Malware Unpacking Workshop

Lilly Chalupowski
August 28, 2019

whois lilly.chalupowski

Table: *who.is results*

Name	Lilly Chalupowski
Status	Employed
Creation Date	1986
Expiry	A Long Time from Now (Hopefully)
Registrant Name	GoSecure
Administrative Contact	Travis Barlow
Job	TITAN Malware Research Lead

Agenda
What will we cover?

- Disclaimer
- Reverse Engineering
  - Registers
  - Stack
  - Heap
  - Assembly
  - Calling Conventions
- Tools
  - x64dbg
  - Cutter
  - Radare2
  - Detect it Easy
  - HxD
- Injection Techniques
  - DLL Injection
  - PE Injection
  - Process Hollowing
  - Atom Bombing
- Workshop

Disclaimer
Don't be a Criminal

disclaimer.log

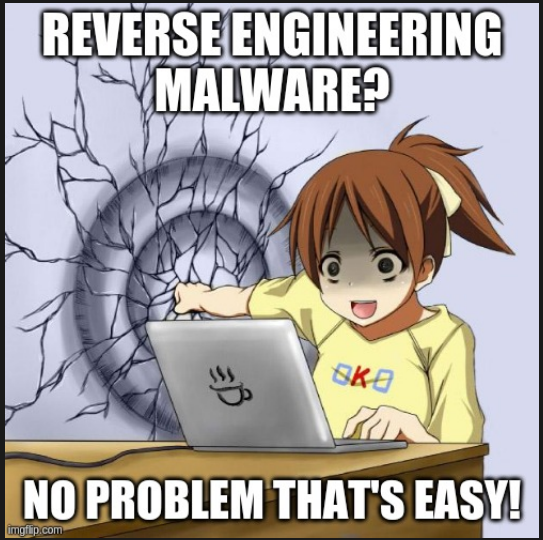
The tools and techniques covered in this presentation can be dangerous and are being shown for educational purposes.

It is a violation of Federal laws to attempt gaining unauthorized access to information, assets or systems belonging to others, or to exceed authorization on systems for which you have not been granted.

Only use these tools with/on systems you own or have written permission from the owner. I (the speaker) do not assume any responsibility and shall not be held liable for any illegal use of these tools.

Reverse Engineering

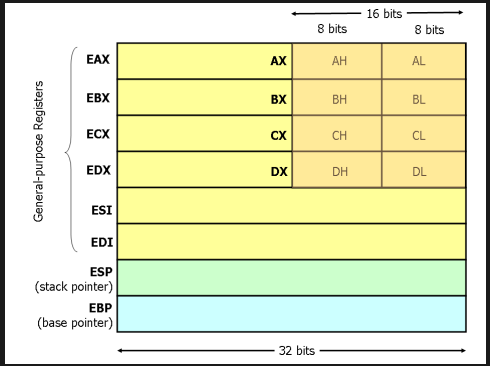
It's easy don't worry!



Registers
Not this one!

Registers
Not the kind with money in them

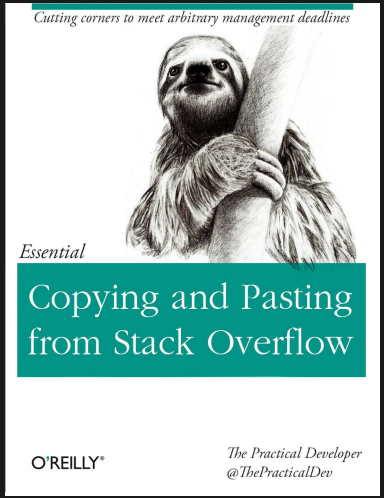
- EAX - Return Value of Functions
- EBX
- ECX - Counter in Loops
- EDI - Destination memory operations
- ESI - Source memory operations
- ESP - Stack pointer
- EBP - Base frame pointer

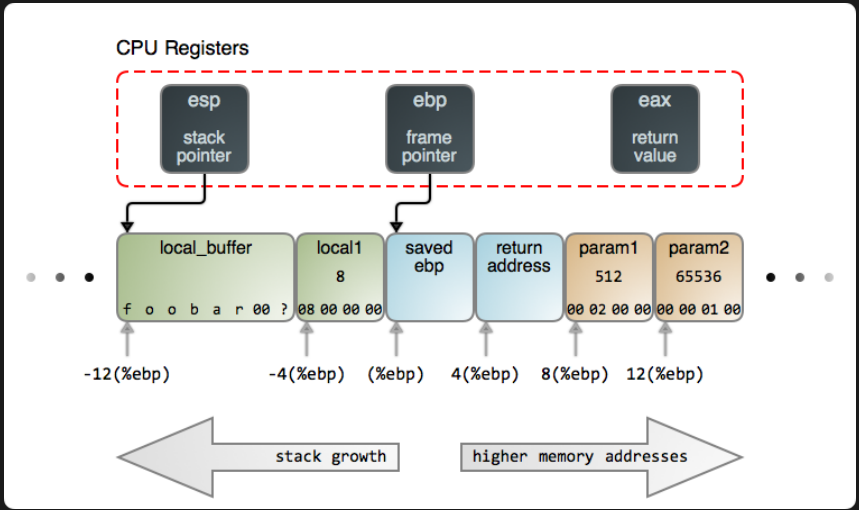


Did You Know: In computer architecture, a processor register is a quickly accessible location available to a computer's central processing unit (CPU).

The Stack

- Last-In First-Out
  - push
  - pop
- Downward Growth
- Function Local Variables
- ESP
- Increment / Decrement = 4
  - Double-Word Aligned




CPU Registers
esp
stack pointer
ebp
frame pointer
eax
return value
local_buffer
f o o b a r 00 ?
local1
8
08 00 00 00
saved ebp
return address
param1
512
00 02 00 00
param2
65536
00 00 01 00
-12(%ebp)
-4(%ebp)
(%ebp)
4(%ebp)
8(%ebp)
12(%ebp)
stack growth
higher memory addresses

Control Flow
Keeping it under control

- Conditionals
  - CMP
  - TEST
  - JMP
  - JCC
- EFLAGS
  - ZF / Zero Flag
  - SF / Sign Flag
  - CF / Cary Flag
  - OF/Overflow Flag

Calling Conventions
Subtitle goes here

- CDECL
  - Arguments Right-to-Left
  - Return Values in EAX
  - Calling Function Cleans the Stack
- STDCALL
  - Used in Windows Win32API
  - Arguments Right-to-Left
  - Return Values in EAX
  - The called function cleans the stack, unlike CDECL
  - Does not support variable arguments
- FASTCALL
  - Uses registers as arguments
  - Useful for shellcode

Windows Memory Structure

subtitle

- Stack - Grows up to lower addresses
- Heap - Grows down to higher addresses
- Program Image
- TEB - Thread Environment Block
  - GetLastError()
  - GetVersion()
  - Pointer to the PEB
- PEB - Process Environment Block
  - Image Name
  - Global Context
  - Startup Parameters
  - Image Base Address
  - IAT (Import Address Table)

- Syntax
  - Intel
  - AT&T
- Common Instructions
  - MOV
  - XOR
  - IMUL
  - DIV
  - PUSH
  - POP

Assembly Flavors

I know you were thinking it!