

“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??



Black Hat Briefings

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

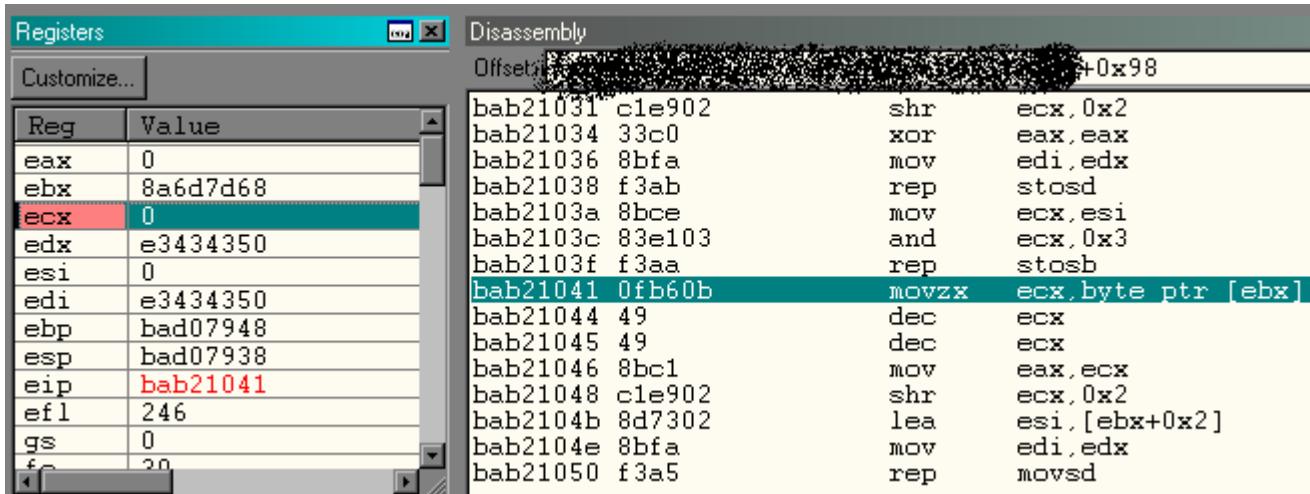


Host

- USB is like TCP
 - Built on a state machine
 - Believes that it will get what it wants



Windows Expecting Us to Be Nice



The screenshot shows a debugger interface with two main panes: 'Registers' and 'Disassembly'.
Registers:

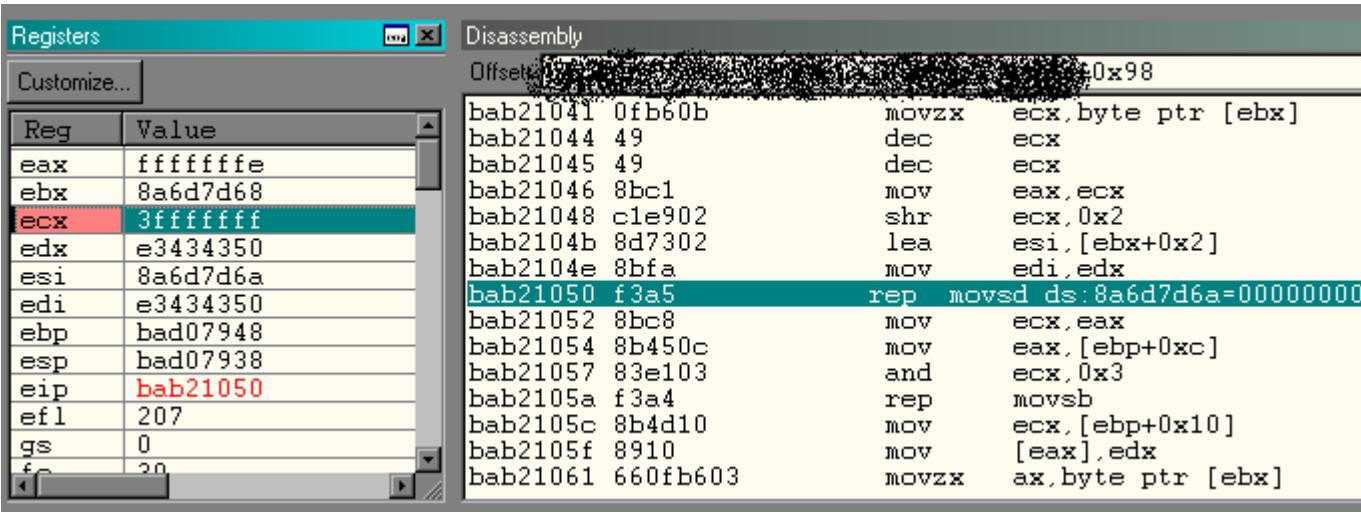
Reg	Value
eax	0
ebx	8a6d7d68
ecx	0
edx	e3434350
esi	0
edi	e3434350
ebp	bad07948
esp	bad07938
eip	bab21041
efl	246
gs	0
fs	20

Disassembly:

```
Offset: bab21031 c1e902 shr    ecx,0x2
bab21034 33c0 xor    eax,eax
bab21036 8bfa mov    edi,edx
bab21038 f3ab rep    stosd
bab2103a 8bc e     mov    ecx,esi
bab2103c 83e103 and   ecx,0x3
bab2103f f3aa rep    stosb
bab21041 0fb60b movzx  ecx,byte ptr [ebx]
bab21044 49 dec    ecx
bab21045 49 dec    ecx
bab21046 8bc1 mov    eax,ecx
bab21048 c1e902 shr    ecx,0x2
bab2104b 8d7302 lea    esi,[ebx+0x2]
bab2104e 8bfa mov    edi,edx
bab21050 f3a5 rep    movsd
```



Windows Expecting Us to Be Nice (Cont'd)



The screenshot shows a debugger interface with two main panes: 'Registers' and 'Disassembly'.
Registers Pane:

Reg	Value
eax	ffffffffff
ebx	8a6d7d68
ecx	3fffffff
edx	e3434350
esi	8a6d7d6a
edi	e3434350
ebp	bad07948
esp	bad07938
eip	bab21050
efl	207
gs	0
fs	30

Disassembly Pane:

Offset	Instruction	Description
bab21041	0fb60b	movzx ecx, byte ptr [ebx]
bab21044	49	dec ecx
bab21045	49	dec ecx
bab21046	8bc1	mov eax, ecx
bab21048	c1e902	shr ecx, 0x2
bab2104b	8d7302	lea esi, [ebx+0x2]
bab2104e	8bfa	mov edi, edx
bab21050	f3a5	rep movsd ds:8a6d7d6a=00000000
bab21052	8bc8	mov ecx, eax
bab21054	8b450c	mov eax, [ebp+0xc]
bab21057	83e103	and ecx, 0x3
bab2105a	f3a4	rep movsb
bab2105c	8b4d10	mov ecx, [ebp+0x10]
bab2105f	8910	mov [eax], edx
bab21061	660fb603	movzx ax, byte ptr [ebx]



POOF!!

Registers

Reg	Value
eax	3
ebx	0
ecx	bad077bc
edx	5e
esi	0
edi	e35b7000
ebp	bad07444
esp	bad073fc
eip	8052a5d8
efl	246
gs	0
fs	30
es	23
ds	23

Disassembly

Offset	Address	OpCode	OpName	OpDesc
				+0x98
		bab2103c	83e103	and ecx,0x3
		bab2103f	f3aa	rep stosb
		bab21041	0fb60b	movzx ecx,byte ptr [ebx]
		bab21044	49	dec ecx
		bab21045	49	dec ecx
		bab21046	8bc1	mov eax,ecx
		bab21048	c1e902	shr ecx,0x2
		bab2104b	8d7302	lea esi,[ebx+0x2]
		bab2104e	8bfa	mov edi,edx
		bab21050	f3a5	rep movsd
		bab21052	8bc8	mov ecx,eax
		bab21054	8b450c	mov eax,[ebp+0xc]
		bab21057	83e103	and ecx,0x3
		bab2105a	f3a4	rep movsb
		bab2105c	8b4d10	mov ecx,[ebp+0x10]
		bab2105f	8910	mov [eax],edx
		bab21061	660fb603	movzx ax,byte ptr [ebx]
		bab21065	668901	mov [ecx],ax

Command - Kernel 'com:port=com1,baud=115200' - WinDbg:6.4.0007.0

*** Fatal System Error: 0x000000050
(0xE35B7000,0x00000001,0xBAB21050,0x00000001)

Driver at fault:
*** - Address BAB21050 base at BAB18000, DateStamp 41107d68

Break instruction exception - code 80000003 (first chance)

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??

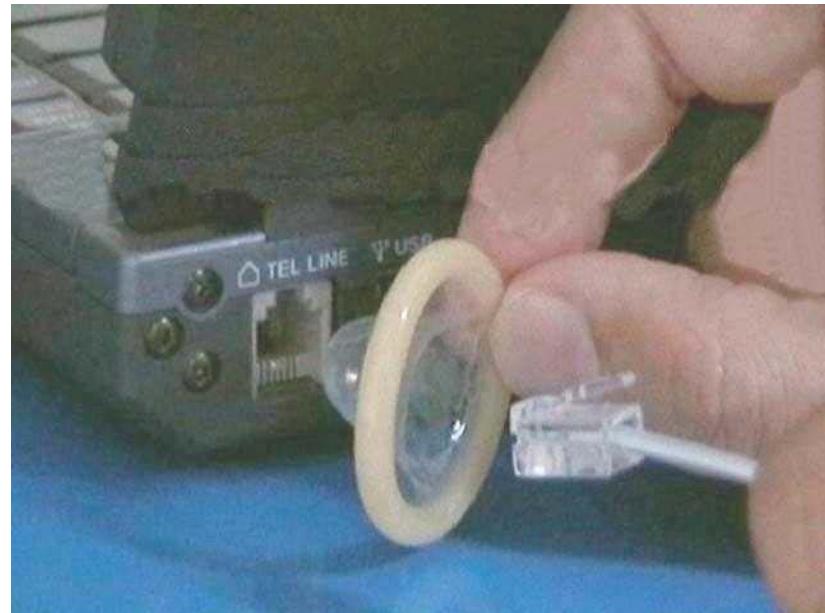


Meta-Hub

- 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



Black Hat Briefings

Epoxy the USB Port Shut



Just kidding



Software Solution

- <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:
 - <http://www.cypress.com/portal/server.pt?space=CommunityPage&control=SetCommunity&CommunityID=209&PageID=259&fid=10&rpn=SL811HS>
- Useful Pages:
 - <http://www.beyondlogic.org/usbnutshell/usb1.htm>
 - <http://usbdeveloper.com/>



QUESTIONS?

Darrin Barrall and David Dewey
SPI Dynamics

