Loïc Duflot

CPU Bugs, CPU Backdoors and Consequences on Security

13th European Symposium on Research in Computer Security (ESORICS 2008),
LNCS 5283, pp. 580-599, 2008. DOI: 10.1007/978-3-540-88313-5_37

Presentation By

Andreas Sembrant

From the Abstract

“Present the consequences on security of ***operating systems*** *and virtual machine monitors of the presence of a bug or backdoor in x86 processor.”*

Why you should read it?

- Technical report, but not to difficult.
- “Hacking”/Cracking an Operating System.
- GAS, Assembly, to activate a bug.
- Fun to read.

Good and Bad

+ Illustrates how easy a bug can be exploited.

- No explanation about what Operating System Security is.

Vulnerability

- Bug

  Involuntary implementation mistake.

- Backdoor

  Function whose only purpose is to grant additional privileges.

- Undocumented function

  Function implemented on purpose but not yet documented.

The Good

Illustrates how easy a bug can be exploited.

- ⇒ Only a couple of lines of assembly to activate the backdoor.
- ⇒ All OS security rendered useless.
- ⇒ Shows you how vulnerable your system really is.
- ⇒ Pray that Intel/AMD doesn’t implement any backdoors.

The Bad

No explanation about what Operating System Security is.

- ⇒ Assumes everyone share his idea of what OS security is.
- ⇒ Assumptions on what an attacker does to breach this “OS security”.
- ⇒ Demonstrates only one kind of attack.
- ⇒ No mention on which OS security feature is breached.

Attack

1. Use a bug to get additional privileges.
2. Modify OS data structures.
3. Do something evil.
4. Return the system to its original state.

Assumption

The attacker want to stay hidden. Need to return the system to its original state after the attack.

1. No traces in log files.
2. The computer should not crash as a result of an attack.
3. When attack is over, the processor should be in the same privileges state as before, e.g. ring 3.

Operating System Security

Operating System Security is preventing privilege escalation and restricting access to OS data structures.

Maybe prevent:

- Denial of Service, DOS.
- Damage hardware.

Attack

1. Use a bug to get additional privileges.
2. Modify OS data structures.
3. Do something evil.
4. Return the system to its original state.

1. Use a bug to get additional privileges.
2. Turn of interrupts and do nothing.

Operating System Security

“...consequences on security of ***operating systems****...”*

- What does the *proof of concept* prove if all the parameters are not specified?

- What are the consequences? Some kind of security is breached but what?

Would you implement a backdoor in your program?