Modern Game Console Exploitation

Eric DeBusschere, Mike McCambridge

March 26, 2012

Console Security Overview

- Most modern consoles acheive complete software security by only running signed code, encrypting memory, and utilizing firmware updates to patch vulnerabilities.
- Although hardware security is more expensive, modern consoles use secure boot as well as a chain of trust seeded by a unique hardware console key.
- The Xbox 360 and PS3 go further:
    - The PS3 utilizes isolated SPUs
    - The Xbox 360 uses hardware eFuses to prevent downgrading

Console Security Overview - Internet Protection

- The best modern protection is via the internet and updates:
  - Firmware Updates
  - Internet Banning
  - Requiring users to have an updated console to play new games
- These practices force users to choose between an exploited console and a **complete gaming experience**

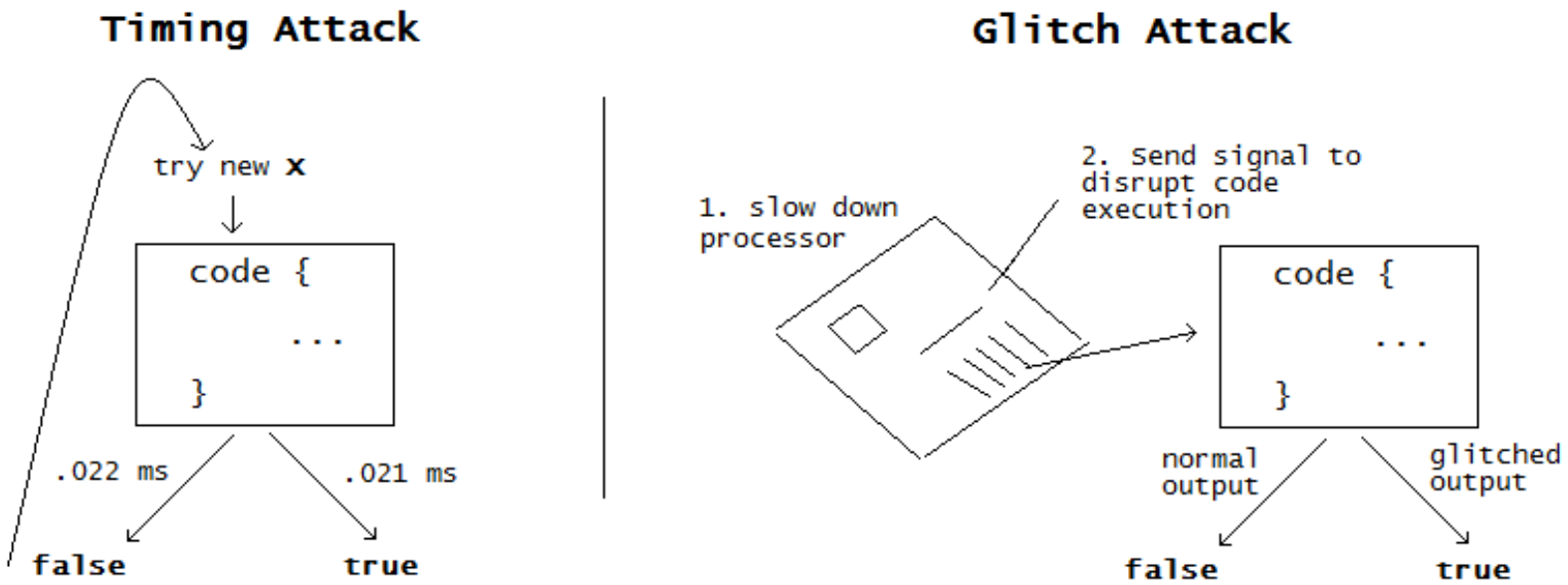
Console Security Overview - Graphic

Modern Console Security Practices

	360	PS3	Wii	Xbox
Per Console Key	●	●	●	
Manufacturer Private Key	●	●	●	●
Encrypted Memory	●	●		
Firmware Updates	●	●	●	
Secure Boot	●	●	●	●
Internet Banning	●	●		
eFuses	●			
Hypervisor	●	●		
Signed Executables	●	●		●
Security Coprocessor	●			
Encrypted Hard Disk		●	●	

Exploitation Strategies Overview

- Software exploitation is generally done through memory overflows or unchecked parameters
- Hardware attacks usually use either a timing or glitch attack as the entry point



Exploitation Strategies Overview

- Attacks build off of each other. Examples:
  - Using a Glitch attack to dump the hypervisor and expose software bugs.
  - Using a timing attack to downgrade to an exploitable kernel.
- After a software vulnerability is discovered and patched, **downgrading** becomes a viable exploitation strategy.

XBOX 360 - Software Security

- Operating System only runs signed code.
- No unencrypted, executable code is written to memory.
    - Prevents memory snooping
- All vulnerabilities are patched whenever a console connects to XBOX Live and downloads the latest update.
- All new console's are sold with latest security updates.

XBOX 360 - Hardware Security

- Due to cost, Xbox 360 is more vulnerable to hardware attacks.
- 360 contains 768 bits of eFuse, a technology by IBM
    - Main Purpose is to prevent downgrading by flashing older kernel
    - Blown whenever a kernel update is performed in clusters of 8 (hex val)
    - Value use along with CPU key to sign and verify firmware software
- Tightly controlled boot process

XBOX 360 - Fusesets

Fuseset	**Purpose**
00 and 01	Retail or Dev console
02	Lockdown counter for 2BL/CB-the bootloader.
03-06	Defines the CPU key, set at factory
07-12	Lockdown counter for 4BL (Kernel)

XBOX 360 - Secure Boot

Bootloader	**Purpose**
1BL	Reads the 2BL code from NAND-Flash and decrypts it into the CPUs SRAM.
2BL	Verifies itself with eFuse. Initializes PCI-Bridge, disables JTAG test port, and initializes memory encryption. Decrypts the 4BL into memory.
4BL	Checks and unpacks the 5BL, applys update patches. Determines update sequence from eFuses.
5BL	Merges the Hypervisor and Kernel into a single image.
6BL/7BL	Updates kernel from base kernel using delta compression.

Custom Firmware for DVD-ROM

- Almost immediately, hackers discovered that simply writing custom firmware for the DVD-ROM drives used in Xbox 360s allowed them to play copied games
- The content of the disc must be identical, otherwise the signature will not remain intact.
- Not very interesting to users interested in running Linux.

King Kong Exploit - Kernel 4532/4538

- Games, specifically King Kong, can write the results of pixel shaders directly into memory
- These writes are checked by the Hypervisor, but Kernel 4532/4538 contained a critical error which allowed the upper 32 bits of memory to be set through a pixel shader.
- In the code below, the input address was (inadvertently) cast to 32 bits to check, but used in its full 64 bit form in execution.

Code:

```
syscall ( uint64 inputAddress) {
        if ((uint32)inputAddress > checkVal)
                illegal call;
        ....
}
```

King Kong Exploit - Using Hardware Only

- KK Attack requires a user start their console using the King Kong disc everytime they want to enter the exploited state.
- Because the attack is a DMA, in theory any software/hardware that has authorization to perform a DMA could be used to trigger it.
- It did not take long for hackers to discover a purely hardware based attack.

King Kong Exploit - JTAG

- A hardware group called the JTAG point was reverse engineered
- This allowed hackers to set DMA Addresses
- The JTAG could not be used to trigger a DMA because it is disabled early in the boot process

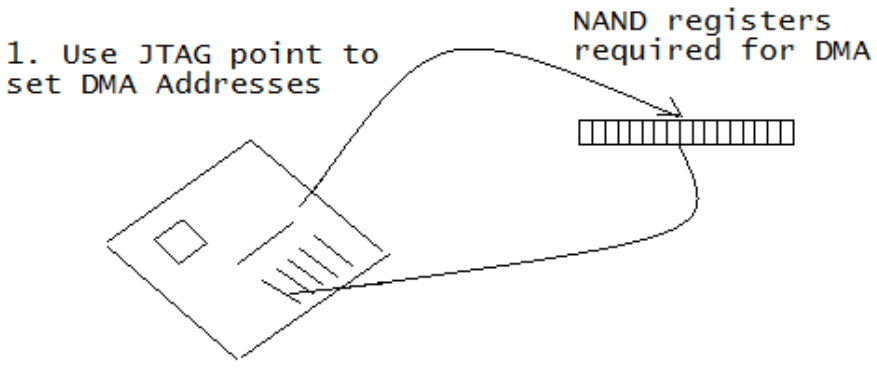


Figure 1: Setting DMA Target Address through JTAG point.

King Kong Exploit - SMC

- The SMC port could launch a DMA, but could not set the target DMA addresses
- Together, however, the JTAG/SMC could trigger a controlled DMA

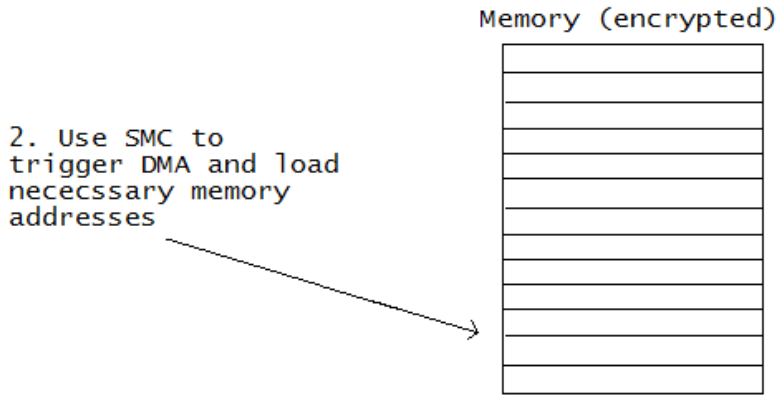


Figure 2: Using SMC to trigger DMA.

King Kong Exploit - Hardware Triggered DMA

- The DMA loads the neccessary memory addresses and initiates the attack.
- The two neccessary memory jumps are performed, and the exploit is complete.

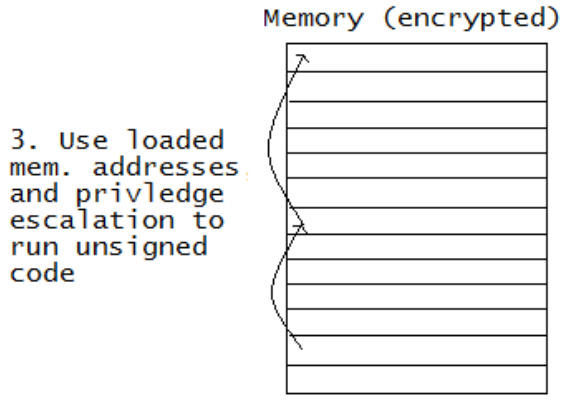
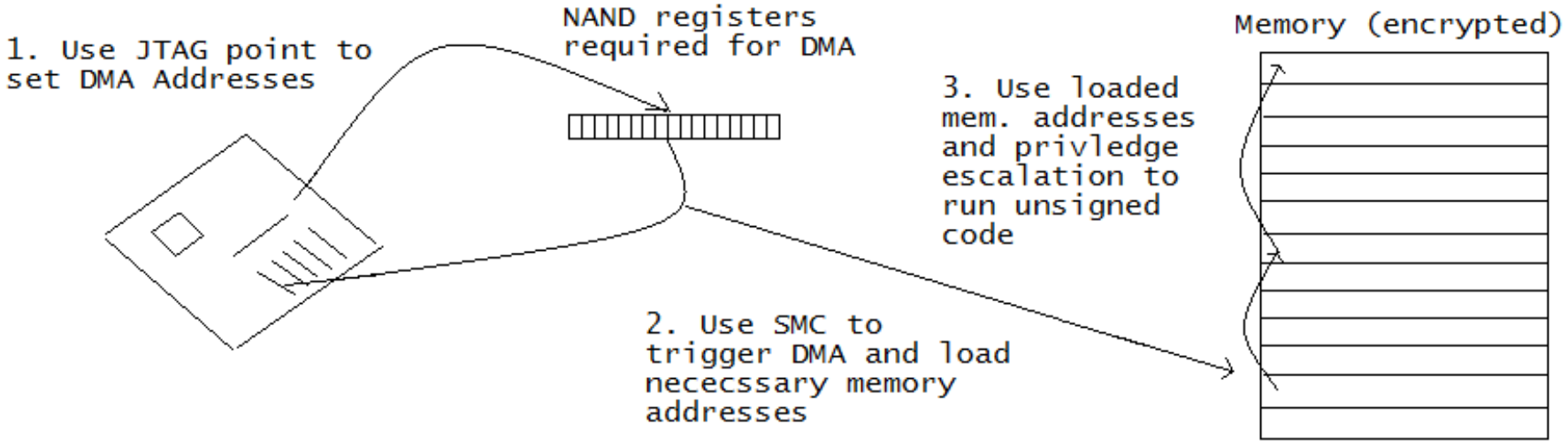


Figure 3: The exploit allows jumping into any 32-bit address in the hypervisor's address space.

King Kong Exploit - Full Hardware Exploit



Zero Pairing and MfgBootLauncher

- In Summer 07, MfgBootLauncher mode was discovered.
- The 2BL has an information header which includes a pairing block
- When this pairing block is all 0s, MfgBootLauncher mode is entered
- MfgBootLauncher does nothing, but Microsoft didn't like it so they made several changes via a firmware update:
    - Decrypting the 4BL now requires the CPU-Key
    - MfgBootLauncher mode allows a user to **bypass the eFuses**

Chicken and Egg Scenario

- This presented a chicken and egg scenario for Xbox 360 hackers
  - If you know your CPU Key, you can downgrade to an exploitable kernel
  - But to get your CPU key, you need to run an exploitable kernel

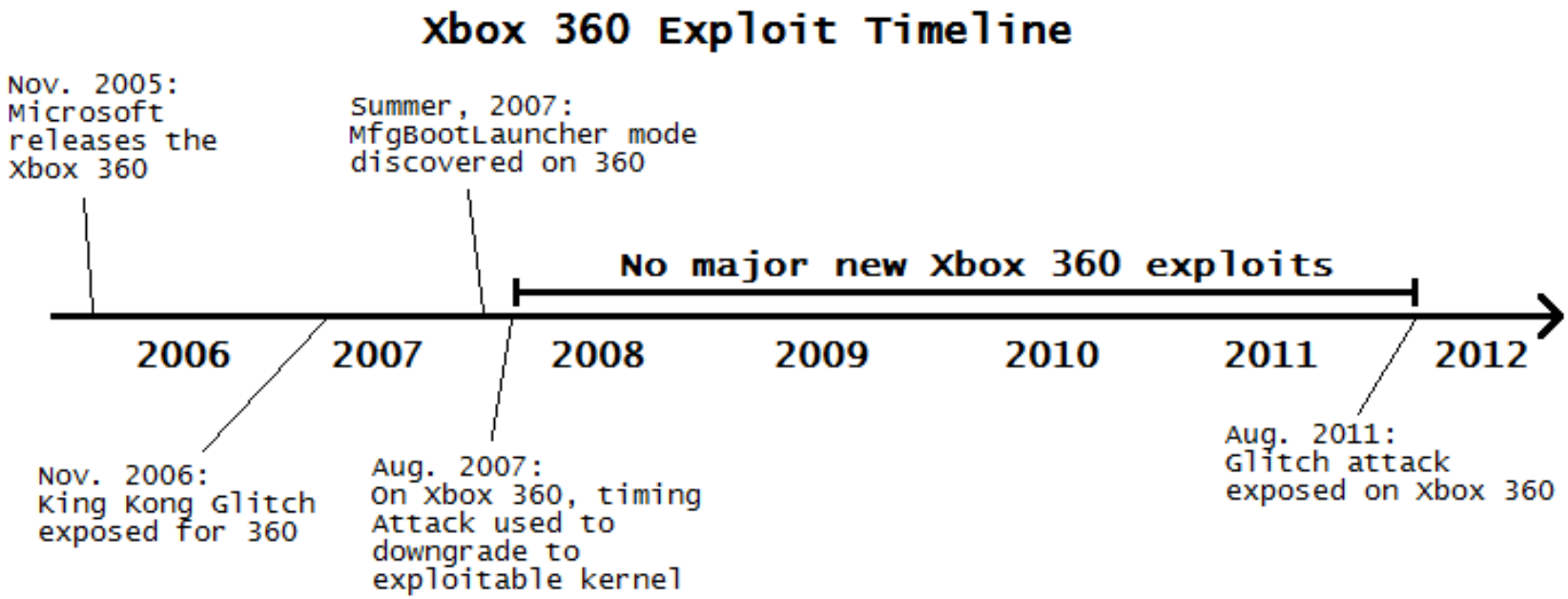
Figure 4: The chicken and egg in the scenario

Xbox 360 - Timing Attack

- Hackers took an unmodified base kernel and patched it with values in the current kernel.
- Now, they only had to get it past the 4BL Hash Check, which was much easier than finding a console's CPU Key.
- Essentially, they **shifted** the problem of finding a console's CPU-Key to the easier problem of getting an unsigned kernel past the Hash Check.
- The Hash Check was done using a memcmp function over a 16 byte value.
    - A difference of 2200 microseconds was found between True and False Values
    - Queue timing attack!

Xbox 360 - Unexploitable

- After the timing attack, the Xbox 360 went three years without a major exploit
  - A small exploit allowed consoles up to Summer 09 to run timing attack



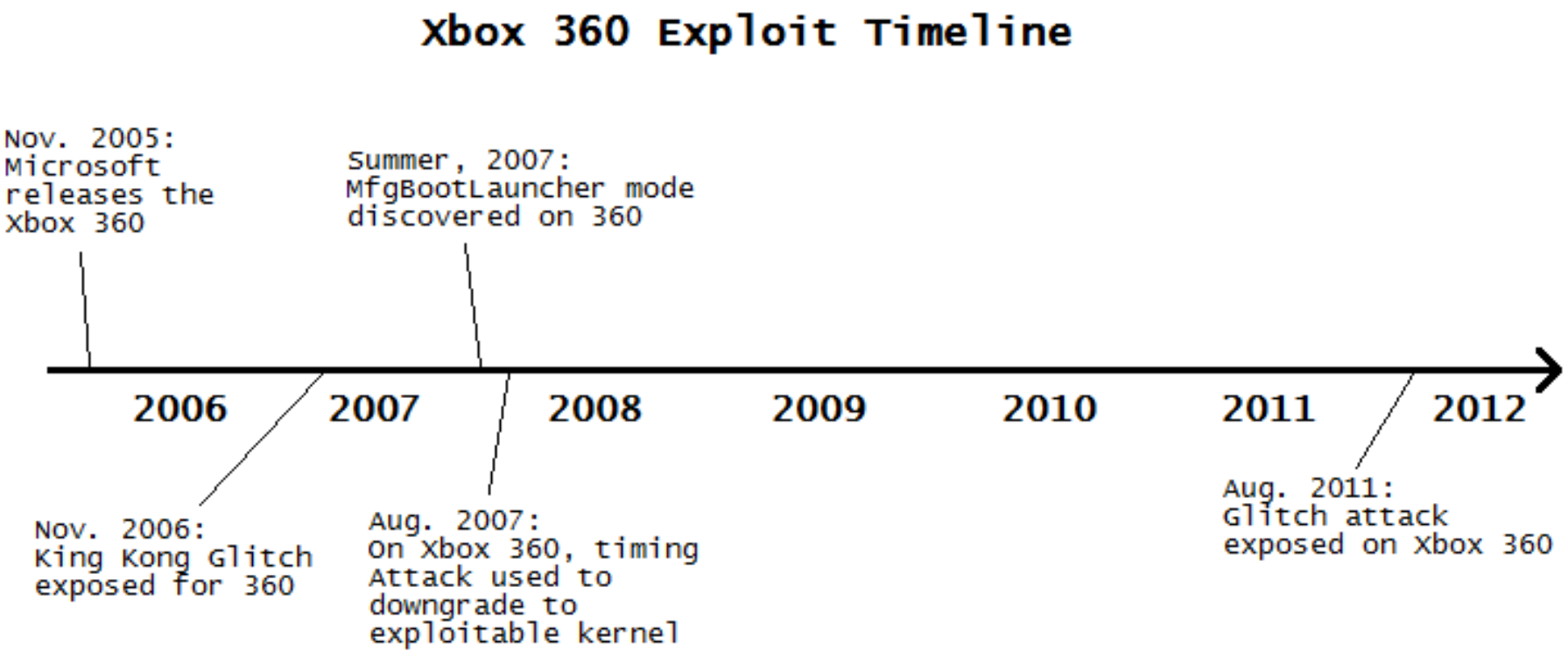
Xbox 360 - Reset Glitch

- Most in the community thought a glitch attack would be too difficult/expensive
- One hacker, GliGli, became desparate enough to try, and succeeded!
- Fat consoles are easy to glitch
    - Asserting CPU-PLL-BYPASS signal slows CPU execution down 128x.

Xbox 360 - Reset Glitch on Slims

- The CPU-PLL-BYPASS could not be asserted on Slims
- Slim consoles have their CB split into two: CB-A and CB-B
- Asserting CPU-RESET on the HANA chip allowed a patched CB-B to sidestep validation
- Attackers derived a patched CB-B by **building off of** exploited Fat console CBs.
    - Slim CB-B is protected with RC4 Stream encryption
    - Hackers guessed that the first few bytes would be the same
    - Dumped CPU key and signed patched CB-B

Xbox 360 - Complete Timeline

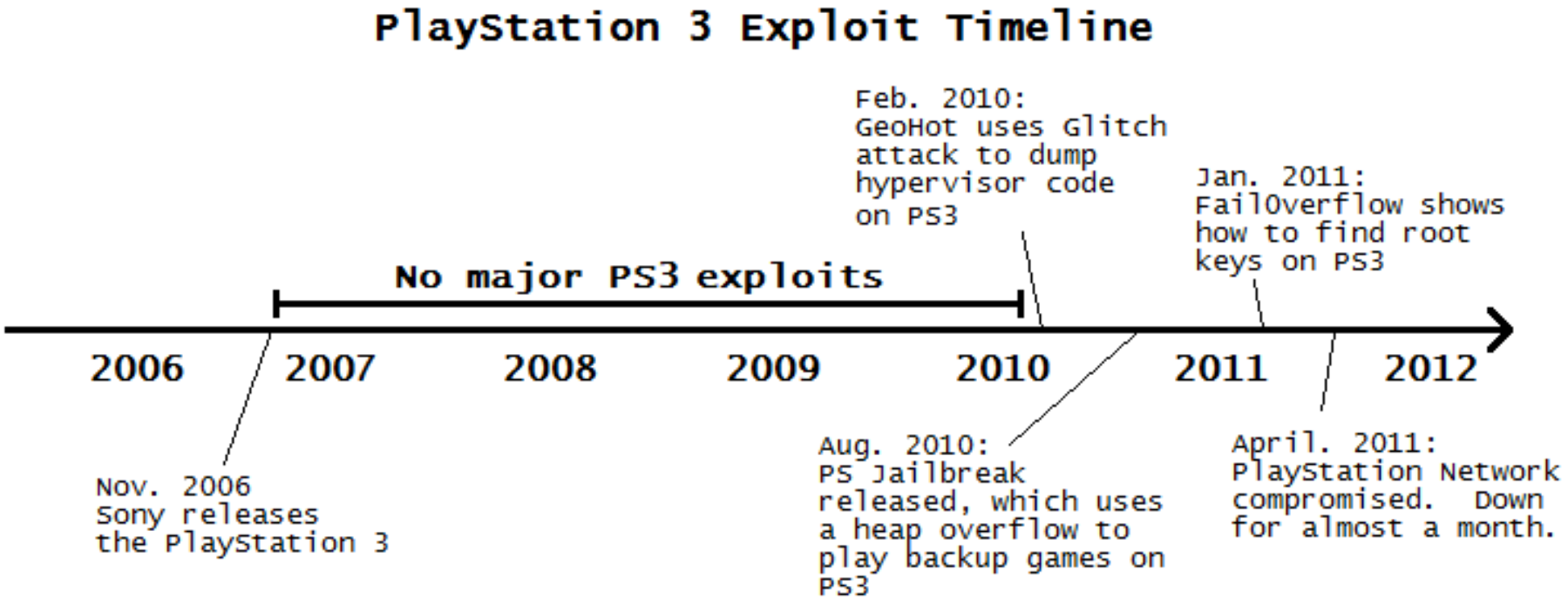


PlayStation 3 - Security

- Runs a layered Operating System: Kernel mode, User Mode, Hypervisor Mode
- Software security is similar to the Xbox 360
    - Signed Executables
    - Encrypted Memory
    - Firmware Updates
    - New Games require the latest updates
- Hardware Security Differences
    - Does not use eFuses - Downgrading is easier than 360
    - Runs isolated SPEs: Synergistic Processing Elements

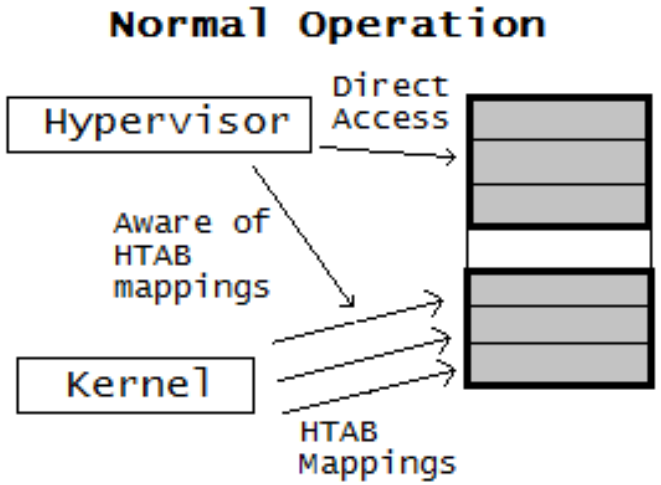
PlayStation 3 - Security Intact

- The PlayStation 3 remained completely unexploited for more than three years
- Speculators believe this was because Sony supported Linux through OtherOS until 2010



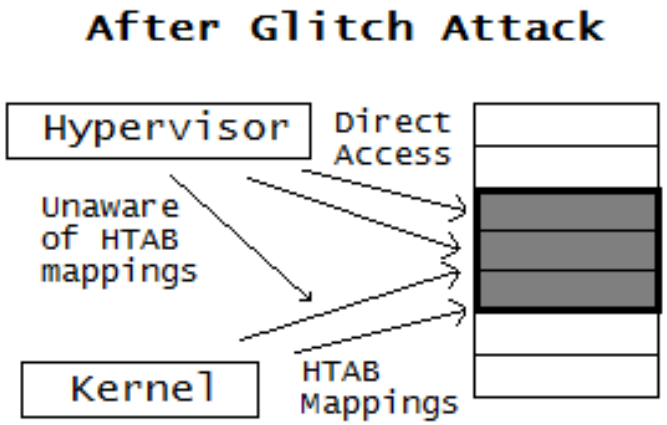
GeoHot Dumps Hypervisor

- GeoHot, a 21 year old hacker famous for his iPhone exploits, initiated the first successful attack against the PS3 in early 2010.
- The entry point is OtherOS, Sony's tool to run Linux
- OtherOS has to ask the Hypervisor for page table mappings to access memory.



GeoHot Dumps Hypervisor

- GeoHot had OtherOS allocate a sizable buffer and ask for a large number of mappings.
- Then, the buffer was deallocated without properly removing the mappings.
- Normally, the hypervisor would just deallocate all the mappings.
- However, glitching the memory bus as these deallocations occured caused some mappings to remain intact, allowing GeoHot direct memory access.

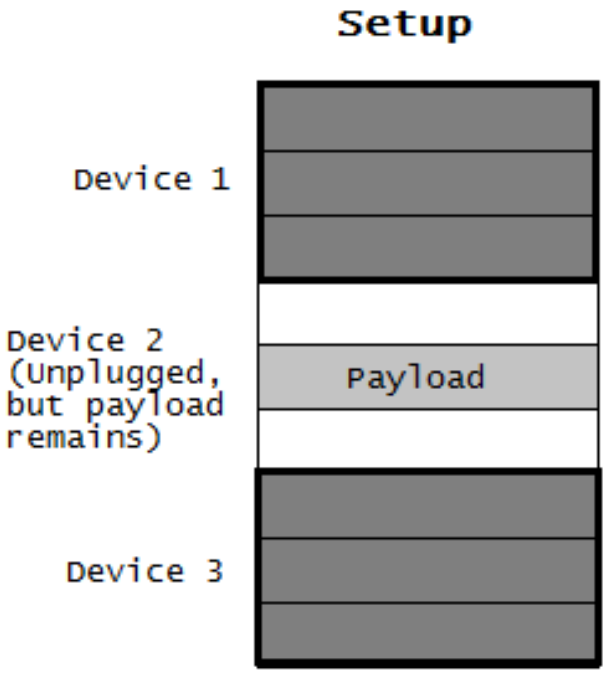


PS Jailbreak

- The first exploit that allowed a user to play backup games stored on the hard drive
- Utilizes a USB device verification bug to trigger a heap overflow and inject unsigned code
- Only compromises the lowest level of the Operating System, and allows a user to play pirated games but not run Linux.
- Initially made available for $100 to $130, though it was quickly reverse engineered.

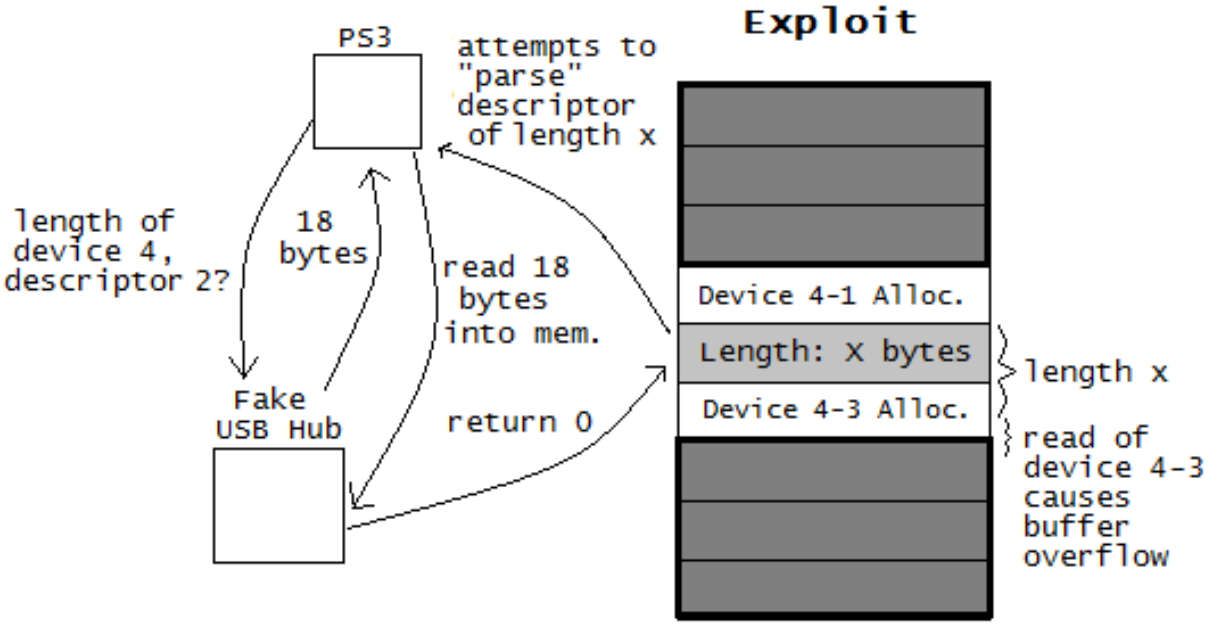
PS Jailbreak - Step 1

- The exploit masquerades as a USB Hub
- First, it “plugs in” three USB devices with large descriptors
- Then, device 2 is unplugged, but its payload remains.



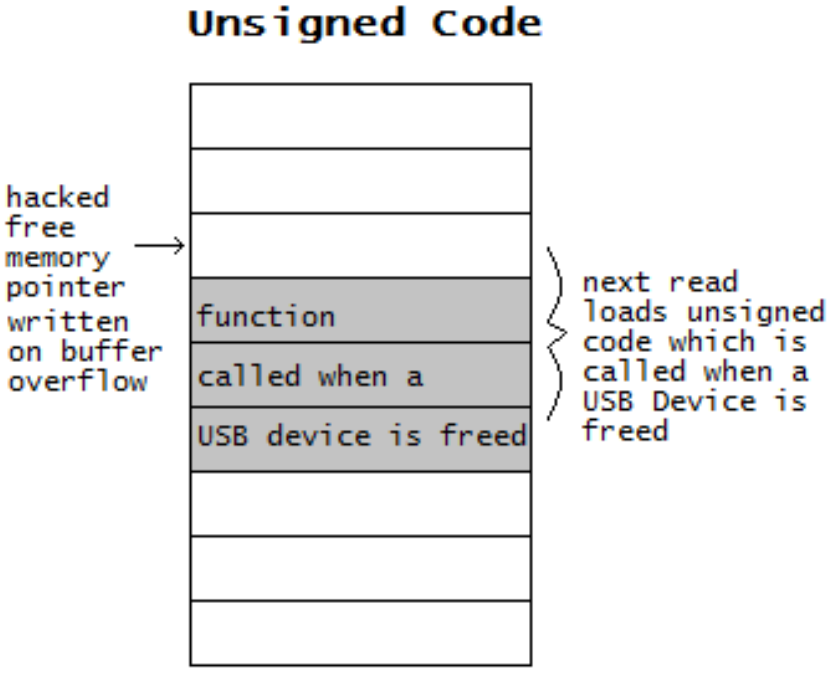
PS Jailbreak - Step 2

- A device containing 3 descriptors is plugged into port 4
- The PS3 reads the size of each descriptor in order to allocate the neccessary memory
    - The allocated memory is between Device 1 and Device 3's descriptors.
- When Device 4's descriptors are actually read, the exploit changes the size of descriptor 2 to 0 bytes
- This causes the PS3 to parse the payload injected in step 1, which contains a descriptor much larger than the space allocated, overflowing the buffer.



PS Jailbreak - Step 3

- The overflow overwrites the malloc boundary tag, and points it to a function called after USB devices are freed.
- Next, the exploit plugs a device into port 5, posing as an official PS3 service Jig
- The PS3 sends a challenge, and the device responds with static data which is stored in memory, thus overwriting the function called after USB devices are freed.
- Then, the exploit simply unplugs one of the devices, and the unsigned code is called!

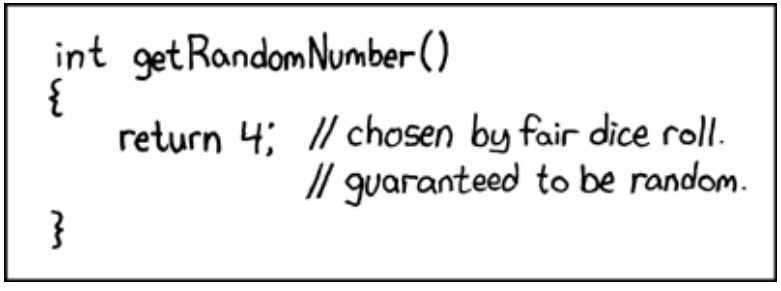


Fail0verflow

- On Jan. 7th, 2011, team Fail0verflow, building off of GeoHot's dump of the Hypervisor and the PS Jailbreak exploit, revealed several astonishing security flaws in the PlayStation 3.
- Most code running on the PS3 is in the common ELF format, but is signed with secure keys and only decrypted in an isolated SPE.
- However, the SPEs do not verify that the Kernel is uncompromised so a low level exploit can simply ask an SPE to decrypt anything it wants!

Fail0verflow Continued

- Even more surprising was the next security flaw they discovered
- The PS3 uses ECDSA to sign all its executables.
- To be secure, ECDSA requires a random number each time a signature is generated
- Unfortunately, Sony used the same random number every time, and discovering the root private keys was trivial.



PlayStation 3 Exploit Status

- Essentially, the PlayStation 3 is completely compromised.
- However, everyone involved got sued by Sony.
    - Fail0verflow and GeoHot are no longer involved in the scene.
- New obsfucation techniques by Sony have made it more difficult to exploit the latest kernel versions.
- Still, all PS3s sold through the end of 2011 are vulnerable, and most new kernel versions can be downgraded to an exploitable variety.

Anonymous Becomes Angry

- Sony's treatment of GeoHot and FailOverflow, as well as its victory in a lawsuit surrounding its decision to end Linux support angered the hacktivist group Anonymous.
- They published a list of demands:

1. `Sony must allow for end-user modification of the PS3, as was available before disabling Linux`
2. `Sony must end any attempts to bring legal action to alter a product they own.`
3. `Sony must not pursue legal action against any collected IP address.`

PSN Network Down

- Shortly following Anonymous's list of demands, a crippling assault was launched against the PlayStation Network
  - It was brought down for 23 days.
  - Credit card information of 77 million users was revealed to the attackers
  - Recovery costs were close to $171 million for Sony
- Anonymous claims "For Once We Didn't Do it."
- In all likelihood, several Anon members acted alone, utilizing knowledge about Sony's security flaws gathered in commonly used communication channels.