Davis Pinkley-Adams

6/22/23

Week 1 MBE lectures

Lecture 08: Secure Systems and Pwning Popular Platforms

This lecture first talks about Secure Systems and Patch Sets

A couple of the categories include:

* OpenBSD
  + Free, multi-platform 4.4BSD-based UNIX-like OS
  + “Try to be the #1 most secure OS”
  + Hardening & Security Tech
    - W^X
    - Priv isolation
    - Jails
    - Rand malloc/mmap
    - Ships crypto
  + Extensive code review and audits
  + Default install disables most remote services.
* SELinux
  + Implementation of mandatory access controls on Linux
  + Users and services only have access to exactly what they need.
* Grsecurity
  + Security enhancement to the Linux kernel that defends against security threats through intelligent access control, memory-based exploit prevention, and other system hardening
  + Large focus on hardening against memory corruption based exploits.
    - High quality PAX ASLR, Memory Sanitization, Heap Hardening, and Active Response

The second part of the lecture slides talk about security flaws in an older generation of consoles that include the Xbox 360, PS3, and the Nintendo 3DS.

* Xbox 360(Nov. 2005)
  + King Kong Exploit
    - Int based bug that caused code execution at the Hypervisor context.
  + All executables(.XEX) are signed by Microsoft but data assets such as textures, models, shaders, and audio are not.
  + Hypervisor sits at the top of the memory next to the Kernel and if you can take it over you have access to physical memory which has the highest privilege of execution.
  + It seems like one of the biggest flaws is that only the lower 32 bits of the syscall are sanity checked, but the whole 64 bit number is used in address calculation.
* Nintendo 3DS(Feb. 2011)
  + App Processor(ARM11) – runs games, apps, and visuals
  + Security Processor(ARM9) – crypto, system IO, talks to hardware, like a Hv
  + PXI – pipeline for the cores to talk to each other
  + VerifyRsaSha256()
    - Stack smash bug that results in code execution on ARM9
    - Only present in versions 1.0.0 – 4.5.0
    - A bug that uses memcpy with user controlled data, and a user specified size
* PlayStation 3(Nov. 2006)
  + GeoHot through OtherOS able to own the hypervisor
  + PS3 Jailbreak
    - With the dump of the PS3 Kernel (LV2), heap overflow was found to be in the USB Handling during startup.
    - Main bug in this overflow was a long device descriptor that leads to meory corruption on the heap.
  + ECDSA Key Extraction
    - Elliptical Curve Digital Signature Algorithm
    - Need control of LV2 so you can make crypto requests to the security SPE
    - 
    - When m is identical for two signatures, so is R, and
    - 
    - Two signatures from the Crypto SPE are needed to compute the private Sony key.
  + Much occurred after this and sony had to make many changes to the way their software worked.
  + In the end the PS3 was totally broken and their was nothing Sony could do to prevent it.
* The current generation of consoles(last generation by today’s standards) are more secure than those before them. It would be fun to look into the security of consoles that have previously come out like that of the Nintendo Switch, PS5, and Xbox X.

Lecture 09: Address Space Layout Random Randomization

* Memory segments are no longer in static address ranges, but instead unique for every execution.
* Stack smashing can get you control of the EIP but how do you know where you need to go?
* How do you check for ASLR?
  + Cat /proc/sys/kernel/randomize\_va\_space
    - 0: No ASLR
    - 1: Conservative Randomization
      * (Stack, Heap, Shared libs, PIE mmap(), VDRO)
    - Full Randomization
      * Conservative Randomization + memory managed via brk()
* Not everything is random on linux
* PIE(Position Independent Executable)
  + Executables compiled such that their base address does not matter
* Use this command to make an executable position independent
  + Gcc -pie -fPIE -o tester tester.c
* Normally remote services are compiled this way so as to not get pwned.
* What are some ways that we can bypass ASLR?
  + Info leaks
    - Learking any sort of pointer to code means ASLR might have been defeated.
    - A single pointer into a memory segment allows you to compute the location of everything around it.
  + Partial address overwrite + crash state
  + Partial address overwrite + bruteforce