OVERVIEW OF STACK CANARIES

Agenda

- Previous cycles:
  - Stack-based buffer overflow
  - DEP and ASLR mitigations
  - ROP
- This cycle:
  - Stack Canaries
    - a.k.a “Security Cookies” or “Stack Protectors”
  - Demonstration

Stack-based Buffer Overflow

- GOAL: Take control of EIP to execute code of your choosing
- Overwrite the return address with the address where your code resides

STACK

0x33333330	shellcode
0x33333334	shellcode
0x33333338	SAVED_EBP
0x3333333C	**0x33333330**
0x33333340	… stuff …
0x33333344	… stuff …

Stack Canaries

- 4-byte value placed on the stack
- Used to detect buffer overflows
- Protects Return Pointer, Saved Frame Pointer, and other stack variables

STACK WITHOUT CANARY:

0x33333330	Stack vars
0x33333334	Stack vars
0x33333338	Stack vars
0x3333333C	SAVED_EBP
0x33333340	RET_POINTER
0x33333344	Func args

STACK WITH CANARY:

0x3333332C	Stack vars
0x33333330	Stack vars
0x33333334	Stack vars
0x33333338	**CANARY**
0x3333333C	SAVED_EBP
0x33333340	RET_POINTER
0x33333344	Func args

Types of Stack Canaries

- NULL
  - 0x00000000
- Terminator
  - 0x00000AFF or 0x000AFF0D
- Random
  - Random Value using /dev/random or /dev/urandom
  - Sometimes XOR canary with stored control data

0x33333334	Stack vars
0x33333338	**0x00000000**
0x3333333C	SAVED_EBP
0x33333340	RET_POINTER
0x33333344	Func args

0x33333334	Stack vars
0x33333338	**0x000AFF0D**
0x3333333C	SAVED_EBP
0x33333340	RET_POINTER
0x33333344	Func args

0x33333334	Stack vars
0x33333338	**0x4F89E27C**
0x3333333C	SAVED_EBP
0x33333340	RET_POINTER
0x33333344	Func args

Tool Demonstration

GCC Compile Options

`-fno-stack-protector`	Disable Canary
`-fstack-protector`	Enable Canary on Some Functions containing Buffers of a Certain Size
`-fstack-protector-strong`	Enable Canary on More Functions to Provide Balance Between Security and Performance
`-fstack-protector-all`	Enable Canary on All Functions

Demonstration – Without Canary

```
// DISABLE CANARIES & DEP
#gcc -g -fno-stack-protector -z execstack
mybigecho.c -o mybigecho

// DISABLE ASLR
#echo 0 > /proc/sys/kernel/randomize_va_space

// EXPLOIT WORKS
#./mybigecho < payload
mybigecho() $esp = 0xbffff310
Address buffer  =  0xbffff310
  HOWDY TEXAS!!!!
```

Demonstration – With Canary

```
// ENABLE CANARY (KEEP DEP & ASLR DISABLED)
#gcc –g –fstack-protector -zexecstack
mybigecho.c -o mybigechoSC

// EXPLOIT
#./mybigechoSC < payload
mybigecho() $esp = 0xbffff2f0
Address buffer  =  0xbffff2fc
*** stack smashing detected ***: ./mybigechoSC
terminated
Segmentation fault
```

Summary

3 Main Ideas

- A stack canary is a 4-byte value stored on the stack to detect if the stack is overwritten. Windows refers to them as "security cookies". "Stack protector" is another term for a stack canary.
- Types of stack canaries are: NULL canaries, Terminator canaries, and Random canaries.
- Stack canaries are enabled/disabled by the compiler. In Linux, use the -fstack-protector flag to enable, and -fno-stack-protector to disable.

Future Work

- Tools that provide function pointer protection
- Attacks that thwart stack canaries
- Memory leakage attacks
- Mitigations such as the Enhanced Mitigation Experience Toolkit (EMET) for Windows or grsecurity for Linux

References

- Piessens, F. (2014, July 13). Low Level Software Security: Attacks and Countermeasures [video]. Youtube. Retrieved from https://www.youtube.com/watch?v=ZLZkf8FVcsU
- jip. (2012, December 21). Stack Smashing on a Modern Linux System. Retrieved from https://www.soldierx.com/tutorials/Stack-Smashing-Modern-Linux-System
- Wikipedia. (n.d.). Buffer Overflow Protection. Retrieved from https://en.wikipedia.org/wiki/Buffer_overflow_protection#GNU_Compiler_Collection_.28GCC.29

Post Questions and Comments to the Discussion Board