WASM

Younes BELCOUCHE
Comme DE CERVAL
Nayef LATASSE
Benjamain KHOLTES
Djoulien RASPAUD

Index

What is WASM ?

WASM use case

Vulnerabilities

Demo

Conclusion

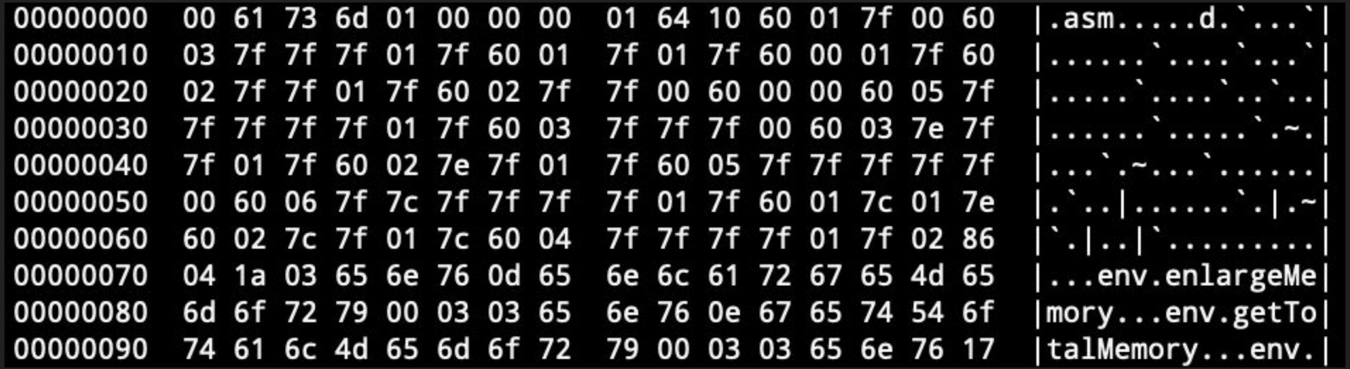
WASM

“WebAssembly (abbreviated *Wasm*) is a binary instruction format for a stack-based virtual machine. Wasm is designed as a portable target for compilation of high-level languages like C/C++/Rust, enabling deployment on the web for client and server applications.”

-- Not from wikipedia (https://webassembly.org)

WASM

A Wasm binary looks like this:



```
00000000  00 61 73 6d 01 00 00 00  01 64 10 60 01 7f 00 60  |.asm.....d.`...`|
00000010  03 7f 7f 7f 01 7f 60 01  7f 01 7f 60 00 01 7f 60  |......`....`...`|
00000020  02 7f 7f 01 7f 60 02 7f  7f 00 60 00 00 60 05 7f  |.....`....`..`..|
00000030  7f 7f 7f 7f 01 7f 60 03  7f 7f 7f 00 60 03 7e 7f  |......`.....`.~.|
00000040  7f 01 7f 60 02 7e 7f 01  7f 60 05 7f 7f 7f 7f 7f  |...`.~...`......|
00000050  00 60 06 7f 7c 7f 7f 7f  7f 01 7f 60 01 7c 01 7e  |.`..|......`.|.~|
00000060  60 02 7c 7f 01 7c 60 04  7f 7f 7f 7f 01 7f 02 86  |`.|..|`.........|
00000070  04 1a 03 65 6e 76 0d 65  6e 6c 61 72 67 65 4d 65  |...env.enlargeMe|
00000080  6d 6f 72 79 00 03 03 65  6e 76 0e 67 65 74 54 6f  |mory...env.getTo|
00000090  74 61 6c 4d 65 6d 6f 72  79 00 03 03 65 6e 76 17  |talMemory...env.|
```

Luckily, WASM is compatible with many languages like python or C.

(https://github.com/appcypher/awesome-wasm-langs)

WASM programs allow us to complete javascript, giving it the possibility to run any piece of code using any higher performance language (in our case, C).

WASM

Its primary characteristics are:

- Efficient and Fast
- “Safe”
- Open and debuggable
- Part of the open web platform
- Hardware independent
- Language independent

(https://webassembly.github.io/spec/core/intro/introduction.html)

WASM Applications

Multiple WASM applications exist on the Web:

- QuakeJS (Multiplayer is possible)
- Geogram
- Port of various Qt Demos

(More here: https://github.com/kripken/emscripten/wiki/Porting-Examples-and-Demos)

Unfortunately, there are no sensitive WASM apps currently deployed on the web.

So we have to create our own flawed program...

How will we proceed ?

Known Exploits

Since we compile from C to WASM, many C vulnerabilities are exploitable using WASM:

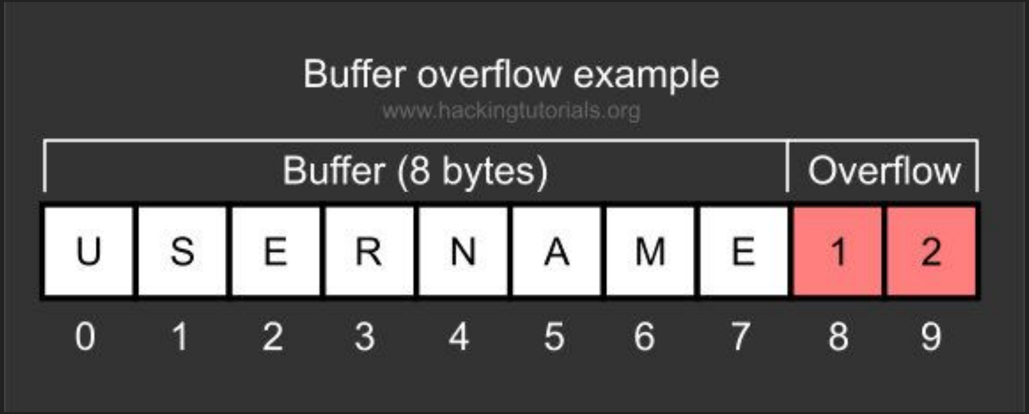
- Buffer Overflow
- Integer Overflow
- Format String attacks

Our exploit today executes an XSS attack on a node server to remote execute codes using buffer overflow.

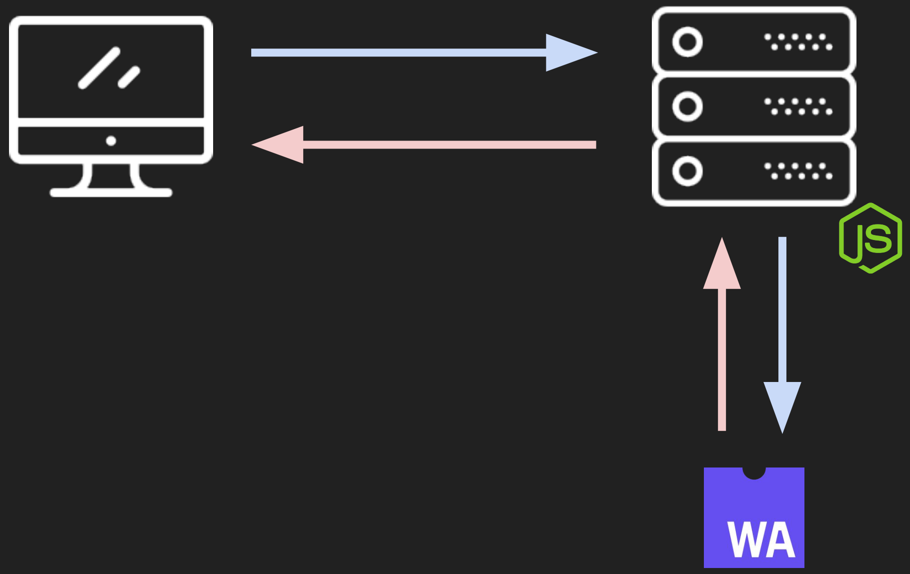
But, what is a Buffer Overflow?

A buffer overflow, is an anomaly where a program, while writing data to a buffer, overruns the buffer's boundary and overwrites adjacent memory locations.

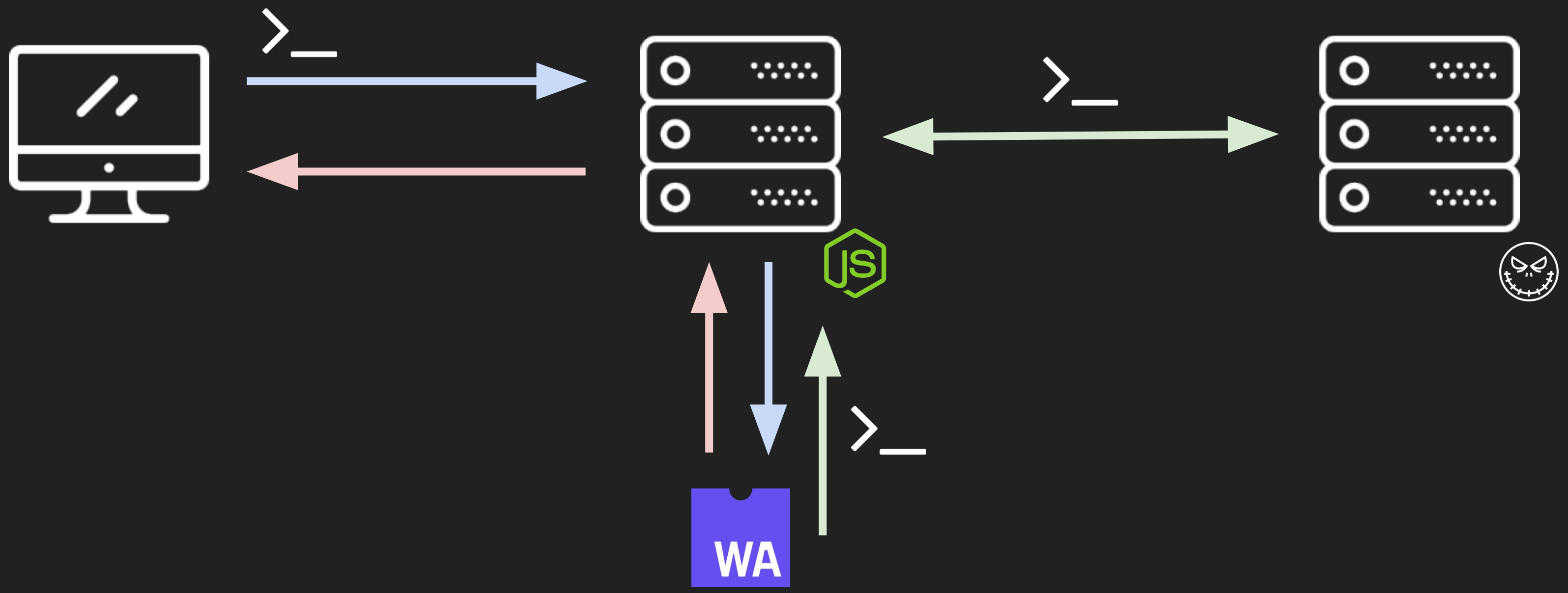
It can be used by malicious guys to execute arbitrary code by writing in the right place.



Demo: normal running



Demo: XSS + buffer overflow



Conclusion

The intercompatibility between wasm and other languages exposes WASM to the security flaws of the other languages.

Even if WASM is advertised as “safe”, developers should be aware that some patterns are still dangerous.