“Plug and Root,” the USB Key to the Kingdom

Darrin Barrall and David Dewey
SPI Dynamics
July 27, 2005
Who wouldn’t plug these in??
They Could Be Owning You

- Very little in the realm of USB security
  - OS level issues
    - Autorun
  - USB Protocol Enforcement
    - USB equivalent of raw sockets
Attack Vector

- Basically a hardware trojan
- Not the idea of walk-up and own (while that is a nice side effect)
Autorun

- By default, only works with non-removable media

- How to make a USB thumb drive “non-removable”
In-System Programming

• Many USB controllers allow for ISP

• Allows an attacker to “re-flash” the device with his own information

• Make the device tell Windows it’s a non-removable device
Here’s Why this Attack is Lame

• Attack is in user space
  – Yes, there are plenty of ways to escalate privileges, but it sure would be nice to not have to do them.

• Autorun must be enabled

• USB protocol is not enforced anywhere
  – Let’s target that.
Peripherals / VID + PID

- Many preconfigured USB controllers available on the market
  - Philips
  - Intel
  - Etc.
- SL811 – Allows for the configuration of all pieces of the USB pie – the proverbial raw socket
Host

• USB is like TCP
  – Built on a state machine
  – Believes that it will get what it wants
Windows Expecting Us to Be Nice
Windows Expecting Us to Be Nice (Cont’d)
POOF!!

---

**Command:** Kernel 'com:port=com1_baud=115200', WinDbg E.4.0007.0

*** Fatal System Error: 0x00000050 (0x435B7000, 0x000000001, 0x1B21050, 0x000000001)

**Driver at fault:**

---

A fatal system error has occurred.
Debugger entered on first try. Bugcheck callbacks have not been invoked.
A fatal system error has occurred.
Connected to Windows XP 2600 x86 compatible target, ptr64 FALSE
Loading Kernel Symbols
The Rest is Up to You

- Heap Overflow

- Who’s up for the challenge??
Power Up

- USB gives us ~5V
- Blowing the USB power supply could be fun – but a little lame
Throw the Switch

- USB does not require the physical removal of a device for it to be “removed”

- This allows a device to be “inserted” and “removed” as needed
Faces

- SL811 does not store the descriptors internally
- This allows the chip to appear to be ANY device supported by the OS
- This allows the device to enter and execute portions of drivers that are not thoroughly field tested
Emulation

- Emulating other devices
- Device drivers are typically written with a lot of trust
- Our emulating device will exploit that trust relationship
Writable Read-Only Devices

- Host-side code makes a request to read an address from the “read-only” device
- The meta-device returns garbage data
- The host is happy thinking it just read data
- The address requested is the four bytes of data recorded by the meta-device
Empty the “Trash”

- Hand one to your janitor and $20
Class

- Class drivers allow multiple vendors to create similar devices without the need for individual drivers
- Allows for a broad attack against the class driver
Patched??

• Say the driver you’ve been exploiting eventually gets patched

• VID++; //Need I say more??
Meta-Hub

- Hubs are so different, they have their own section in the USB specs
- Many more attack vectors
- Possible BlackHat 2006 speech??
- See you then!
Defense
Epoxy the USB Port Shut

Just kidding
Software Solution

- [http://www.safend.com/](http://www.safend.com/)
- Requires the client to be installed on every machine
- Tell the software that you are a device that is allowed to be there
- No USB protocol enforcement??
Nice Idea

• Software solution to enforce USB protocol and disable Autorun
Hardware

- Nice theory

- In-line USB device that would perform protocol enforcement to perform all the validation the OS should do
References

- Toaster Oven Reflow:
  - http://www.seattlerobotics.org/encoder/200006/oven_art.htm
- Parts:
  - http://www.digikey.com
- All Things USB:
  - http://www.usb.org/
- All Things USB 1.1:
  - http://www.usb.org/usb1.1spec
- SL811 Datasheet:
- Useful Pages:
  - http://www.beyondlogic.org/usbnutshell/usb1.htm
  - http://usbdeveloper.com/
QUESTIONS?

Darrin Barrall and David Dewey
SPI Dynamics