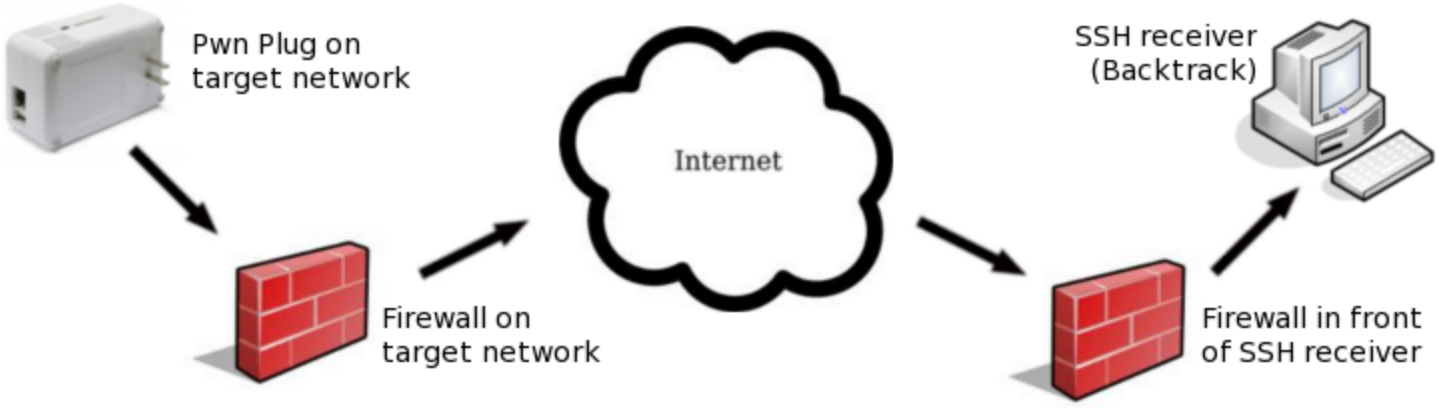
1. Enter your Pwnie device's "pwnie" user password and voila! You're now remotely connected to the device through the reverse shell.
2. Proceed to "Deploying to target network"

Standard SSH / SSH Egress Buster Note: If there's no firewall between your Pwnie device and your shell receiver system, be sure the shell receiver system SSH server is listening on the ports you selected for the Standard Reverse SSH and SSH Egress Buster shells in the UI. For example, if you set port 31337 for Standard Reverse SSH, add the line "Port 31337" to /etc/ssh/sshd_config, then restart SSHd (/etc/init.d/ssh restart).

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



1. Place your shell receiver system behind a public-facing firewall.
2. Configure the appropriate port forwarders on your firewall:

    - **Standard Reverse SSH:**
      Forward the port selected in the UI to port 22 of your shell receiver.

    - **SSH over HTTP:**
      Forward port 80 to port 80 of your shell receiver system

    - **SSH over SSL:**
      Forward port 443 to port 443 of your shell receiver system

    - **SSH over DNS:**
      Forward UDP port 53 to UDP port 53 of your shell receiver system

    - **SSH over ICMP:**
      Requires your shell receiver system to be directly connected to the Internet (no firewall).

    - **SSH over 3G:**
      Forward the port selected in the UI to port 22 of your shell receiver system

    - **SSH Egress Buster:**
      Forward all ports selected in the UI to port 22 of your shell receiver system

1. 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).
2. (Optional) Enable Stealth Mode
3. You can now deploy your Pwnie device to your target network. The device will automatically "phone home" to your shell receiver system, providing encrypted remote access to your target network.

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

Using SSH port forwarders on Backtrack

Example 1: Connecting to remote RDP servers

1. On Backtrack:

   # 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 device is physically connected to.

2. Login to the device when prompted.
3. Connect to the remote RDP server through the SSH tunnel by using "localhost":

   # rdesktop localhost

Example 2: Connecting to remote web servers

1. On Backtrack:

   # 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).

2. Login to your Pwnie device when prompted.
3. Open Firefox and configure it to use localhost as an HTTP proxy on port 8080.
4. You can now connect to any web server on the remote network by entering the IP address or URL into Firefox.

Using the wireless hardware

802.11 wireless

Ensure the wireless antenna is connected to your Pwnie device before continuing (see "Getting Started").

Connecting to an open wifi network

1. Set the wireless interface to managed mode:

   ```
   # iwconfig wlan0 mode managed
   ```

2. Bring up the interface:

   ```
   # ifconfig wlan0 up
   ```

3. Scan for access points in the area:

   ```
   # iwlist scan
   ```

4. Associate with an access point with SSID "example" on channel 6:

   ```
   # iwconfig wlan0 essid "example"
   # iwconfig wlan0 channel 6
   ```

5. Restart the interface:

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

6. Acquire a DHCP address:

   ```
   # dhclient wlan0
   ```

Running Airodump-ng & Kismet

1. Bring down the interface:

   ```
   # ifconfig wlan0 down
   ```

2. To launch airodump-ng:

   ```
   # airodump-ng wlan0
   ```

   Pwn Plug Note: The output of airodump-ng will only display properly within an SSH session (running airodump-ng from the serial console is not recommended).

3. When finished, press CTRL+C to exit
4. To launch Kismet:

```
# kismet
```

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

Tip: Certain wireless tools may leave the wireless adapter in a mode that's not compatible with other wireless tools. It's generally recommended to set the interface to a "down" state before running most wireless tools:

```
# ifconfig wlan0 down
```

Packet injection & WEP cracking

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 device.

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

2. Look for the following output:

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

3. To auto-crack all WEP-enabled access points on channel 6 using wepbuster:

```
# ifconfig wlan0 down
# wepbuster 6
```

Tip: WEP cracking performance is dependant on the amount of wireless client traffic being generated on the target wifi network. The more traffic on the wireless network, the faster the cracking process.

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

- **Pwn Plug R2:** Connect the SENA UD100 Bluetooth USB adapter to the device. The adapter will appear as Bluetooth device "hci1".
- **Pentesting Appliance:** By default, the Bluetooth USB adapter is already connected to the USB port behind the front bezel of the appliance. The adapter appears as Bluetooth device "hci0".

1. Confirm the output of the following commands:

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

# hciconfig hci1
hci1:   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
```

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

```
# hciconfig hci1 up
# hciconfig hci1 noscan
```

4. To scan for remote Bluetooth devices

```
# hcitool -i hci1 scan --flush --info --class
```

5. To ping the address of a remote Bluetooth device

```
# l2ping -i hci1 XX:XX:XX:XX:XX:XX
```

6. To dump Bluetooth packets:

```
# hcidump -i hci1 -t -X
```

7. To pair with a remote Bluetooth device:

```
# bluez-simple-agent hci1 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. Disconnecting the adapter while the interface is UP may cause system stability issues.

```
# hciconfig hci1 down
```

4G/GSM cellular

Using the unlocked GSM adapter

The unlocked GSM adapter supports five GSM cell bands (HSDPA / GSM / UMTS / EDGE / GPRS) and is compatible with AT&T, T-mobile, Vodafone, Orange, and GSM carriers in over 160 countries.

GSM carriers in the Americas:
http://en.wikipedia.org/wiki/List_of_mobile_network_operators_of_the_Americas

GSM carriers in Europe:
http://en.wikipedia.org/wiki/List_of_mobile_network_operators_of_Europe

Note: 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. In the US, SIM cards from AT&T and T-mobile devices (including iPhones) are supported.

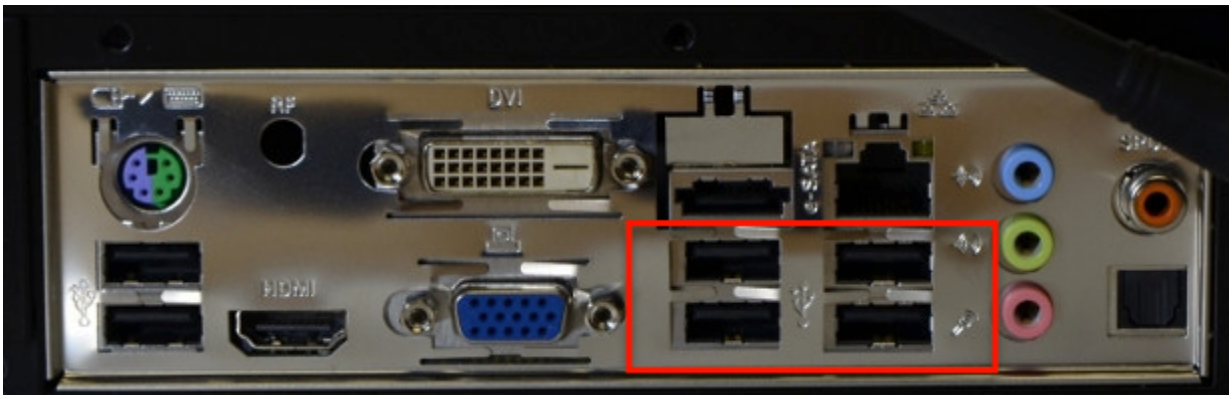
   Note: The mobile service attached to the SIM card must have mobile broadband data service. Verify you can access the Internet from your phone using the SIM card before proceeding.

2. Slide open the 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 the device's USB port.

   Pentesting Appliance: Connect the GSM adapter to one of the 4 rear USB ports as shown:



6. Confirm the GSM adapter is detected properly (note adapter detection may take 15-20 seconds):

   # lsusb
   Bus 003 Device 003: ID 12d1:1506 Huawei Technologies Co., Ltd. E398 LTE/UMTS/GSM Modem/Networkcard

7. To query the GSM modem for adapter details:

   # gsmctl -d /dev/ttyUSB0 me

   Note: If the command returns "SIM failure", the SIM card is either missing or not inserted properly.

   Tip: If the modem does not respond on /dev/ttyUSB0 after 10 seconds, try /dev/ttyUSB1, /dev/ttyUSB2, or /dev/ttyUSB3

8. To list cellular operators in range:

   # gsmctl -d /dev/ttyUSB0 op

9. To show currently attached operator:

   # gsmctl -d /dev/ttyUSB0 currop

10. To show signal strength of current operator connection:

    # gsmctl -d /dev/ttyUSB0 sig

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

    # gsmctl -d /dev/ttyUSB0 pin

12. To send a text message:

    # gsmsendsms -d /dev/ttyUSB0 *[destination 11-digit cell number]* "Test"

13. To make an outbound phone call:

    # gsmctl -d /dev/ttyUSB0 -o dial *[11-digit phone number]*

Connecting to the Internet via 3G/4G

1. Call the appropriate pppd dialup script:

   For the unlocked GSM adapter:

   ```
   # pppd nodetach call e160 &
   ```

   For the Verizon / Virgin Mobile adapters:

   ```
   # pppd nodetach call 1xevdo &
   ```

   For the T-mobile Rocket 4G adapter:

   ```
   # pppd nodetach call tmobile &
   ```

2. Assuming a 3G/4G cellular data signal is available, the adapter will establish an Internet connection within 10-20 seconds. Once connected, you will see a solid LED on the top of the adapter.
3. [Optional] Reset the default route to use the 3G/4G interface (ppp0):

   ```
   # route del default
   # route add default ppp0
   ```

4. Test 3G/4G Internet connectivity:

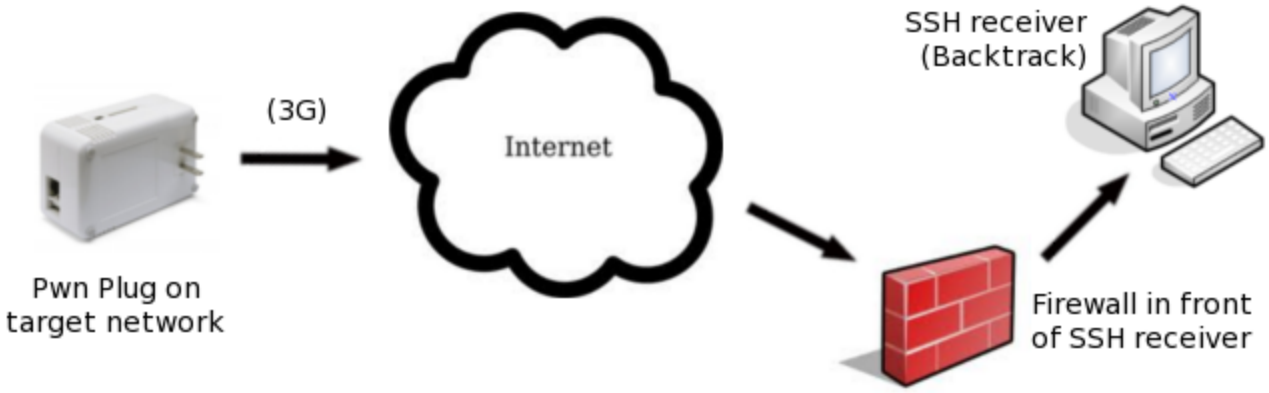
   ```
   # ping google.com
   # traceroute google.com
   ```

5. To close the 3G/4G connection and restore Internet connectivity on eth0:

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

Using the SSH-over-3G shell

The SSH-over-3G reverse shell provides secure, out-of-band access to your Pwnie device wherever a 3G cellular data signal is available. While this bypasses your target network's perimeter, a reverse shell is still recommended; many cell carriers do not assign public IP addresses to 3G data access devices.



1. If you haven't done so already, complete the reverse shell setup steps (see "Activating the reverse shells" and "Configuring the SSH receiver")
2. In the "Reverse Shells" page in UI, enable the "SSH over 3G/GSM" shell.
3. Configure the shell to connect to your firewall's public IP address (or DNS name if available).
4. Enter the destination port you'd like the Pwnie device to use for the SSH connection.
5. Select your 3G adapter from the drop-down list.
6. Click the "Configure all shells" button.
7. Configure your firewall to forward the port selected in the UI to port 22 on your Backtrack machine.
8. On the Backtrack SSH receiver, watch for the inbound SSH-over-3G connection:

   ```
   # watch -d "netstat -lntup4 | grep 3337"
   ```

9. Once the connection appears, connect to your Pwnie device as shown:

   ```
   # ssh pwnie@localhost -p 3337
   ```

10. Enter your Pwnie device SSH user password and voila! You're now remotely connected to the device through the reverse shell.. over 3G!

    Note: The 3G connection will be released and reconnected at the selected retry interval until a reverse SSH tunnel is established.

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.

Accessing Metasploit via msfrpcd

The msfrpcd service can be customized by editing the file /opt/pwnix/pwnix-config/services/msfprcd.conf:

```
# nano /opt/pwnix/pwnix-config/services/msfrpcd.conf
```

To restart the msfrpcd service:

```
# service pwnix_msfrpc restart
```

To access the msfrpcd service from a remote host, you will need a "front-end" application. We suggest using Backtrack for this. Log into Backtrack with your credentials, and run the following from the commandline:

```
# startx
# cd /opt/metasploit
# ./diagnostic_shell
# ./msf3/msfgui
```

Login to msfgui using the credentials specific in /opt/pwnix/pwnix-config/services/msfprcd.conf

> **Note:** You may have to open msfgui more than once to get it to prompt you for a server. By default it starts an msfrpcd instance on the local Backtrack system.

Running additional pentesting tools

Thanks to the rock stars at the Kali Linux project (kali.org), all below pentesting tools are pre-installed 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	wepbuster
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

- PTES: http://www.pentest-standard.org/index.php/PTES_Technical_Guidelines
- Metasploit Unleashed: http://www.offensive-security.com/metasploit-unleashed/Main_Page

Using advanced features

Stealth Mode

When enabled, stealth mode does the following:

- Disables IPv6 support (prevents noisy IPv6 broadcasting)
- Disables ICMP replies (won't respond to ping requests)
- Disables the UI (closes port 1443)
- Sets the local SSH server to listen on the loopback address only (closes port 22 to the outside)
- Still allows all reverse shells to function as expected

Important: Enabling stealth mode will prevent direct access to your Pwnie device's SSH server and Pwnix UI over the network. Once stealth mode is enabled, access to the device can only be obtained through a reverse shell, the local serial console (Pwn Plugs), or a locally-attached keyboard/monitor (Pentesting Appliance).

To enable Stealth Mode, run the following commands:

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

To disable Stealth Mode:

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

Tip: For additional stealthiness, run the following commands:

- If using DHCP, kill the dhclient process (closes listening UDP port 68):

  ```
  # killall dhclient
  ```

- Randomize your MAC address:

  ```
  # macchanger -r eth0
  ```

- Disable ARP replies (careful! this may affect network connectivity):

  ```
  # ifconfig eth0 -arp
  ```

NAC/802.1x Bypass (Pwn Plugs only)

NAC Bypass overview

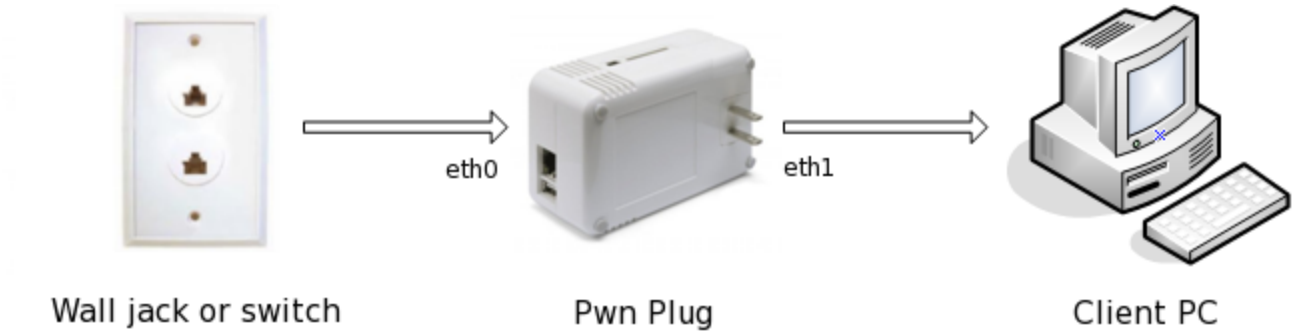
NAC Bypass can circumvent most wired NAC/802.1x/RADIUS implementations, providing a reverse shell backdoor and full connectivity to NAC-restricted networks.

Special thanks to Skip Duckwall and his 802.1x bridging research: http://8021xbridge.googlecode.com

Here's how it works:

1. First, the device is placed in-line between an 802.1x-enabled client PC and a wall jack or switch.
2. Using a modified layer 2 bridging module, the device transparently passes the 802.1x EAPOL authentication packets between the client PC and the switch.
3. Once the 802.1x authentication completes, the switch grants connectivity to the network.
4. The first outbound port 80 packet to leave the client PC provides the device with the PC's MAC/IP address and default gateway.
5. To avoid tripping the switch's port security, the device then establishes a reverse SSH connection using the MAC and IP address of the already authenticated client PC.
6. Once connected to the device's SSH console, you will have access to any internal subnets accessible by the client PC.

Tip: Since "NAC bypass mode" effectively turns the device into a transparent bridge, it can be used even where NAC/802.1x controls are not present on the target network.



Enabling NAC Bypass mode

Important: Enabling NAC Bypass mode will prevent direct access to your Pwnie device's SSH server and Pwnix UI over the network. Once stealth mode is enabled, access to the device can only be obtained through a reverse shell or the local serial console.

Note: These steps must be followed in the exact sequence shown to avoid tripping switch port security (which often completely disables the switch port and may alert network personnel).

1. Setup your desired reverse shells (see "Using the reverse shells").
2. Login to your device via SSH and run the following command:

   ```
   # service pwnix_nac_bypass start
   ```

4. Poweroff the device. At next boot, the device will be in NAC bypass mode.

   Note: After rebooting you will no longer be able to directly connect to the device via the Pwnix UI or SSH.

5. Deploy the device to your target environment as follows:

   - Connect the device to a power outlet.
   - Wait at least 30 seconds for the device to fully boot into NAC bypass mode.
   - Disconnect the client PC's Ethernet cable from the wall jack.
   - Connect the device's primary Ethernet port (eth0) to the Ethernet wall jack

     Note: eth0 is the right-most Ethernet jack on the rear of the unit

   - Immediately connect the Ethernet-over-USB adapter (eth1) to the client PC.

     Note: eth1 is the left-most Ethernet jack on the rear of the unit

6. The client completes its normal 802.1x authentication process transparently through the device.

7. When the first outbound HTTP port 80 packet leaves the client PC, the reverse shell connection schedule will re-initiate automatically.

NAC Bypass troubleshooting

1. Log into the device's serial console (see "Accessing the local serial console")
2. Confirm all outbound packets are tagged with the client PC's MAC and IP address:

```
# tcpdump -nnei eth0
```

4. Confirm 802.1x EAPOL authentication packets are being forwarded by the bridge:

On the Windows client PC:

a. Start the Wired Autoconfig service
b. Open the LAN connection properties / Authentication tab
c. Open "PEAP" settings
d. Uncheck the "Validate server certificate" checkbox and click OK
e. Click the "Additional settings" button
f. Check "specify authentication mode"
g. Select "user authentication" from the drop-down box
h. Click the "Replace credentials" button
i. username: testuser
j. password: testpasswd
k. Click OK, then OK again to close network connection setup
l. To generate EAPOL packets, restart the Wired Autoconfig service

On your Pwnie device:

a. tcpdump -nnei eth0 |egrep EAPOL
b. Look for outbound EAPOL packets. Example:

15:38:54.333292 00:0c:29:5c:74:41 > 01:80:c2:00:00:03, ethertype EAPOL (0x888e), length 60: EAPOL start

Tip: To manually force a link refresh from the command line: **mii-tool -r eth0 ; mii-tool -r eth1**

Disabling NAC Bypass mode

1. Log into the device through a reverse shell or the serial console (see "Accessing the serial console").
2. Run the following command:

```
# service pwnix_nac_bypass stop
```

3. Reboot

Setting up an encrypted partition (Pentesting Appliance only)

1. Run the following script to create an encrypted partition at /opt/pwnix/crypt-store

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

2. When prompted for key type, choose "passphrase".
3. Enter a desired passphrase for the encrypted partition
4. Answer "yes" when prompted (these prompts are expected when first encrypting a partition)
5. Copy a test file to the encrypted partition:

   ```
   # echo test > /opt/pwnix/crypt-store/test-file
   ```

6. Unmount the encrypted partition and very the contents of your test file is unreadable:

   ```
   # umount /opt/pwnix/crypt-store
   # cat /opt/pwnix/crypt-store/test-file
   ```

7. To re-mount the encrypted partition (enter your passphrase when prompted):

   ```
   # mount -t ecryptfs /opt/pwnix/crypt-store/ /opt/pwnix/crypt-store/
   ```

Maintaining your Pwnie device

Updating the Pwnix software

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".

Accessing the local serial console (Pwn Plugs only)

The serial console is useful for debugging or when a network connection is unavailable.

1. Connect the supplied mini-USB cable between your Pwnie device's mini-USB serial port and a Linux machine. On some older Linux kernels, the following commands may be required:

```
# modprobe usbserial
# modprobe ftdi_sio vendor=0x9e88 product=0x9e8f
```

Tip: For Windows/Mac systems see:
http://www.plugcomputer.org/Documentation/howtos/serial-terminal/

2. Connect to your Pwnie device serial console using screen (note on some distros this must be run as root):

```
# screen /dev/ttyUSB0 115200
```

Tip: If screen terminates after a few seconds, use "dmesg" to confirm your Pwnie device is showing up as a USB serial device. Example:

[15360.948161] usb 5-3: FTDI USB Serial Device converter now attached to ttyUSB0

If the serial interface is showing up as something other than "ttyUSB0" (such as ttyUSB1), adjust the "screen" command accordingly.

3. Press ENTER twice

Tip: If a login/command prompt does not appear, or if you see a line of question marks or strange-looking characters, try pressing CTRL+C several times or disconnecting/reconnecting the mini-USB serial cable.

4. At the login prompt, login with the pwnie user account. (default login is **pwnie : pwnplug8000**)

Tip: To exit a screen session, press CTRL+A, then backslash (\ ), then Y

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 get support

- Pwnie Express Support Portal: http://www.pwnieexpress.com/pages/support
- Pwnie Express Community Support Forum: http://forum.pwnieexpress.com