Title: Linux Kernel Exploitation Techniques

Description:

The number of user-land exploitation countermeasures outweighs the kernel protection mechanisms implemented by most modern distributions. Due to the complexity associated with exploiting user-land vulnerabilities, Linux kernel, with its huge publicly available codebase, has become an appealing target for exploit developers. A successful exploitation of a kernel vulnerability generally results in privilege escalation bypassing any user-land protections and exploit mitigations implemented by the OS.

This course teachers common kernel exploitation techniques on modern Linux distributions (x86_x64 architecture and 3.x/4.x kernels). It provides up-to-date information on current kernel hardening implementations and exploit mitigations. It is designed for students already familiar with user-land exploitation who want to play with the heart of the OS and gain fundamental knowledge required to develop reliable and effective kernel exploits. The course is structured as several theory modules (providing the necessary background material), followed by hands-on lab exercises demonstrating learned concepts in practice.

Even though this course is designed for beginners in kernel exploitation, a number of more advanced topics, such as reliable exploitation of heap vulnerabilities and SMEP/SMAP/KPTI bypasses, are discussed. The goal of this training is to demonstrate general exploitation concepts that can be applied to common classes of kernel memory corruption vulnerabilities.

This course is largely self-contained but please ensure you meet the entry requirements detailed below.

Key learning objectives:

- Exploiting kernel heap and stack vulnerabilities
- Exploiting integer vulnerabilities
- Reliable exploitation of use-after-free (UAF) vulnerabilities
- SMEP/SMAP/KPTI bypasses

Prerequisite knowledge:

- Familiarity with x86 (_64) architecture
- Linux working proficiency
- C and assembly programming knowledge
- Familiarity with GDB (GNU Debugger)

- Fundamental knowledge of common user-space exploitation techniques (e.g., stack and heap overflows, integer type conversion vulnerabilities and overflows, etc.)

Hardware/Software requirements:

- Base OS: Windows, OS X, Linux
- Ivy Bridge+ CPU (optional)
- At least 20GB of free disk space
- At least 8GB of RAM
- VMWare Workstation (v9+) or Fusion (v5+) (trial versions are sufficient)

Course agenda:

- Introduction to Linux kernel exploits
- Kernel debugging
- Privilege escalation techniques
- Read/write primitives and ret2usr attacks
- IDT overwrites (Interrupt Descriptor Table)
- Fixating the system and recovering the kernel state
- Information leaks
- Controlled, partially-controlled and uncontrolled read/write primitives
- Out of bounds (OOB) access vulnerabilities
- Integer vulnerabilities (signedness, typecasting, overflows)
- Kernel stack overflows
- Dynamic memory management/SLAB allocator
- Heap vulnerabilities (heap overflows, UAF, off-by-X)
- Reliable UAF exploitation on SMP systems
- SMEP/SMAP/KPTI bypasses

Bio:

Vitaly is a security researcher specialising in reverse engineering and exploit development. He has a solid academic background in programming languages, algorithms and cryptography. He is currently focused on OS security (kernel space exploitation techniques and countermeasures on POSIX systems) and software hypervisors.