
VERACODE
Using WebAssembly to run, extend, and secure your .NET application
Niels Tanis
Sr. Principal Security Researcher
Genetec
ElevateDev'24

Who am I?

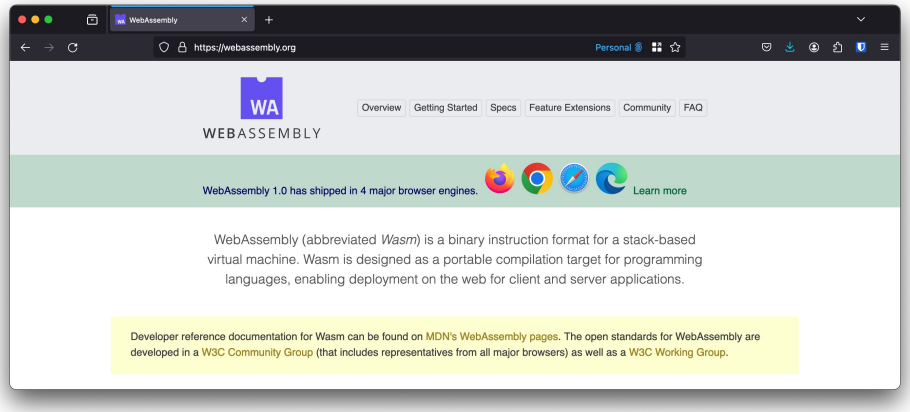
- Niels Tanis
- Sr. Principal Security Researcher
  - Background .NET Development, Pentesting/ethical hacking, and software security consultancy
  - Research on static analysis for .NET apps
  - Enjoying Rust!
- Microsoft MVP – Developer Technologies

VERACODE

MVP Microsoft® Most Valuable Professional

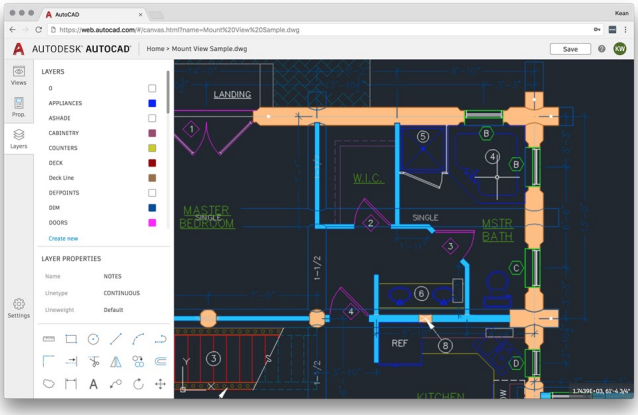
@nielstanis@infosec.exchange

WebAssembly



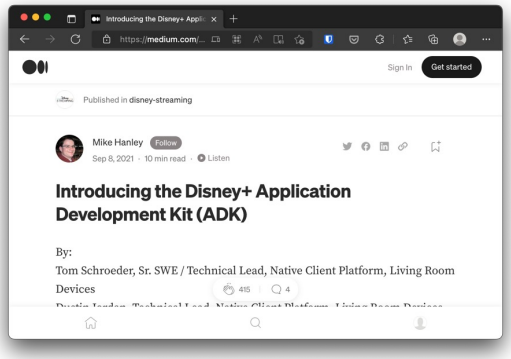
https://webassembly.org/

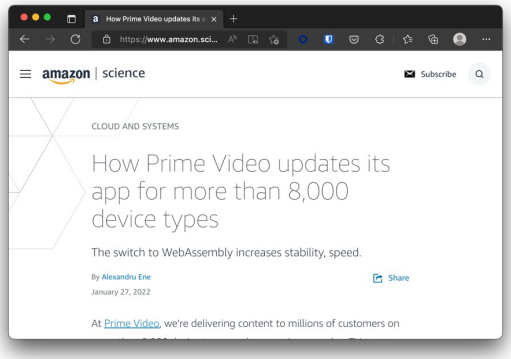
WebAssembly - AutoCAD



@nielstanis@infosec.exchange

WebAssembly - SDK's





https://medium.com/disney-streaming/introducing-the-disney-application-development-kit-adk-ad85ca139073
https://www.amazon.science/blog/how-prime-video-updates-its-app-for-more-than-8-000-device-types

Agenda

- Introduction
- WebAssembly Design & Internals
- Running .NET on WebAssembly
- Extending .NET with WebAssembly
- Securing .NET with WebAssembly
- Conclusion
- Q&A

WebAssembly Design

- **Be fast, efficient, and portable**
  - Executed in near-native speed across different platforms
- **Be readable and debuggable**
  - In low-level bytecode but also human readable
- **Keep secure**
  - Run on sandboxed execution environment
- **Don't break the web**
  - Ensure backwards compatibility

WebAssembly

- Binary instruction format for stack-based virtual machine similar to .NET CLR running MSIL or JVM running bytecode
- Designed as a portable compilation target

https://hacks.mozilla.org/2017/02/creating-and-working-with-webassembly-modules/
https://webassembly.org/

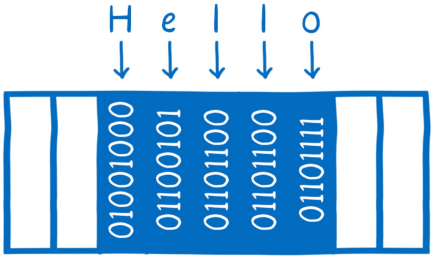
WebAssembly

- The security model of WebAssembly:
  - Protect users from buggy or malicious modules
  - Provide developers with useful primitives and mitigations for developing safe applications

https://hacks.mozilla.org/2017/02/creating-and-working-with-webassembly-modules/
https://webassembly.org/

WebAssembly Memory

- Isolated per WASM module
- A contiguous, mutable array of uninterpreted bytes



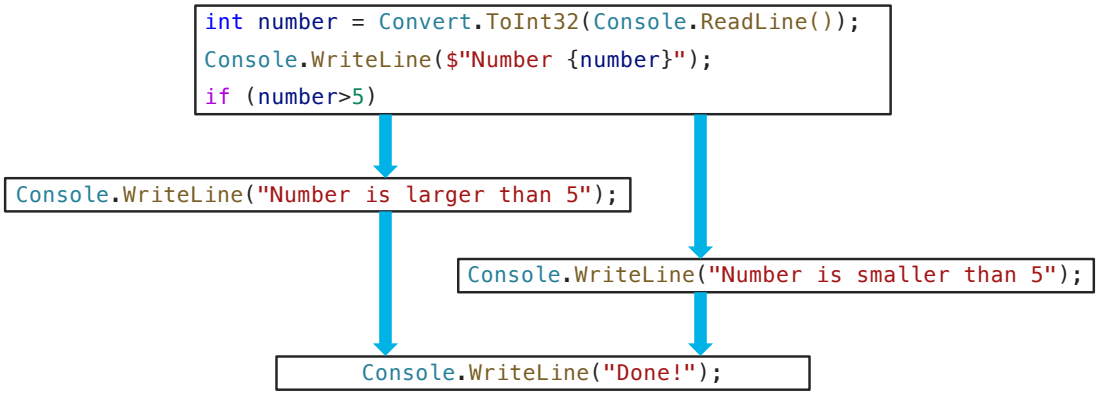
@nielstanis@infosec.exchange

WebAssembly Control-Flow Integrity

```
int number = Convert.ToInt32(Console.ReadLine());
Console.WriteLine($"Number {number}");
if (number>5)
{
        Console.WriteLine("Number is larger than 5");
}
else
{
        Console.WriteLine("Number is smaller than 5");
}
Console.WriteLine("Done!");
```

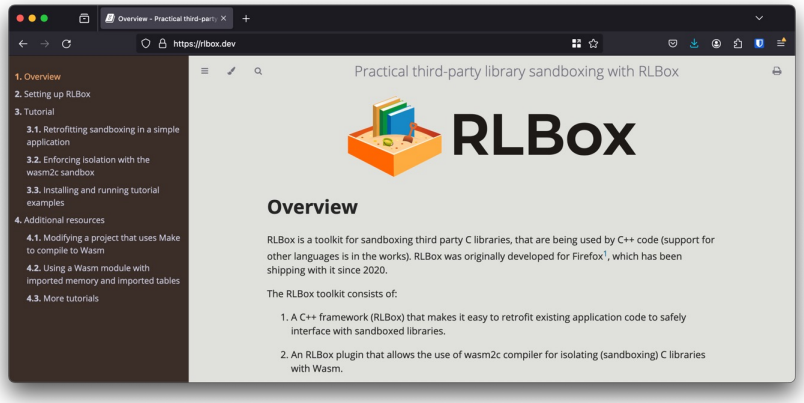

@nielstanis@infosec.exchange

WebAssembly Control-Flow Integrity



@nielstanis@infosec.exchange

FireFox RLBox



@nielstanis@infosec.exchange

https://rlbox.dev/
https://hacks.mozilla.org/2020/02/securing-firefox-with-webassembly/

Running .NET on WebAssembly

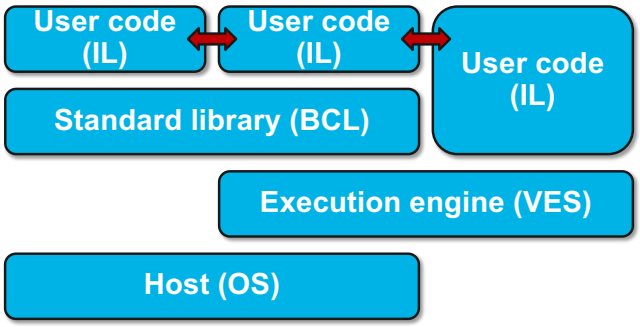
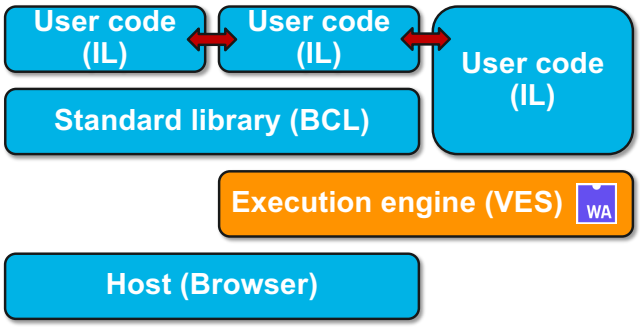


Diagram: https://github.com/itowlson/wasmday22/blob/main/slides/Wasm%20Interfaces%20and%20.NET.pptx

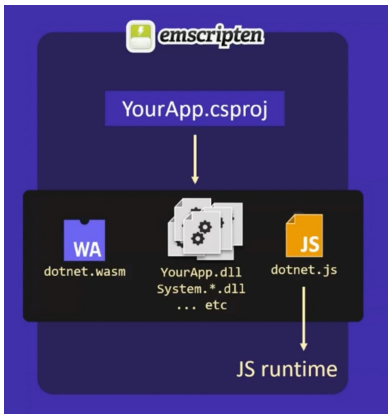
Running .NET on WebAssembly



@nielstanis@infosec.exchange

Diagram:
https://github.com/itowlson/wasmday22/blob/main/slides/Wasm%20Interfaces%20and%20.NET.pptx

Blazor WebAssembly



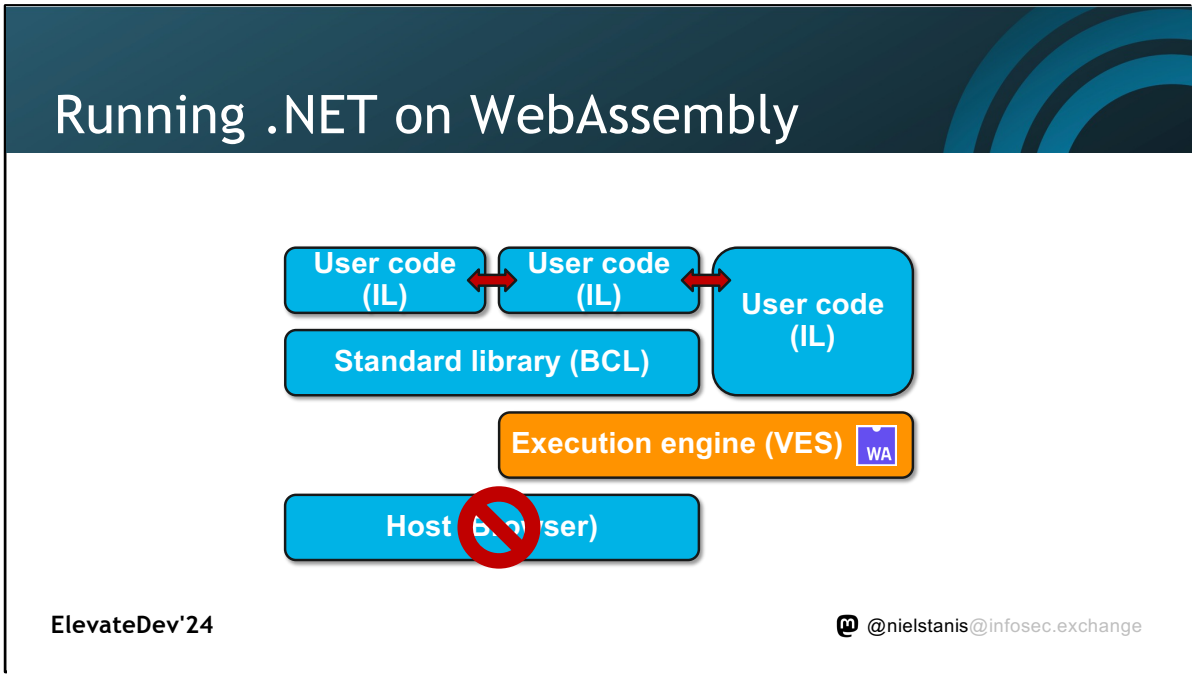


Diagram: https://github.com/itowlson/wasmday22/blob/main/slides/Wasm%20Interfaces%20and%20.NET.pptx

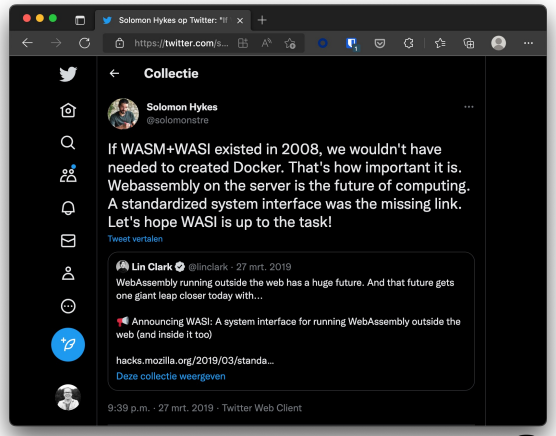
WebAssembly System Interface WASI

- Introduced in March 2019 by Bytecode Alliance
- WasmTime implementation as reference
- POSIX inspired, engine-independent, non-Web system-oriented API for WebAssembly

WebAssembly System Interface WASI

- Strong sandbox with Capability Based Security
- Preview1, supports e.g. FileSystem actions
- Future support for sockets and other system resources.
- Anyone recall .NET Standard? ☺

Docker vs WASM & WASI

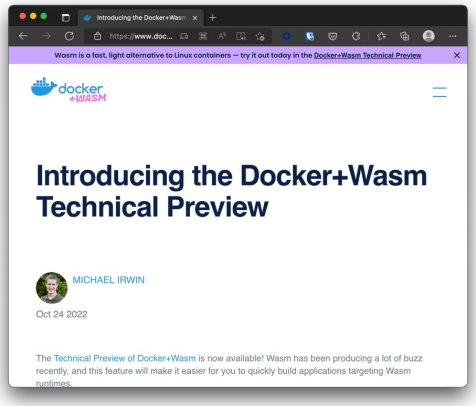


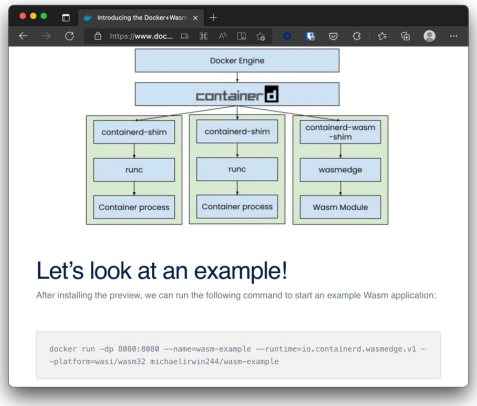
@nielstanis@infosec.exchange

Docker vs WASM & WASI

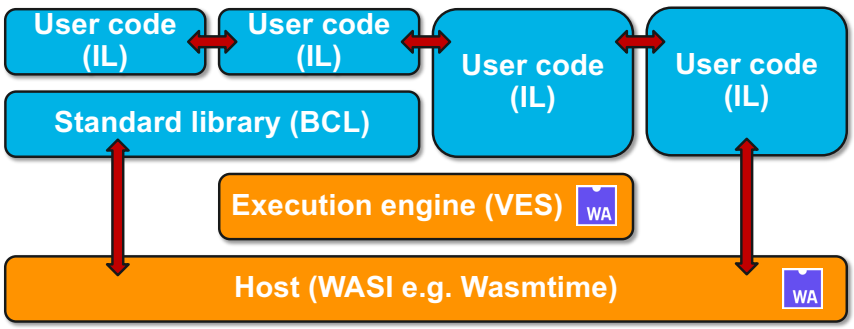
@nielstanis@infosec.exchange

Docker & WASM


Wasm is a fast, light alternative to Linux containers — try it out today in the Docker+Wasm Technical Preview
docker +WASM
Introducing the Docker+Wasm Technical Preview
MICHAEL IRWIN
Oct 24 2022
The Technical Preview of Docker+Wasm is now available! Wasm has been producing a lot of buzz recently, and this feature will make it easier for you to quickly build applications targeting Wasm runtimes.

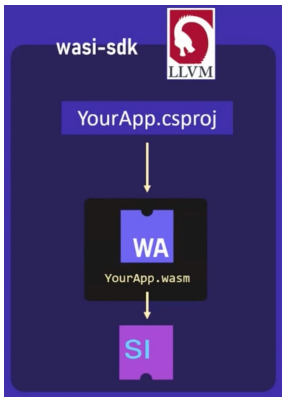

Docker Engine
containerd
containerd-shim
runc
Container process
containerd-shim
runc
Container process
containerd-wasm-shim
wasmedge
Wasm Module
Let's look at an example!
After installing the preview, we can run the following command to start an example Wasm application:
docker run -dp 8080:8080 --name=wasm-example --runtime=io.containerd.wasmedge.v1 --platform=wasi/wasm32 michaelirwin244/wasm-example

WebAssembly System Interface WASI



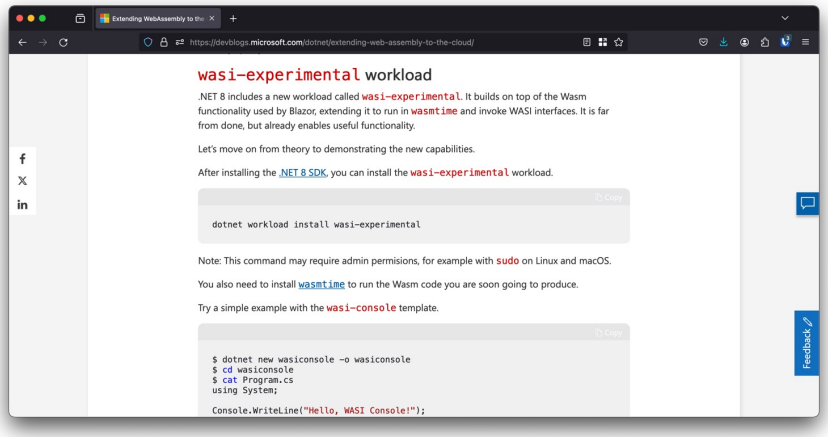
@nielstanis@infosec.exchange

Experimental WASI SDK for .NET



https://github.com/SteveSandersonMS/dotnet-wasi-sdk

.NET 8 WASI-Experimental



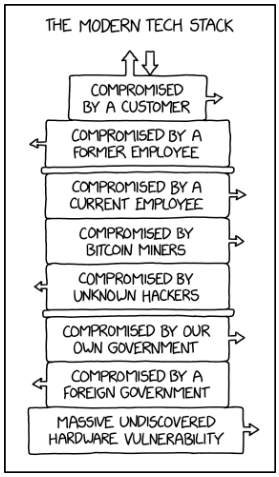
https://github.com/dotnet/runtime/issues/65895
https://github.com/SteveSandersonMS/dotnet-wasi-sdk

https://devblogs.microsoft.com/dotnet/extending-web-assembly-to-the-cloud/

Extending .NET with WASM

- WasmTime.NET NuGet package
- Can run WASM inside of any .NET application
- Extend with Rust based WASM module
- Limit capabilities
- Demo time!

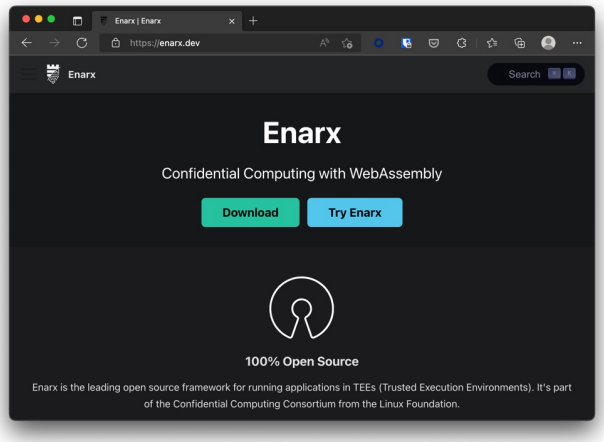
Trusted Computing – XKCD 2166



@nielstanis@infosec.exchange

https://xkcd.com/2166/

Enarx

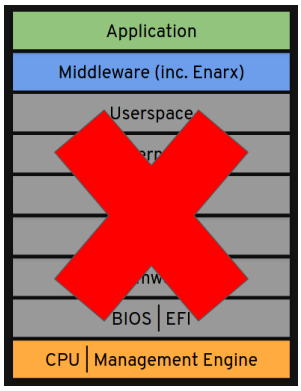


@nielstanis@infosec.exchange

https://enarx.dev/

Enarx Threat Model

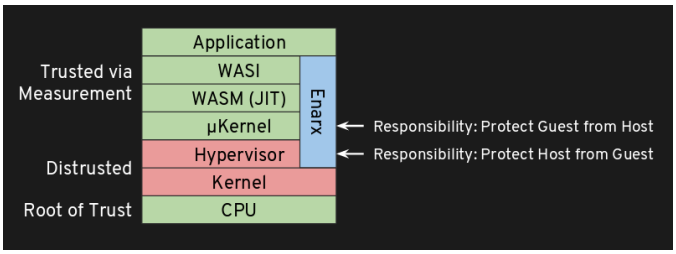
- Don’t trust the host
- Don’t trust the host owner
- Don’t trust the host operator
- Hardware cryptographically verified
- Software audited and cryptographically verified



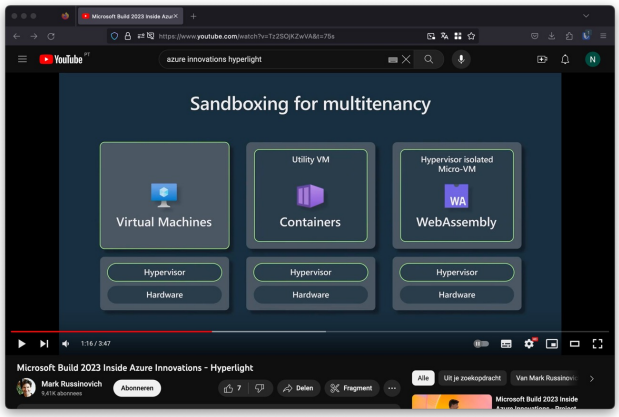
@nielstanis@infosec.exchange

Enarx

- Leverages Trusted Execution Environment (TEE) direct on processor
  - AMD's SEV, Intel's SGX and IBM's PEF
- Attestation of hardware and Enarx runtime

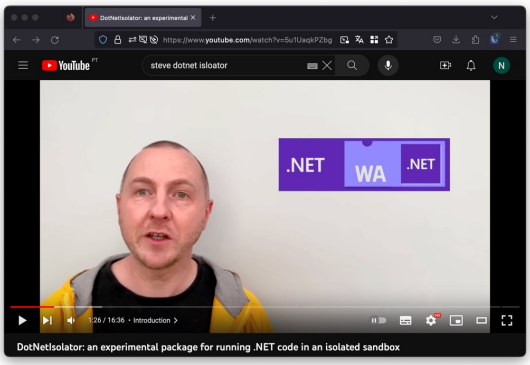


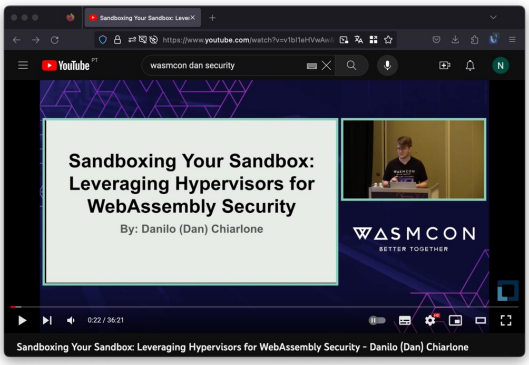
Project Hyperlight



@nielstanis@infosec.exchange

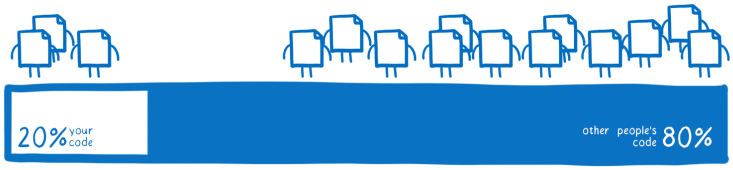
DotNetIsolator & Project Hyperlight





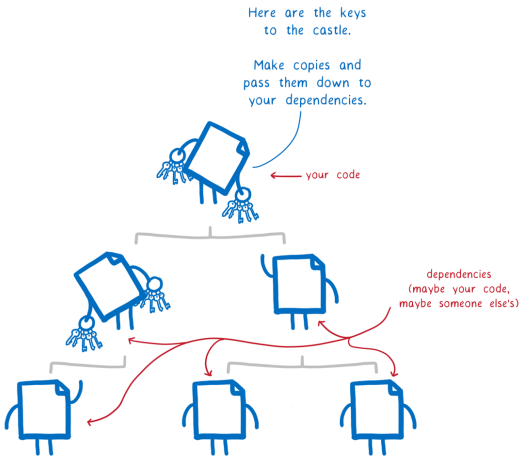
WASM – What's next?

composition of an average code base

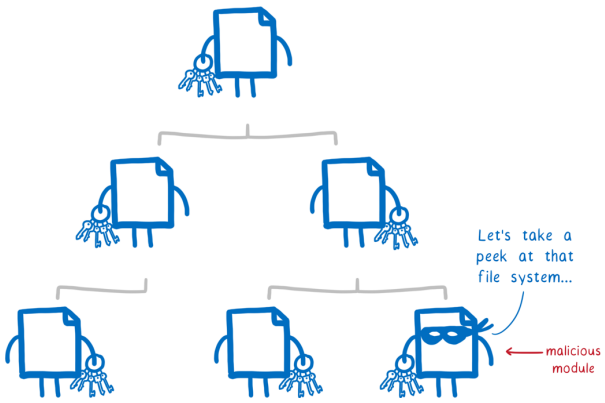


@nielstanis@infosec.exchange

Dependencies

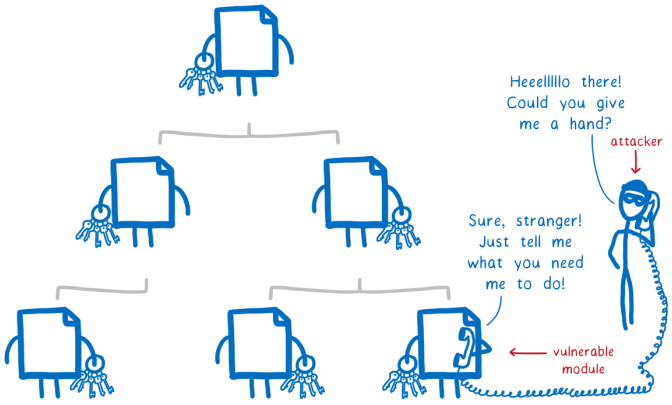


Malicious module



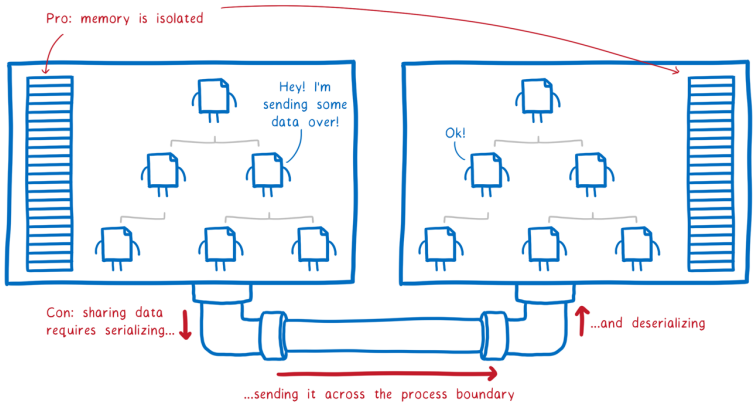
@nielstanis@infosec.exchange

Vulnerable module



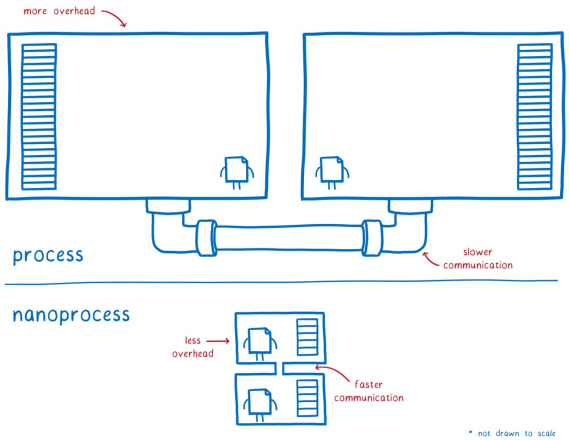
@nielstanis@infosec.exchange

Process Isolation



@nielstanis@infosec.exchange

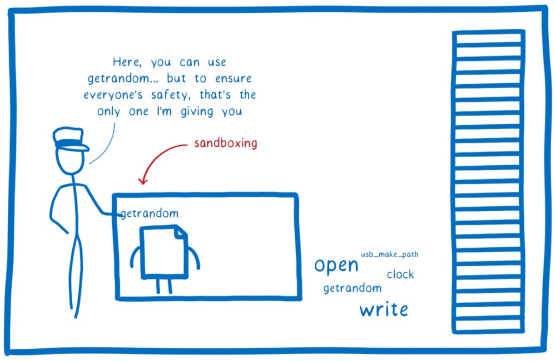
WebAssembly Nano-Process



@nielstanis@infosec.exchange

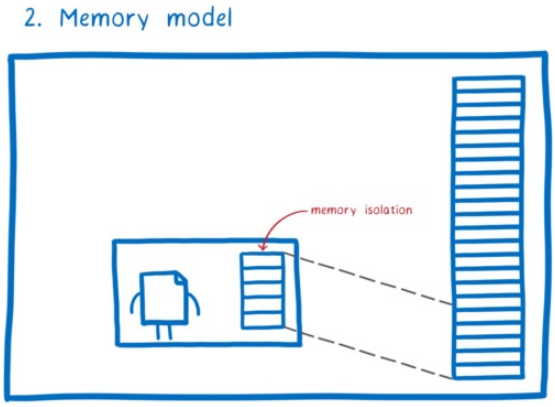
WebAssembly Nano-Process

1. Sandboxing



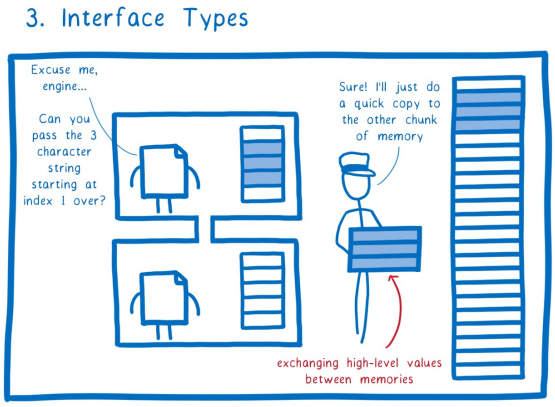
@nielstanis@infosec.exchange

WebAssembly Nano-Process

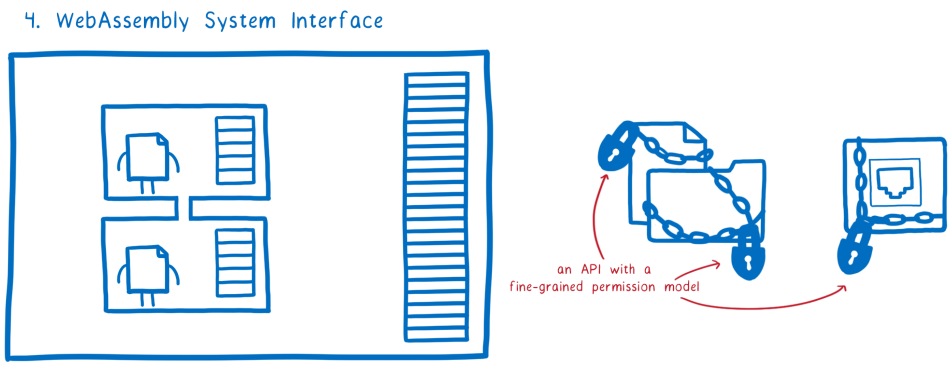


@nielstanis@infosec.exchange

WebAssembly Nano-Process

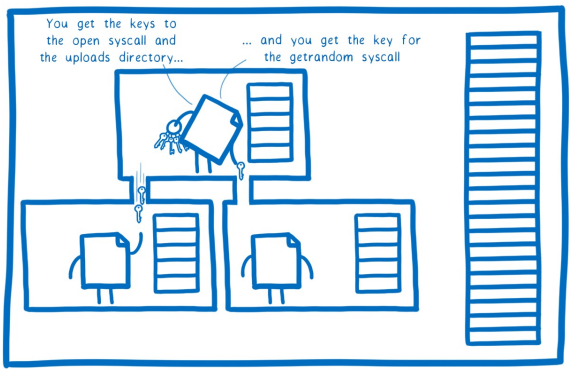


WebAssembly Nano-Process

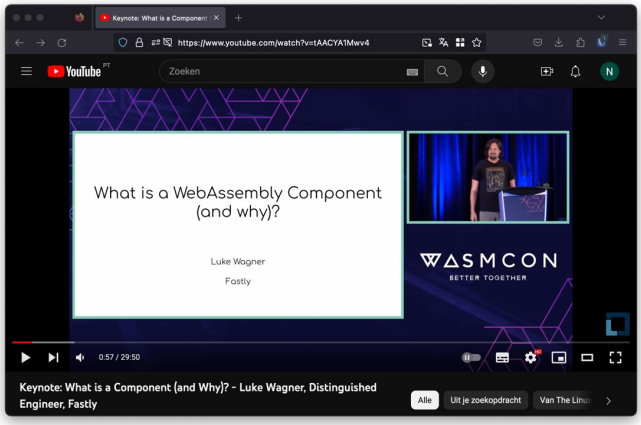


WebAssembly Nano-Process

5. The missing link



WebAssembly Component Model

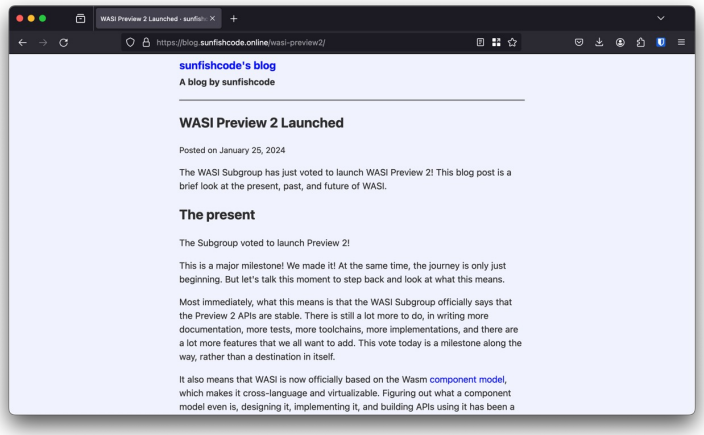


ElevateDev'24

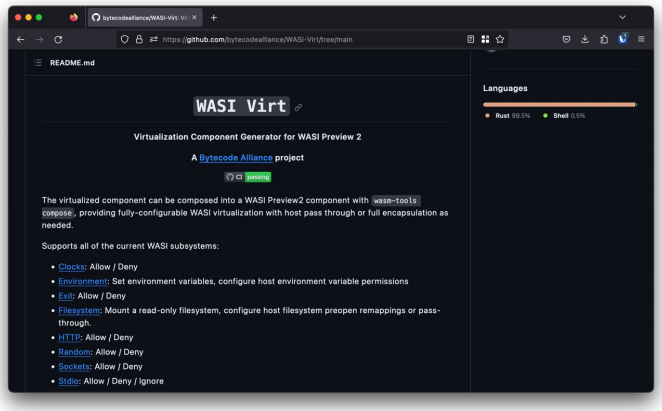
@nielstanis@infosec.exchange

https://www.youtube.com/watch?v=tAACYA1Mwv4

WASI Preview 2



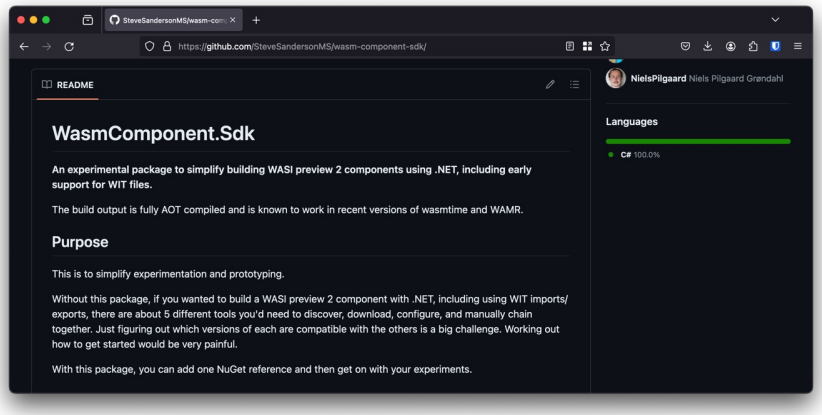
WASI Virt



@nielstanis@infosec.exchange

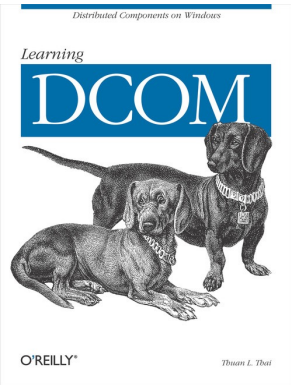
https://www.youtube.com/watch?v=tAACYA1Mwv4

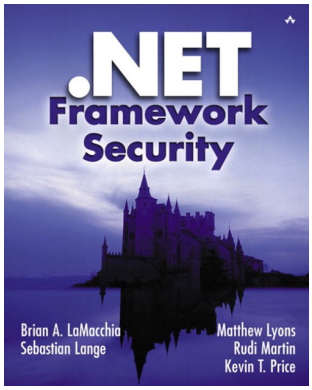
WasmComponent.SDK



https://github.com/SteveSandersonMS/wasm-component-sdk/

Have we seen this before?



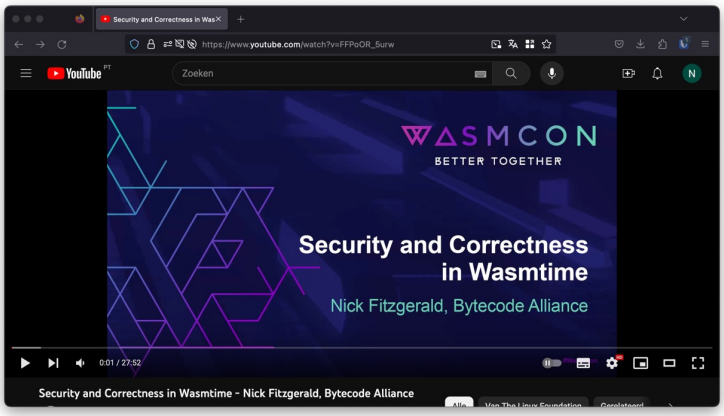


Runtimes and Security

- Most security research published focusses on correctness of WASM runtimes/VM's
- Bytecode Alliance Blogpost September 2022:
  - "Security and Correctness in Wasmtime"
  - Written in Rust → Using all it's LangSec features
  - Continues Fuzzing & formal verification
  - Security process & vulnerability disclosure

https://bytecodealliance.org/articles/security-and-correctness-in-wasmtime

Runtimes and Security



https://www.youtube.com/watch?v=FFPoOR_5urw

Conclusion

- Cloud Native ❤️WebAssembly
- WebAssembly has a lot of potential to be used to run, extend, and secure your applications!
- Its as secure as the WebAssembly runtime implementation!
- WASI Preview 2 big milestone; now tooling can be implemented!

Questions?

- https://github.com/nielstanis/elevatedev24wasm/
- ntanis at Veracode.com
- @nielstanis@infosec.exchange
- https://blog.fennec.dev
- Merci! Thank you!