Returned Oriented Programming

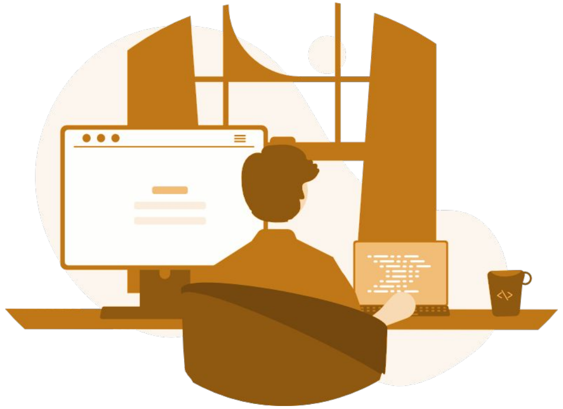
How is it affecting us?

Myles O'Toole

What Is ROPing?

- Returned oriented programming is where a hacker will infiltrate the stack before returning from a function that goes into a program (Code Arcana).
- The code is entangled within areas already segmented with memory.
- Doing so avoids the need for direct injection
- Using already trusted software, but manipulating it to a hacker’s needs (SecureTeam).

What type of vulnerability is it?



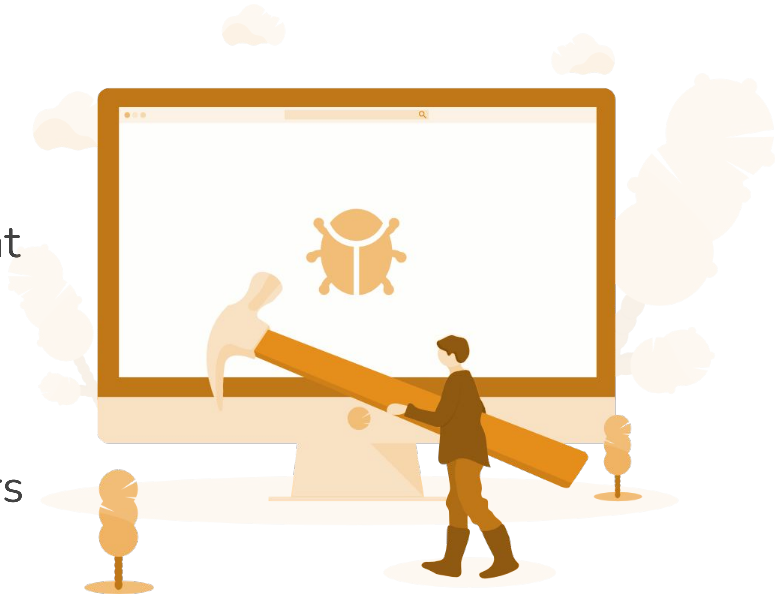
- Returned oriented programming is a memory leak/corruption vulnerability where the attacker can come take advantage.
- The leak and corruption of memory can dismay the security of the program.

How does it work?

- Memory corruption to the program rewriting itself and eventually granting unauthorized access to user privileges.
- May even rewrite itself for a hacker to correlate their own code to further infiltrate the system.
- Memory to memory, memory to register, register to memory.

Why is architecture involved?

- Requires a knowledge of x86 and calling conventions.
- 32 bit and 64 bit programs require different skill sets.
- Requires knowledge of stack.
- Different processors save different registers in stack.



Is there any way to patch the problem?

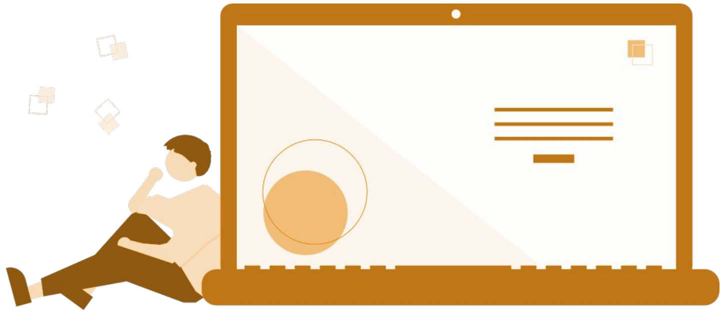
- You can guard your program by using good programming methods and habits.
- Verifying the pointers correlate to the correct area of the stack.
- Make sure the return address to the function is correct and continues on to another separate function.

Are there workarounds?

- Considering protection against function calls that can alter memory.
- Implementing safe code practices safe from API hooking.
- API hooking involves the interception of function calls to examine or change the information between them.

How concerned should we be?

- Awareness of threats when building or examining code.
- Test your codes ability to protect against injections.
- Making sure you know what functions manipulate.
- Protecting functions from injection and spreading to other functions.



References

Apriorit. (2021, November 25). *Rop chain. how to defend from Rop Attacks (basic example)*. Apriorit. Retrieved March 31, 2022, from https://www.apriorit.com/dev-blog/434-rop-exploit-protection

Code arcana. Code Arcana ATOM. (n.d.). Retrieved March 31, 2022, from https://codearcana.com/posts/2013/05/28/introduction-to-return-oriented-programming-rop.html

Elliott, E. (n.d.). *Programming javascript applications*. O'Reilly Online Learning. Retrieved March 31, 2022, from https://www.oreilly.com/library/view/programming-javascript-applications/9781491950289/ch05.html

Faithfull, M. (2020, September 21). *How return-oriented programming exploits work*. SecureTeam. Retrieved March 31, 2022, from https://secureteam.co.uk/articles/how-return-oriented-programming-exploits-work/

Return Oriented Programming (ROP) attacks. Infosec Resources. (2021, June 2). Retrieved March 31, 2022, from https://resources.infosecinstitute.com/topic/return-oriented-programming-rop-attacks/