Beyond Trust: Building Community-Driven Security Analysis for Your .NET Software Supply Chain

Niels Tanis

Sr. Principal Security Researcher

VERACODE

NDC { Manchester }

Who am I?

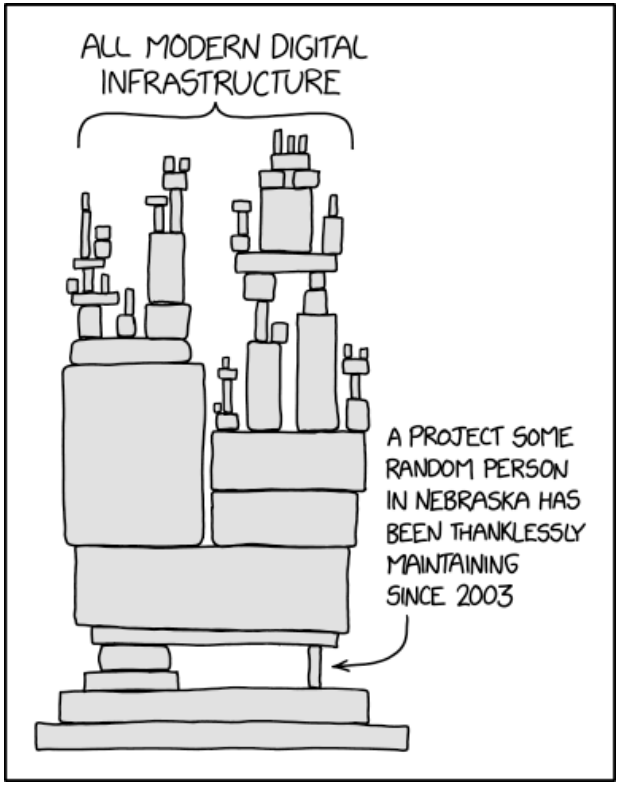
- Niels Tanis
- Sr. Principal Security Researcher
  - Background .NET Development, Pentesting/ethical hacking, and software security consultancy
  - Research on static analysis for .NET apps
  - Enjoying Rust!
- Microsoft MVP – Developer Technologies

VERACODE

MVP Microsoft® Most Valuable Professional

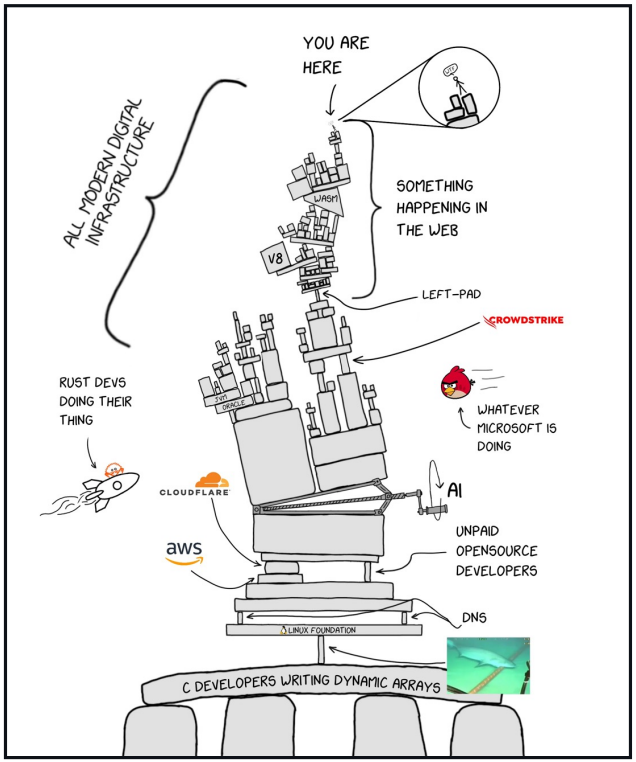
@niels.fennec.dev @nielstanis@infosec.exchange

Modern Application Architecture
XKCD 2347



@niels.fennec.dev @nielstanis@infosec.exchange

You are here 😂



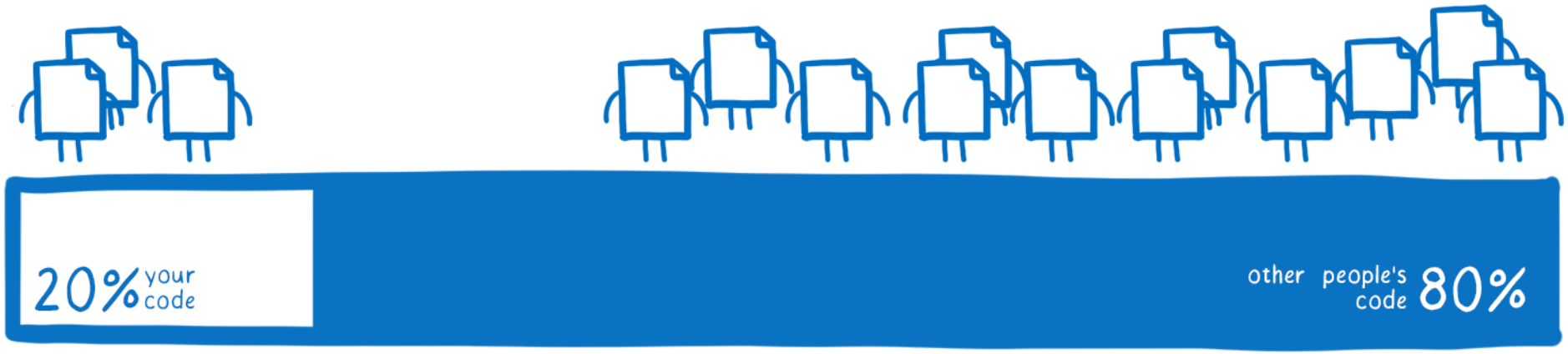
@niels.fennec.dev @nielstanis@infosec.exchange

Agenda

- Security challenges in our .NET software supply chain
- How can we start securing our supply-chain?
- Fennec Labs
  - Understanding what projects do for security
  - Know what's inside package
  - Review package
  - Feeback & sharing
  - Future and idea’s
- Conclusion and Q&A

@niels.fennec.dev @nielstanis@infosec.exchange

Average codebase composition



@niels.fennec.dev @nielstanis@infosec.exchange

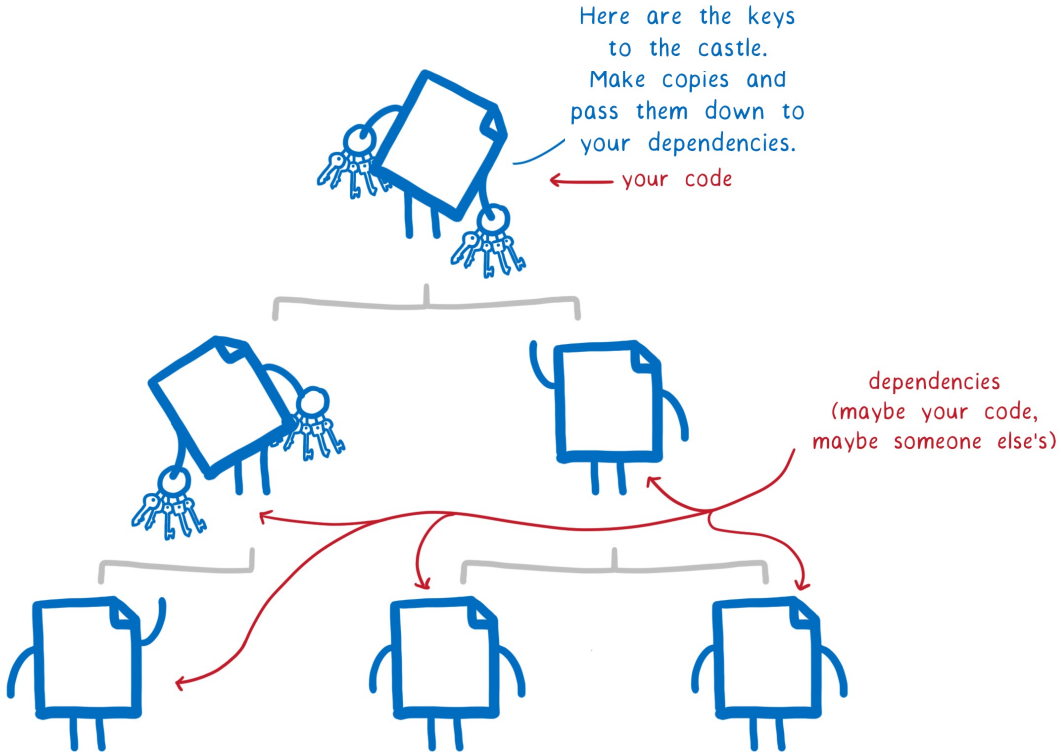
State Of Software Security 2025

64%
of applications have flaws in **first-party code**

70%
of applications have flaws in **third-party code**

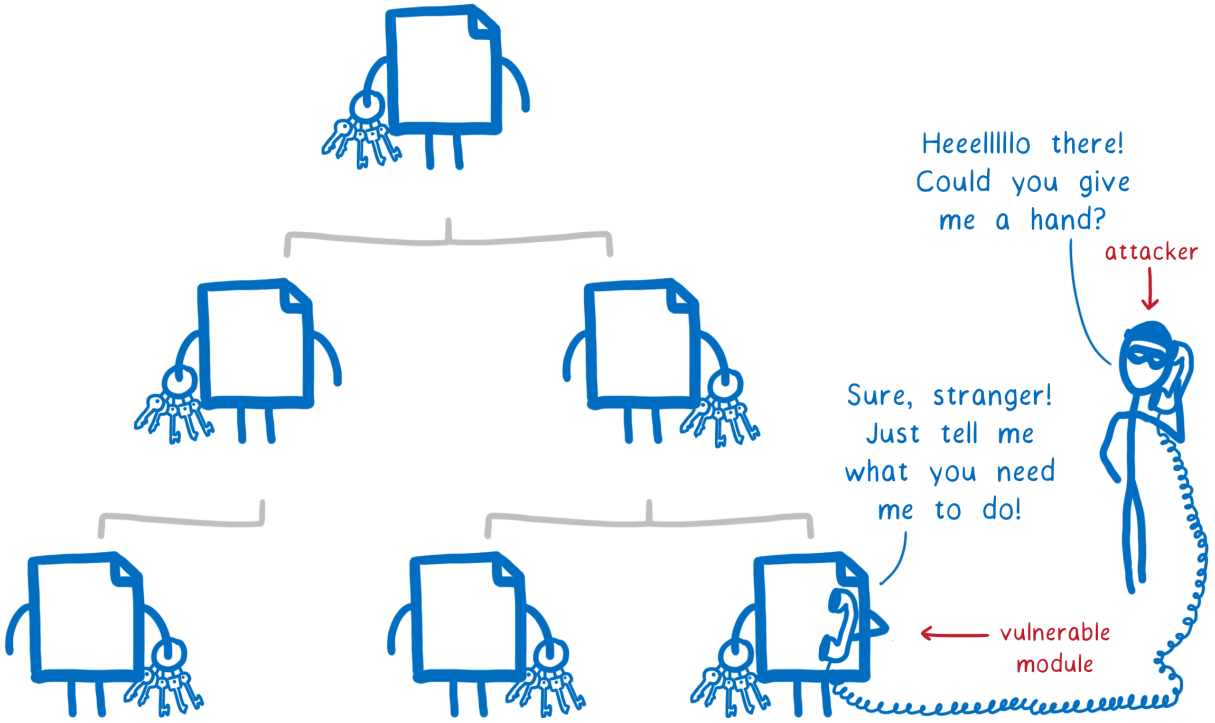
@niels.fennec.dev @nielstanis@infosec.exchange

Average codebase composition



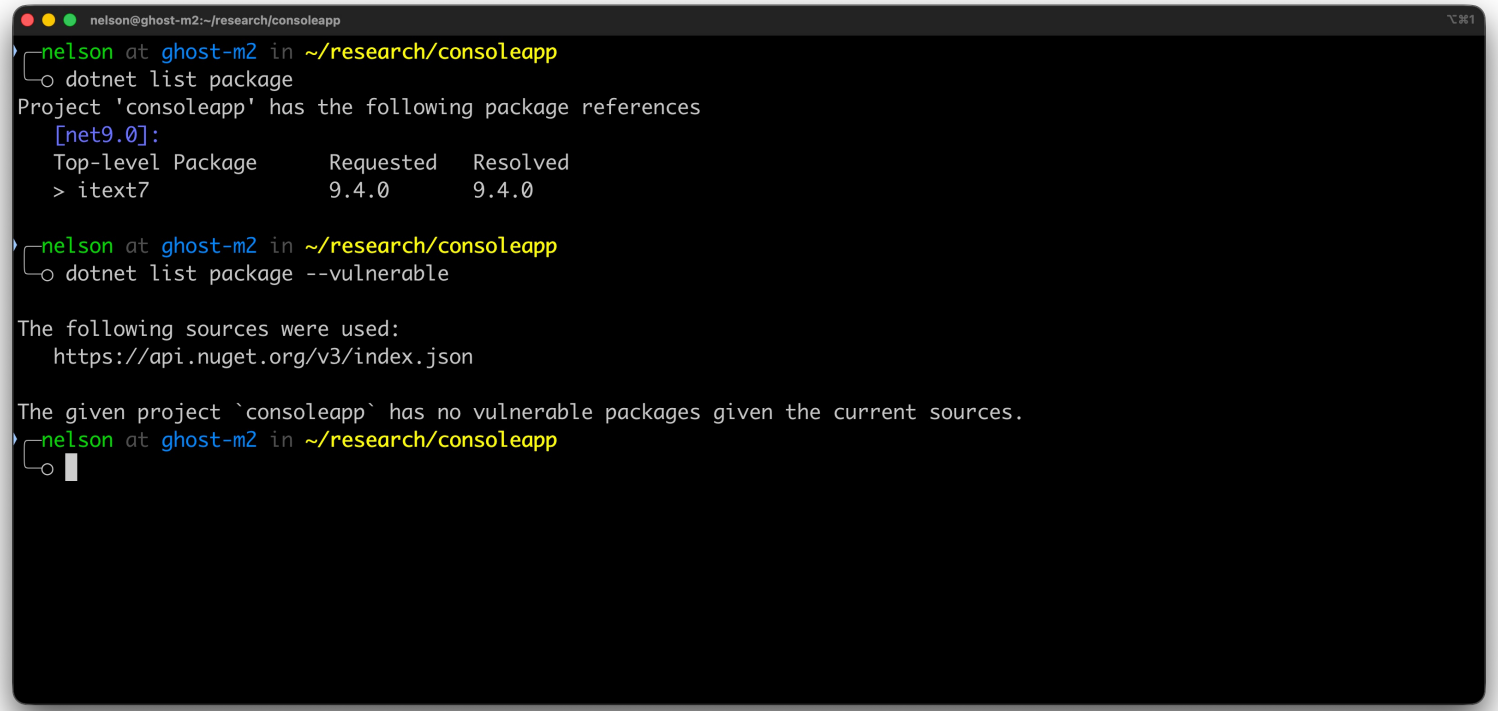
@niels.fennec.dev @nielstanis@infosec.exchange

Vulnerable Assembly



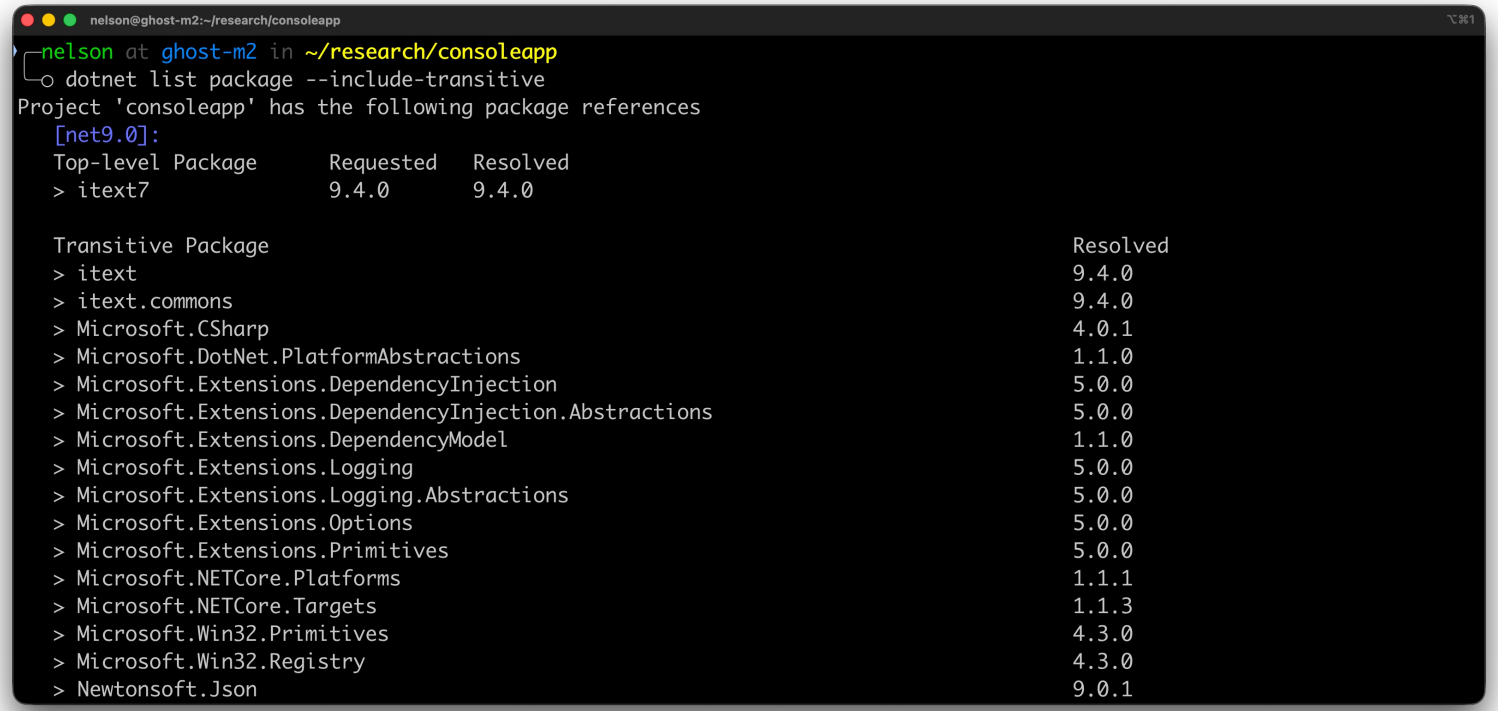
@niels.fennec.dev @nielstanis@infosec.exchange

DotNet CLI



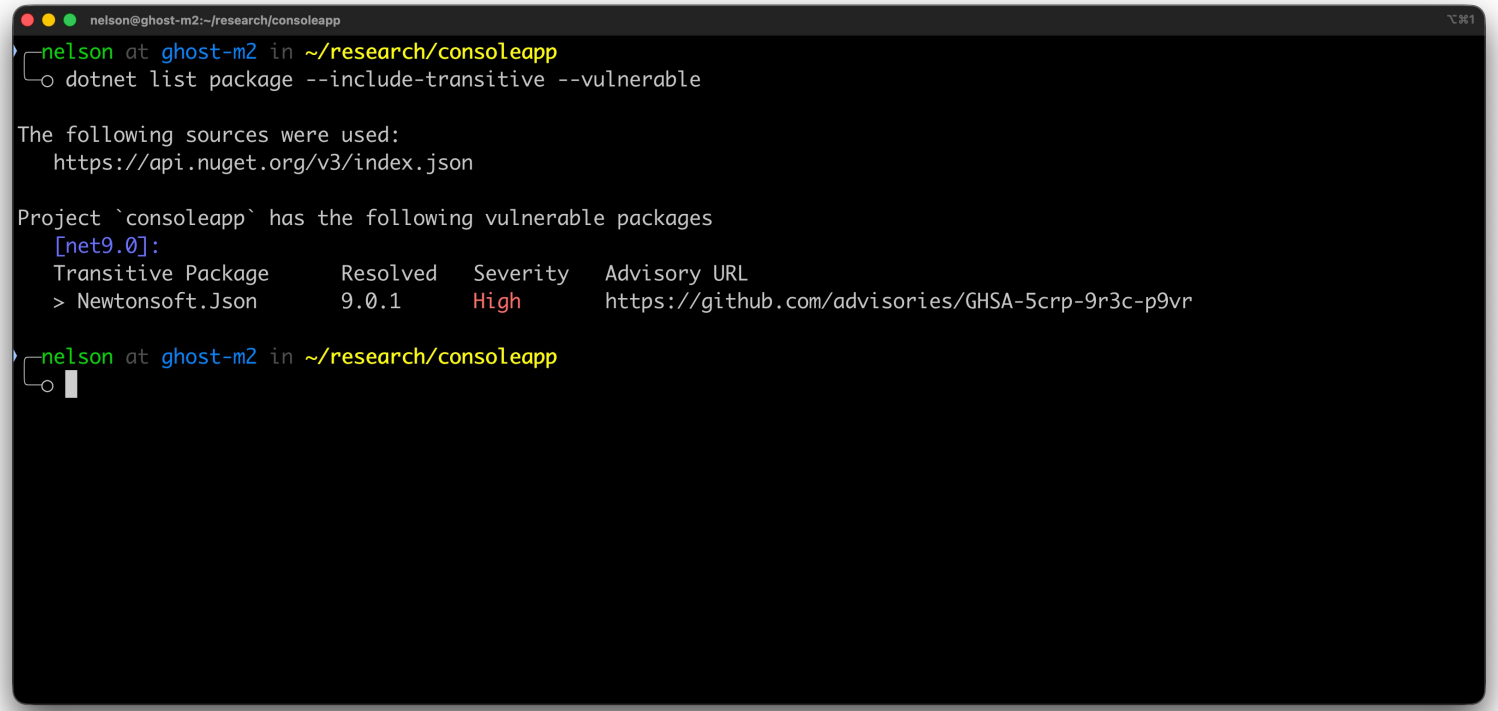
@niels.fennec.dev @nielstanis@infosec.exchange

DotNet CLI



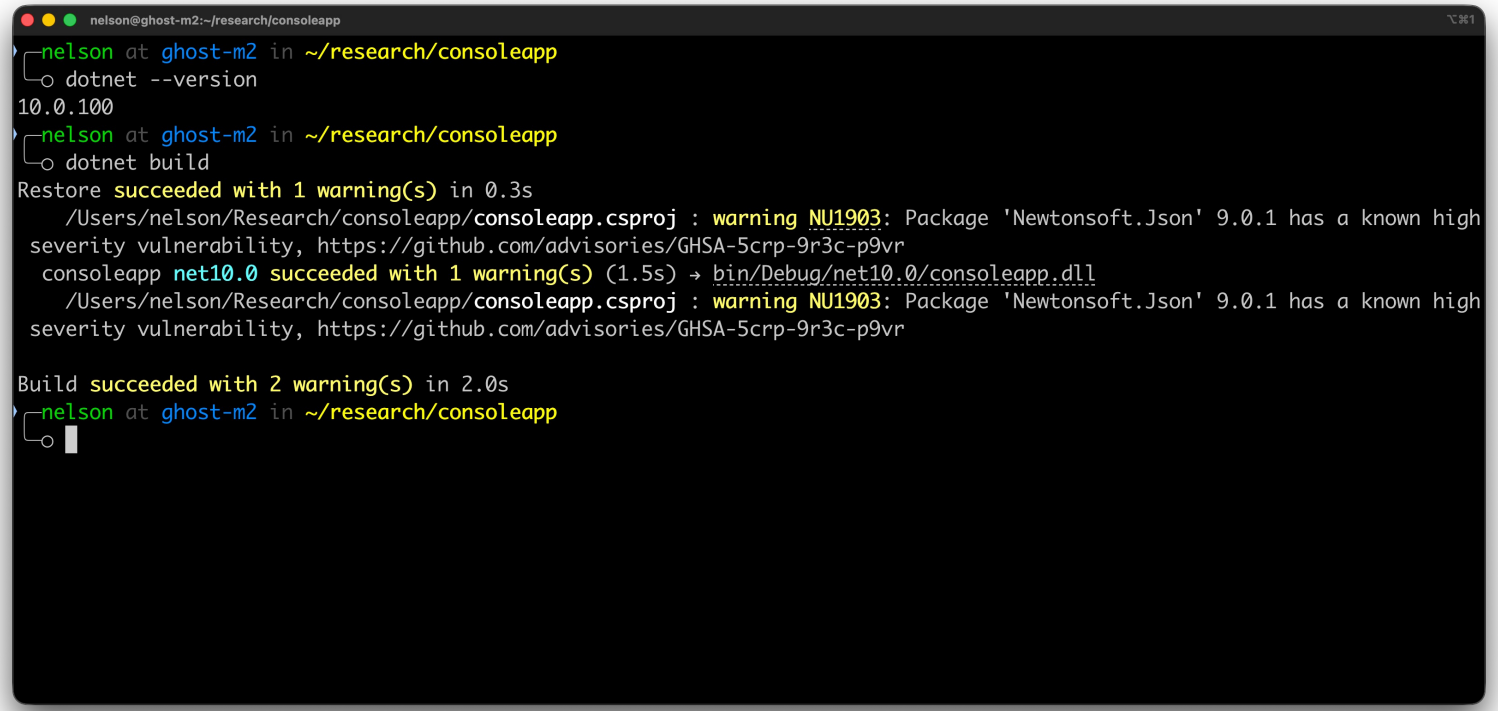
@niels.fennec.dev @nielstanis@infosec.exchange

DotNet CLI



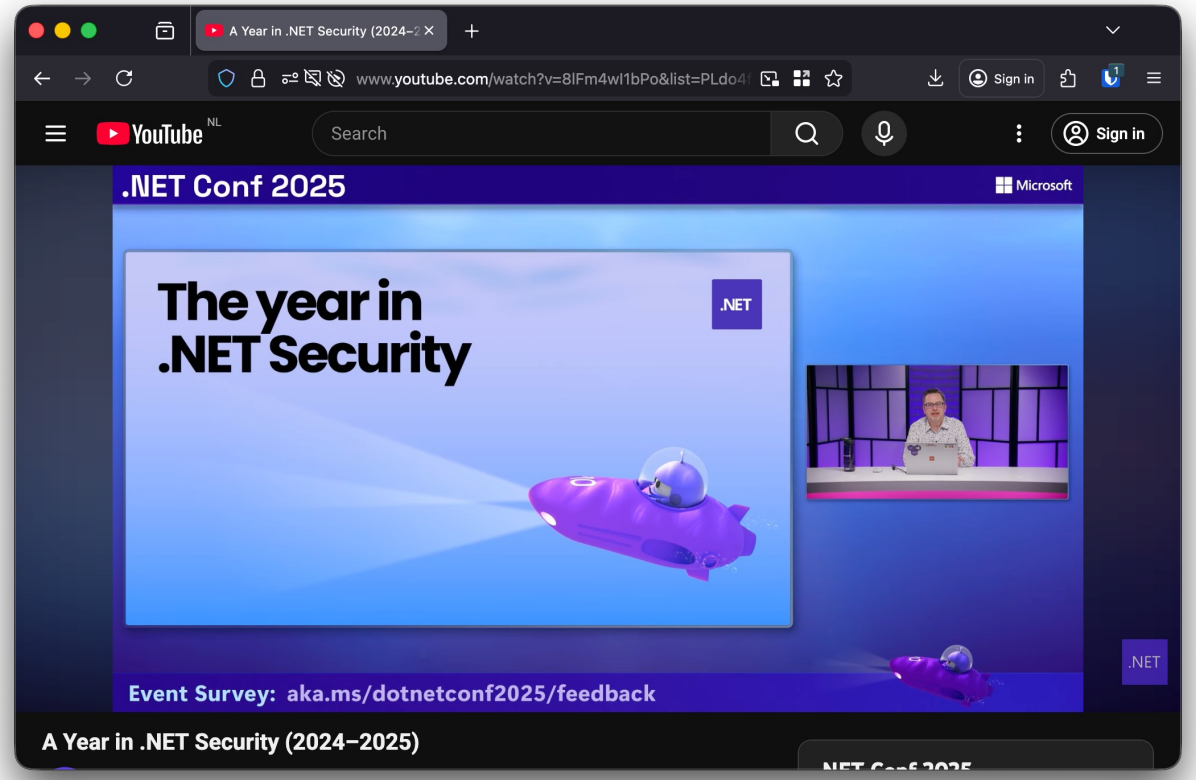
@niels.fennec.dev @nielstanis@infosec.exchange

DotNet CLI - .NET 10



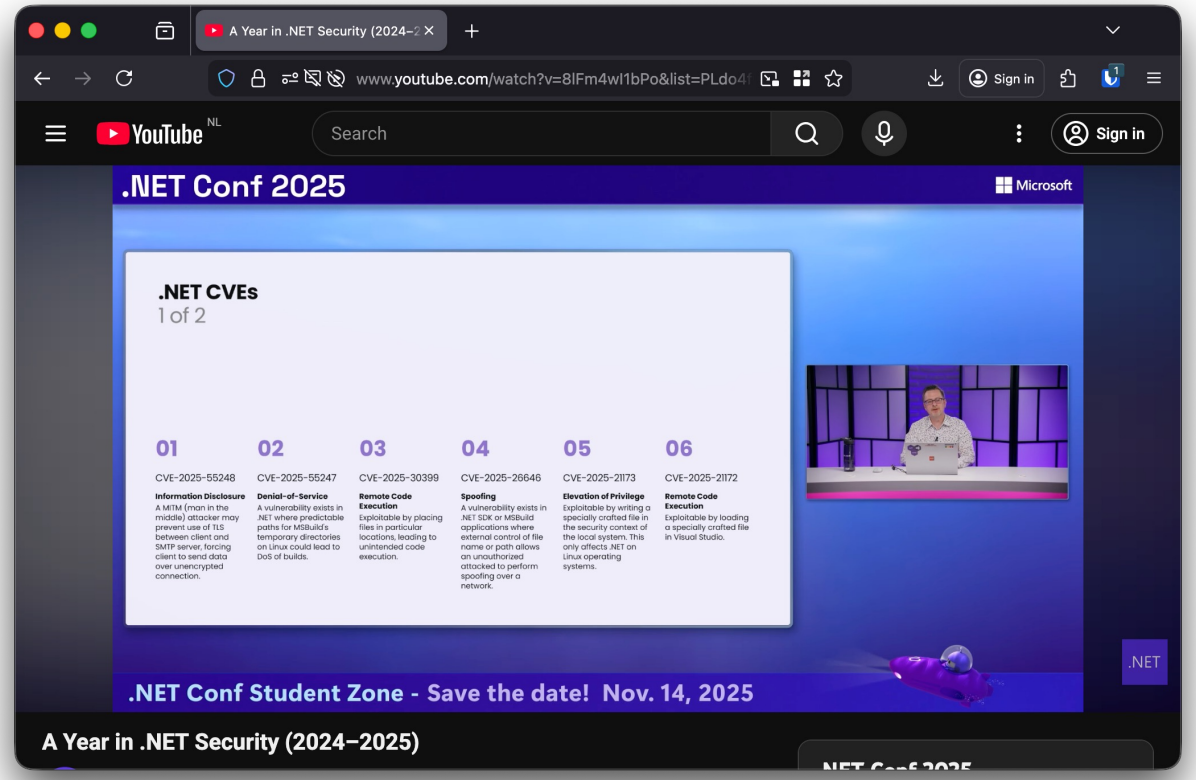
@niels.fennec.dev @nielstanis@infosec.exchange

.NET Conf 2025 – The year in .NET Security



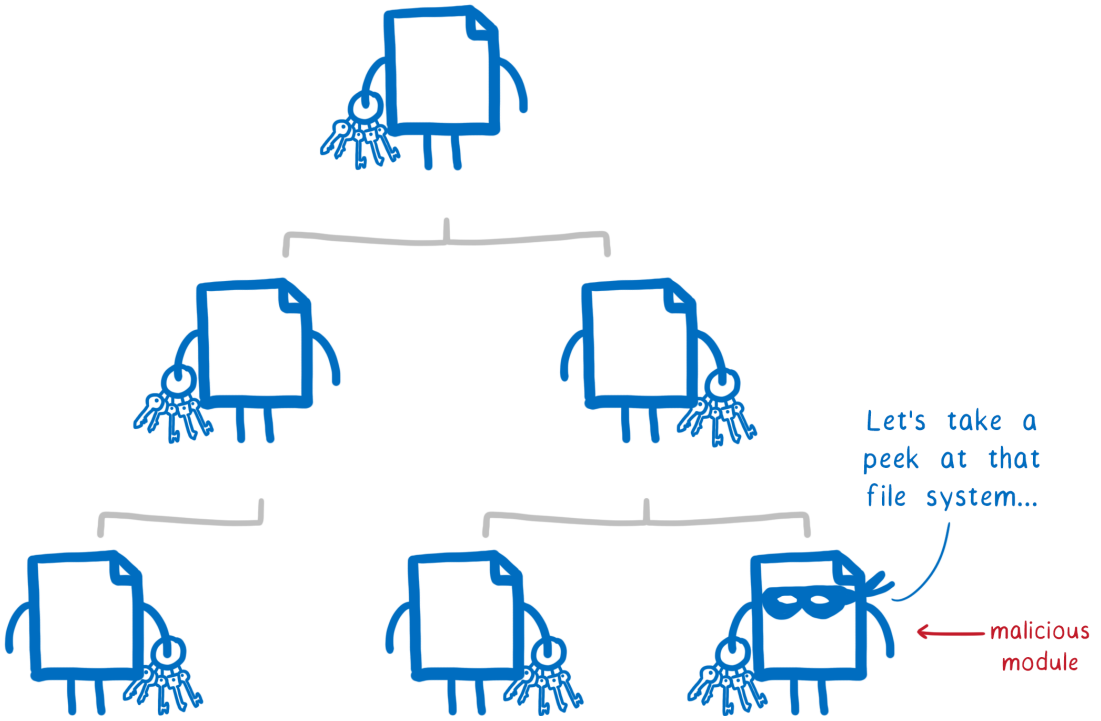
@niels.fennec.dev @nielstanis@infosec.exchange

.NET Conf 2025 – The year in .NET Security



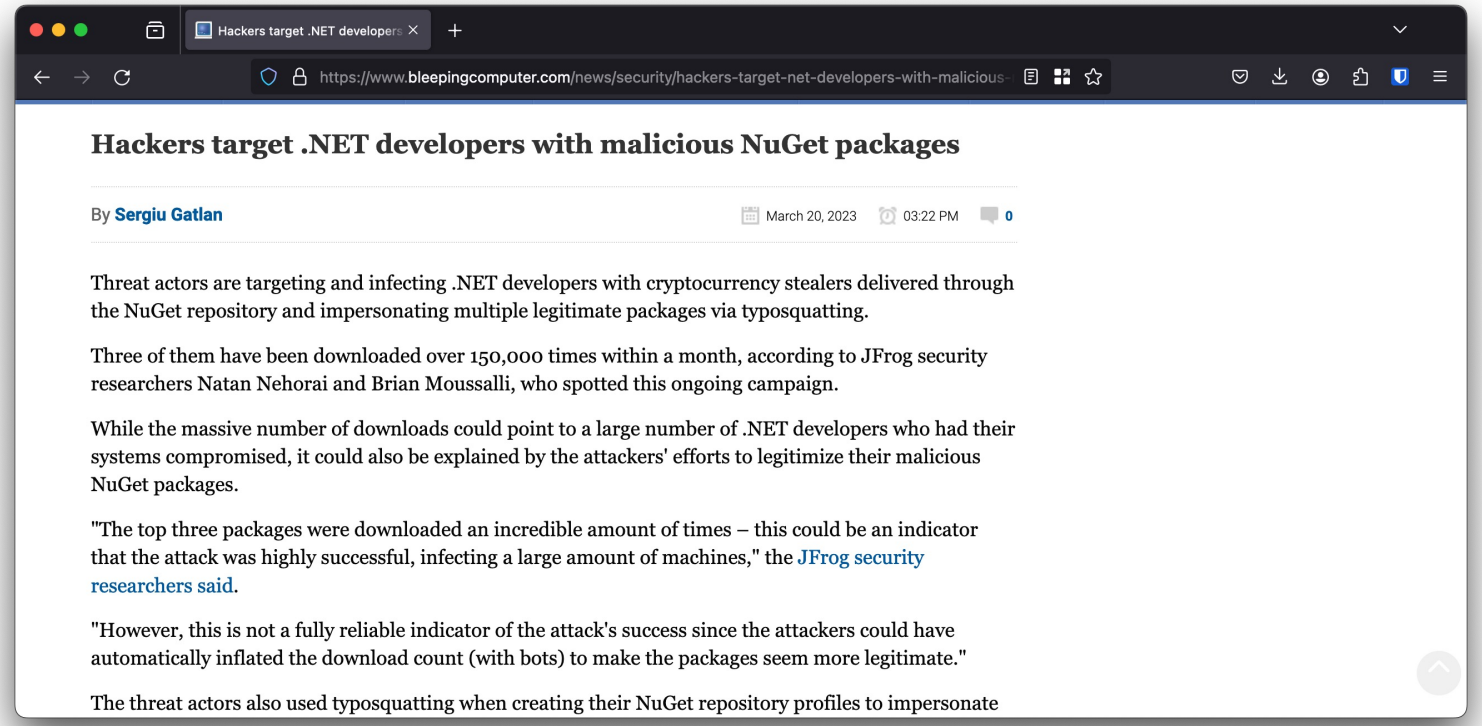
@niels.fennec.dev @nielstanis@infosec.exchange

Malicious Assembly



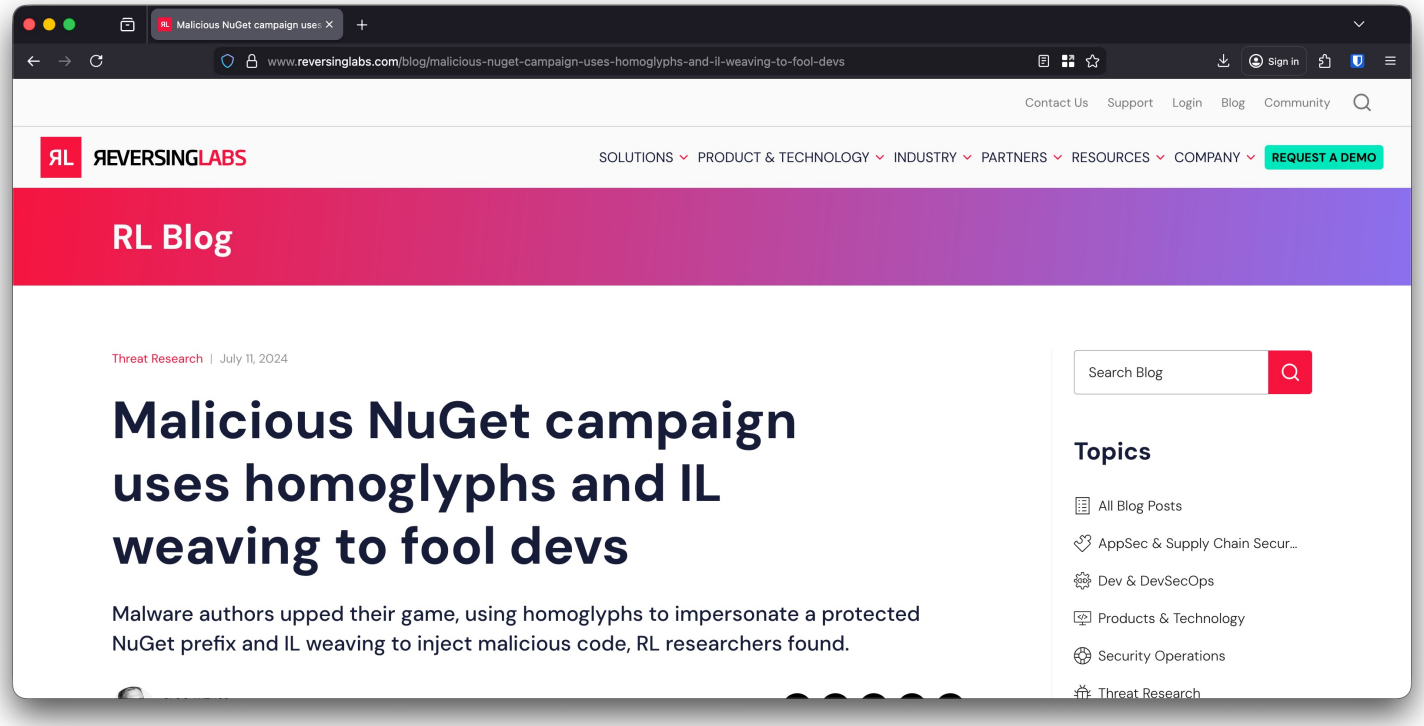
@niels.fennec.dev @nielstanis@infosec.exchange

Malicious Package



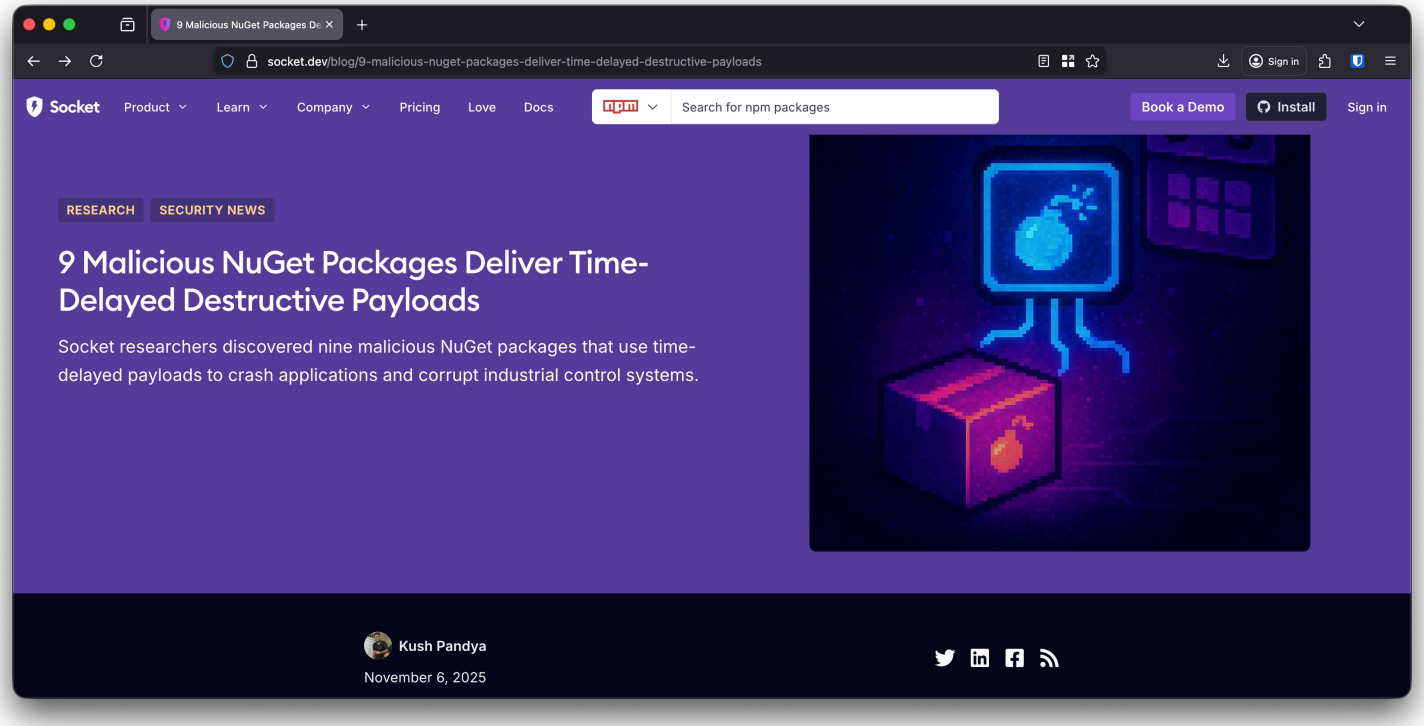
@niels.fennec.dev @nielstanis@infosec.exchange

Malicious Package



@niels.fennec.dev @nielstanis@infosec.exchange

Malicious Package



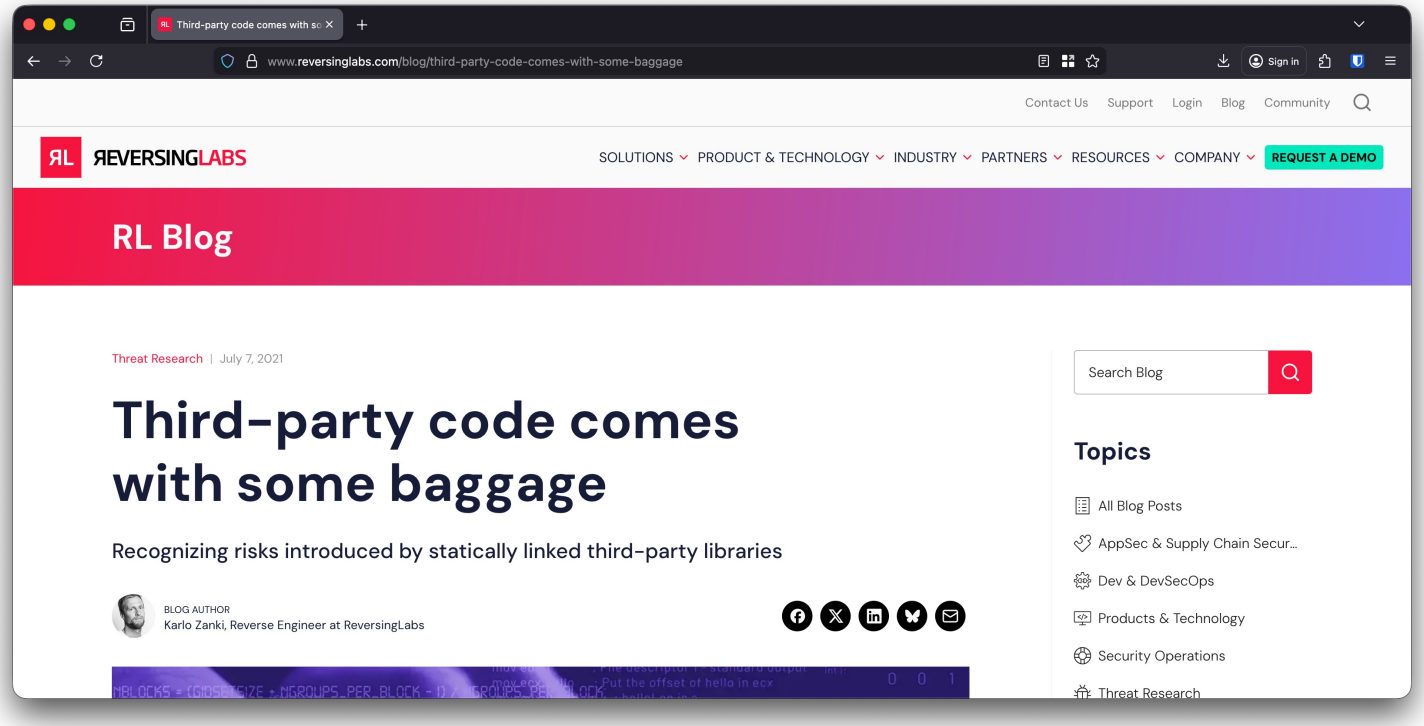
@niels.fennec.dev @nielstanis@infosec.exchange

Malicious Package

- Single publisher called **shanhai666**
- 99% code is valid & functional focusing on databases and industrial PLC's, malicious extension methods
- Delayed logic for Aug '27 & Nov '28
- 20% of database queries are terminated
- Sharp7Extend (typo squatting) for PLC, immediate activation mimics hardware failures

@niels.fennec.dev @nielstanis@infosec.exchange

Do you know what's inside?



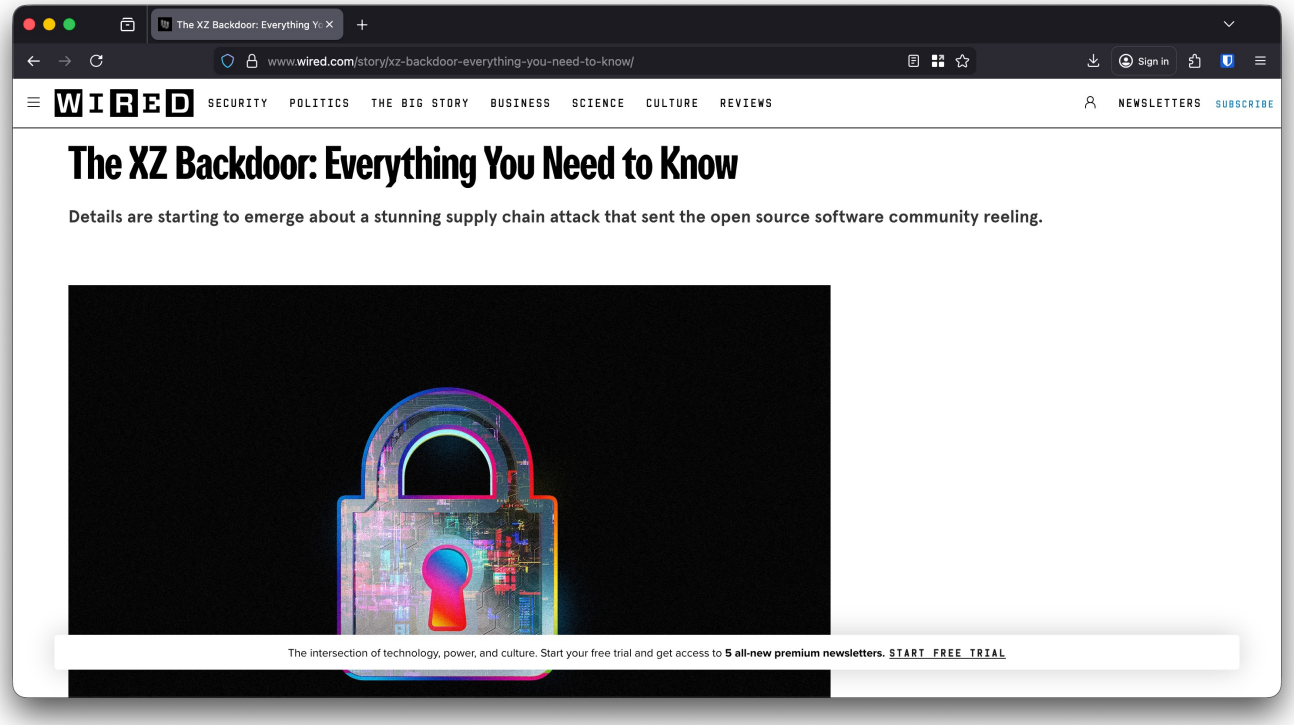
@niels.fennec.dev @nielstanis@infosec.exchange

Do you know what's inside?

- Unmanaged components interop
- Precompiled version of 7Zip, WinSCP and PuTTYgen
- Zlib compression library
  - Only source code distributed
  - Used in medical appliances
  - Severe vulnerabilities since 2005!

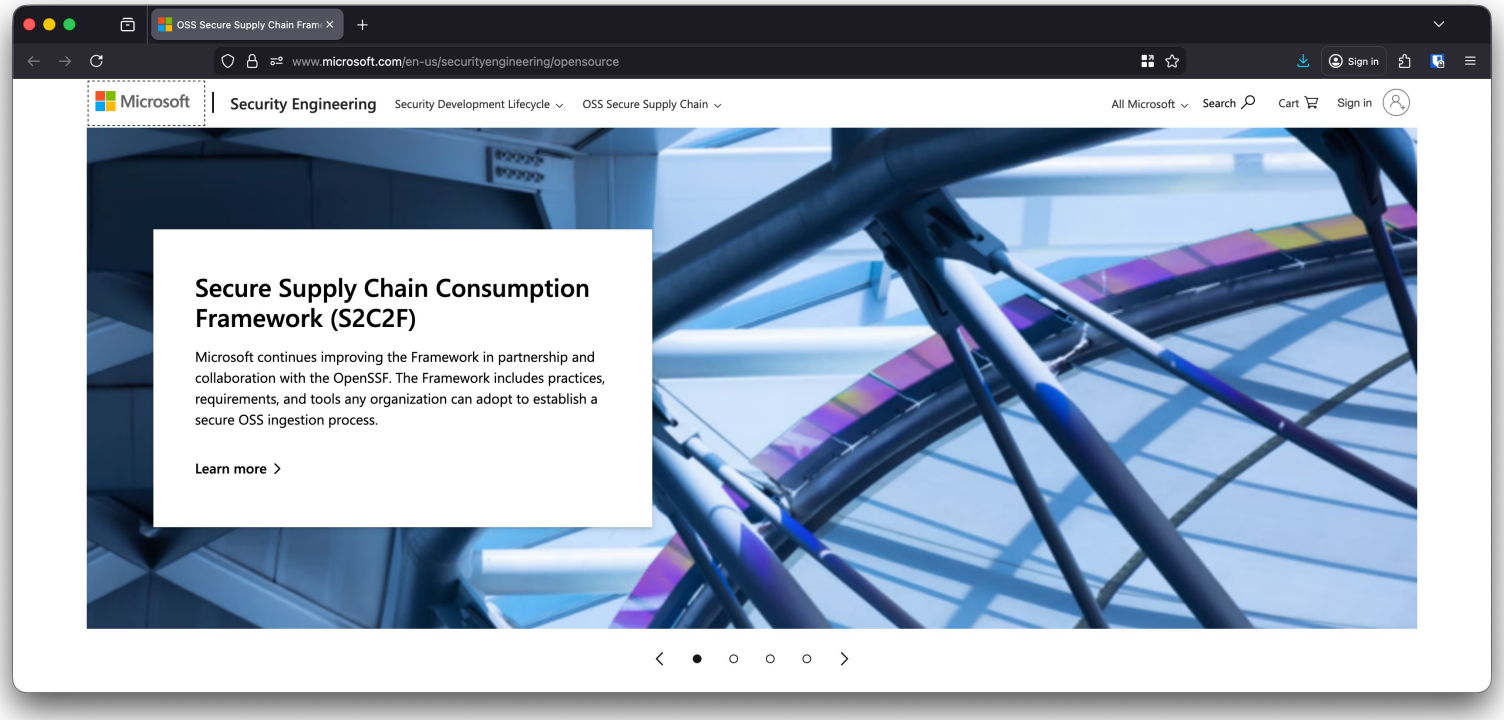
@niels.fennec.dev @nielstanis@infosec.exchange

Do you know what's inside? – xz utils



@niels.fennec.dev @nielstanis@infosec.exchange

Secure Supply Chain Consumption Framework (S2C2F)



@niels.fennec.dev @nielstanis@infosec.exchange

Secure Supply Chain Consumption Framework (S2C2F)

Level 1	Level 2	Level 3	Level 4
Minimum OSS Governance Program	**Secure Consumption and Improved MTTR**	**Malware Defense and Zero-Day Detection**	**Advanced Threat Defense**
• Use package managers [ING-1] • Local copy of artifact [ING-2] • Scan with known vulns [SCA-1] • Scan for software licenses [SCA-2] • Inventory OSS [INV-1] • Manual OSS updates [UPD-1]	• Scan for end of life [SCA-3] • Have an incident response plan [INV-2] • Auto OSS updates [UPD-2] • Alerts on vulns at PR time [UPD-3] • Audit that consumption is through approved ingestion method [AUD-2] • Validate integrity of OSS [AUD-3] • Secure package source file configuration [ENF-1]	• Deny list capability [ING-3] • Clone OSS source [ING-4] • Scan for malware [SCA-4] • Proactive security reviews [SCA-5] • Enforce OSS provenance [AUD-1] • Enforce consumption from curated feed [ENF-2]	• Validate the SBOMs of OSS consumed [AUD-4] • Rebuild OSS on trusted infrastructure [REB-1] • Digitally sign rebuilt OSS [REB-2] • Generate SBOM for rebuilt OSS [REB-3] • Digitally sign protected SBOMs [REB-4] • Implement fixes [FIX-1]

@niels.fennec.dev @nielstanis@infosec.exchange

Hot take; dependency cooldowns

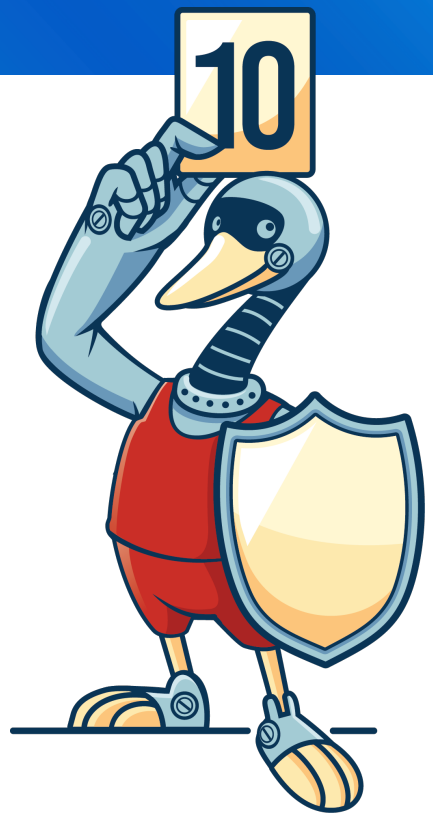
@niels.fennec.dev @nielstanis@infosec.exchange

What's inside? (Audit)

@niels.fennec.dev @nielstanis@infosec.exchange

Open SSF Security Scorecard

- Set Automated Security Checks
- Scoring System 0-10
- Security Best Practices
- Ease of Use
- Community Support



@niels.fennec.dev @nielstanis@infosec.exchange

Maintenance

- Vulnerabilities (**High**)
  - Does the project have unfixed vulnerabilities?
- Maintained (**High**)
  - Is the project at least 90 days old, and maintained?
- Security Policy (**Medium**)
  - Does project contain security policy?

@niels.fennec.dev @nielstanis@infosec.exchange

Maintenance

- License (Low)
  - Does the project declare a licence?
- CII Best Practices (Low)
  - Does the project have a CII Best Practices Badge?
- Dependency Update Tool (High)
  - Does the project use tools to help update its dependencies e.g. Dependabot, RenovateBot?

@niels.fennec.dev @nielstanis@infosec.exchange

Continous Testing

- Continuous Integration Tests (Low)
  - Does the project run tests in CI?
- Fuzzing (Medium)
  - Does the project use fuzzing tools, e.g. OSS-Fuzz?
- Static Analysis (Medium)
  - Does the project use static code analysis tools?

@niels.fennec.dev @nielstanis@infosec.exchange

Source Risk Assessment

- Binary Artifacts (**High**)
  - Does repository contain binaries/compiled artifacts?
- Branch Protection (**High**)
  - GitHub best practices branch protection
- Dangerous Workflow (**Critical**)
  - GitHub best practices for Workflow Actions

@niels.fennec.dev @nielstanis@infosec.exchange

Source Risk Assessment

- Code Review (**High**)
  - Does the project require code review before code is merged?
- Contributors (Low)
  - Does the project have contributors from multiple organizations?

@niels.fennec.dev @nielstanis@infosec.exchange

Build Risk Assessment

- Pinned Dependencies (Medium)
  - Does the project declare and pin dependencies?
- Token Permission (High)
  - Does the project declare GitHub workflow tokens as read only?

@niels.fennec.dev @nielstanis@infosec.exchange

Build Risk Assessment

- Packaging (Medium)
  - Does the project build and publish official packages from CI/CD?
- Signed Releases (High)
  - Does the project cryptographically sign releases?

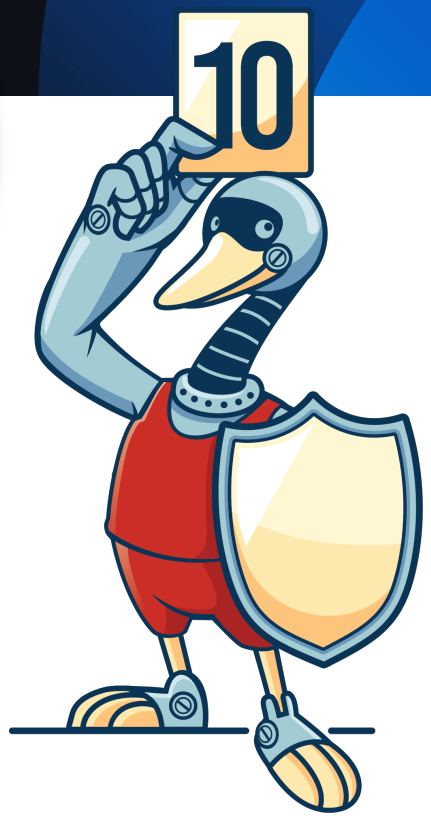
@niels.fennec.dev @nielstanis@infosec.exchange

Demo OpenSSF Scorecard

```
Running checks
```


@niels.fennec.dev @nielstanis@infosec.exchange

Fennec Labs Scorecard



NDC { Manchester }

@niels.fennec.dev @nielstanis@infosec.exchange

.NET Reproducibility

- Reproducible builds
  - Independent path from source to binary code.
- Roslyn Compiler Deterministic Compilation
- How reproducible is a simple console app?

@niels.fennec.dev @nielstanis@infosec.exchange

Fennec Labs Reproduce

NDC { Manchester }

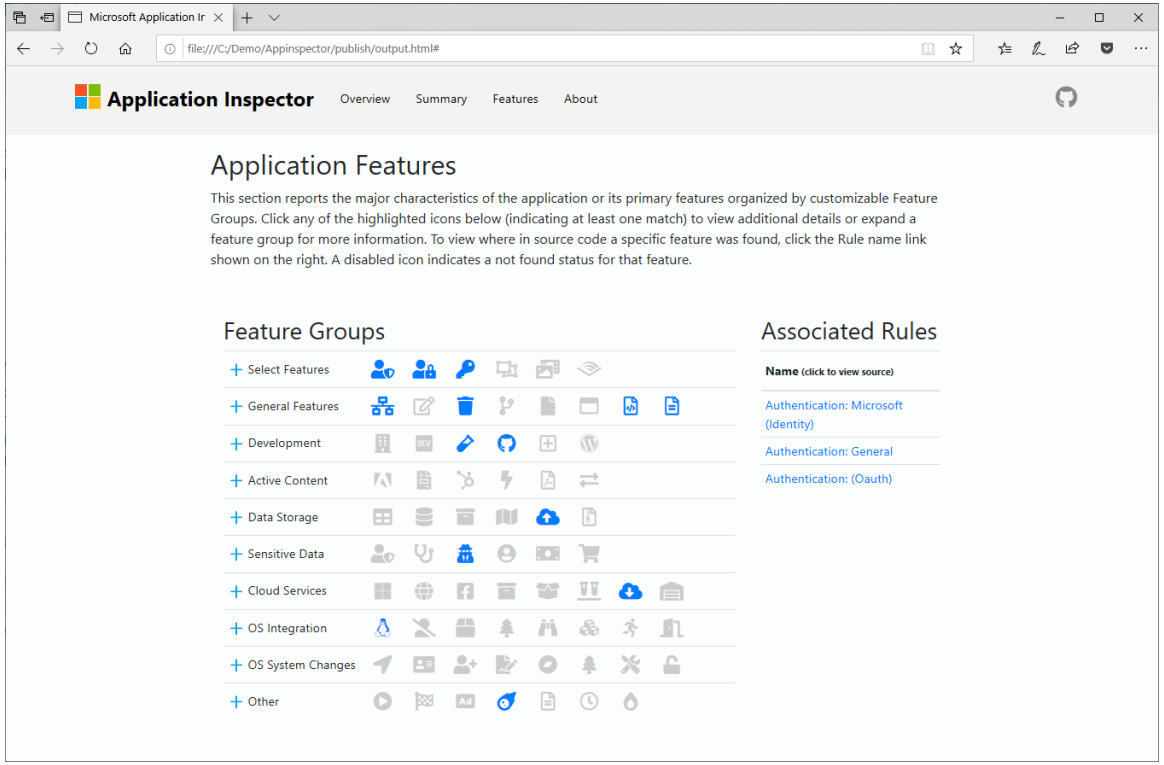
@niels.fennec.dev @nielstanis@infosec.exchange

Fennec Labs Instrument

NDC { Manchester }

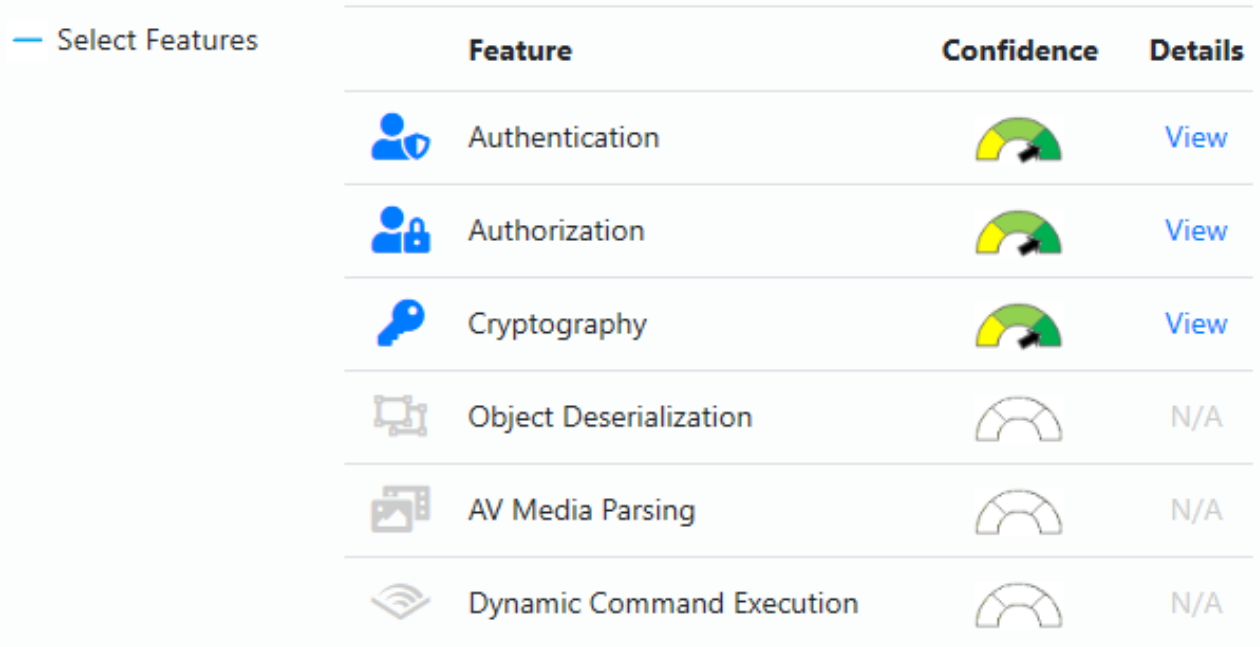
@niels.fennec.dev @nielstanis@infosec.exchange

Application Inspector



@niels.fennec.dev @nielstanis@infosec.exchange

Application Inspector



@niels.fennec.dev @nielstanis@infosec.exchange

Fennec Labs Compare

NDC { Manchester }

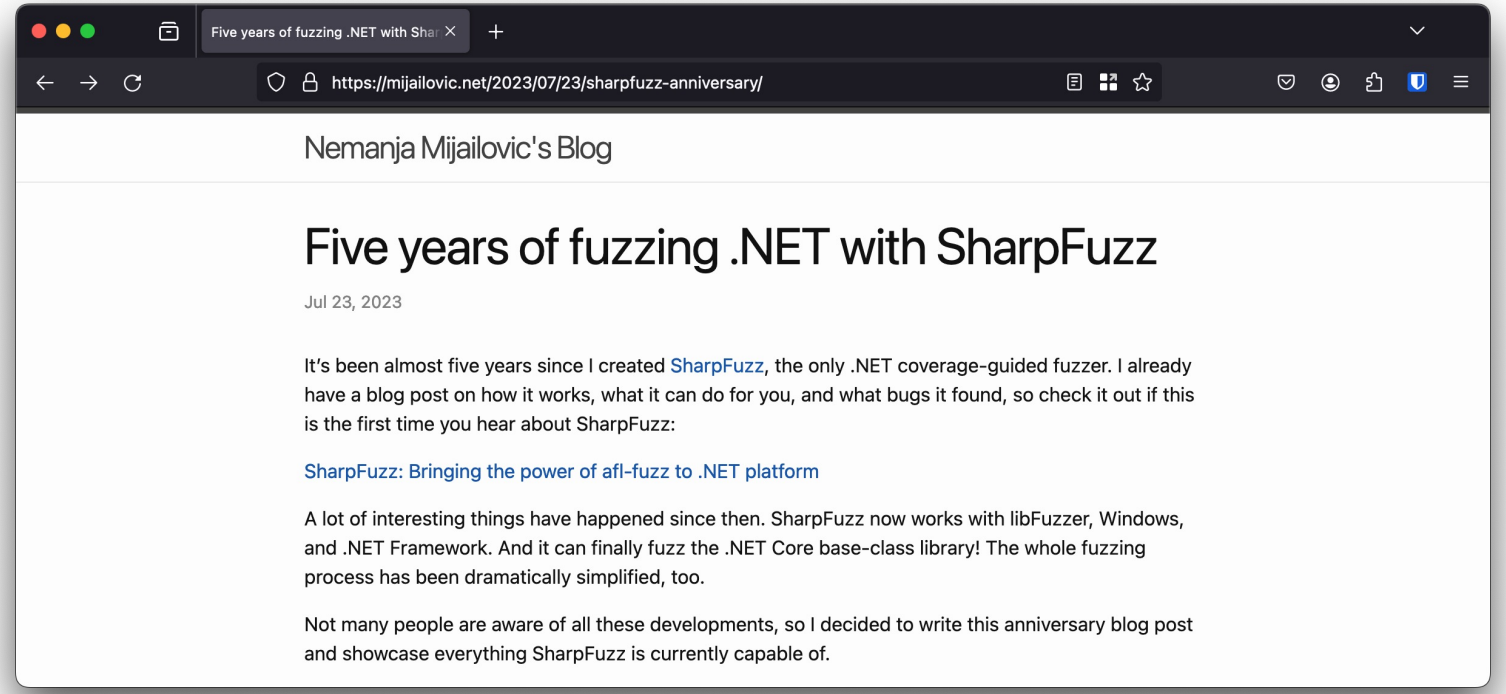
@niels.fennec.dev @nielstanis@infosec.exchange

Future Idea - Fuzzing

- Fuzzing, or fuzz testing
  - Automated software testing method that uses a wide range of ***invalid*** and unexpected data as input to find flaws
- Good in C/C++ memory issues
- Can it be of any value with managed languages like .NET?

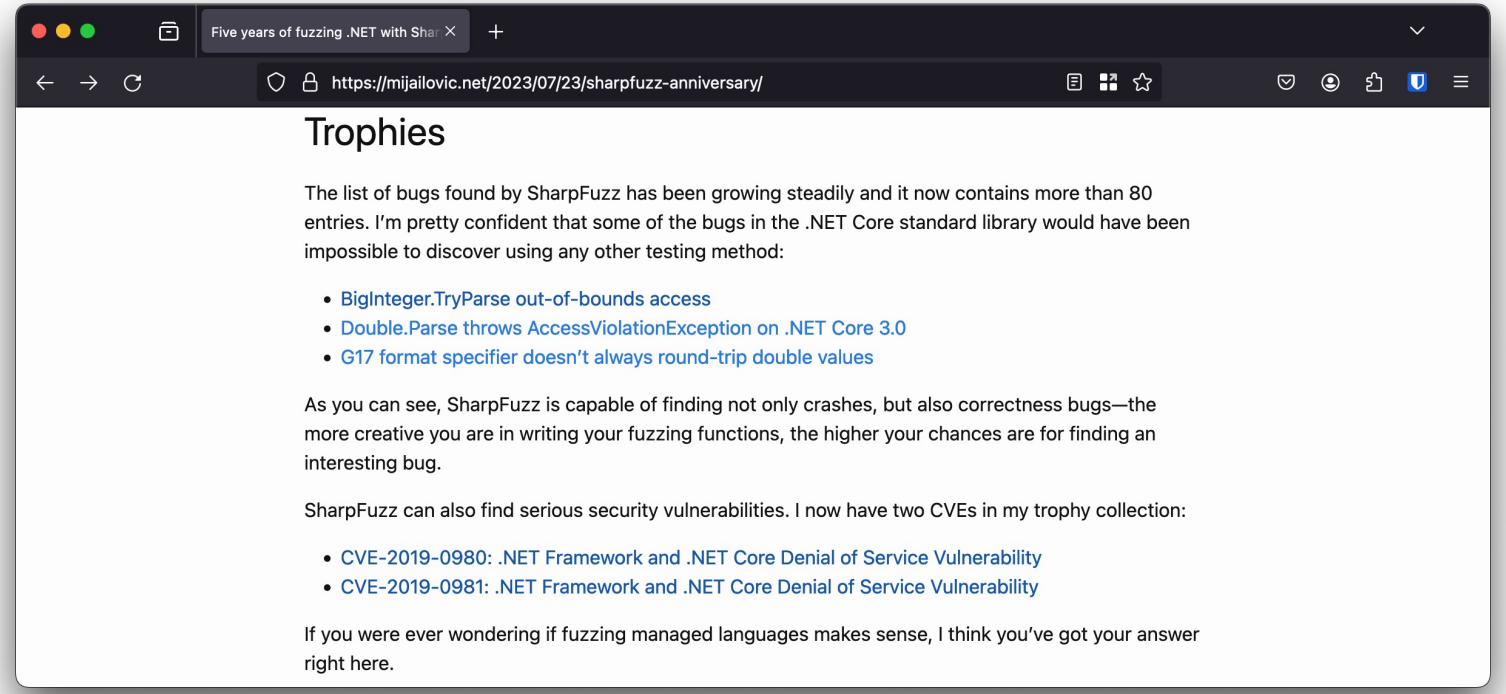
@niels.fennec.dev @nielstanis@infosec.exchange

Fuzzing .NET & SharpFuzz



@niels.fennec.dev @nielstanis@infosec.exchange

Fuzzing .NET & SharpFuzz



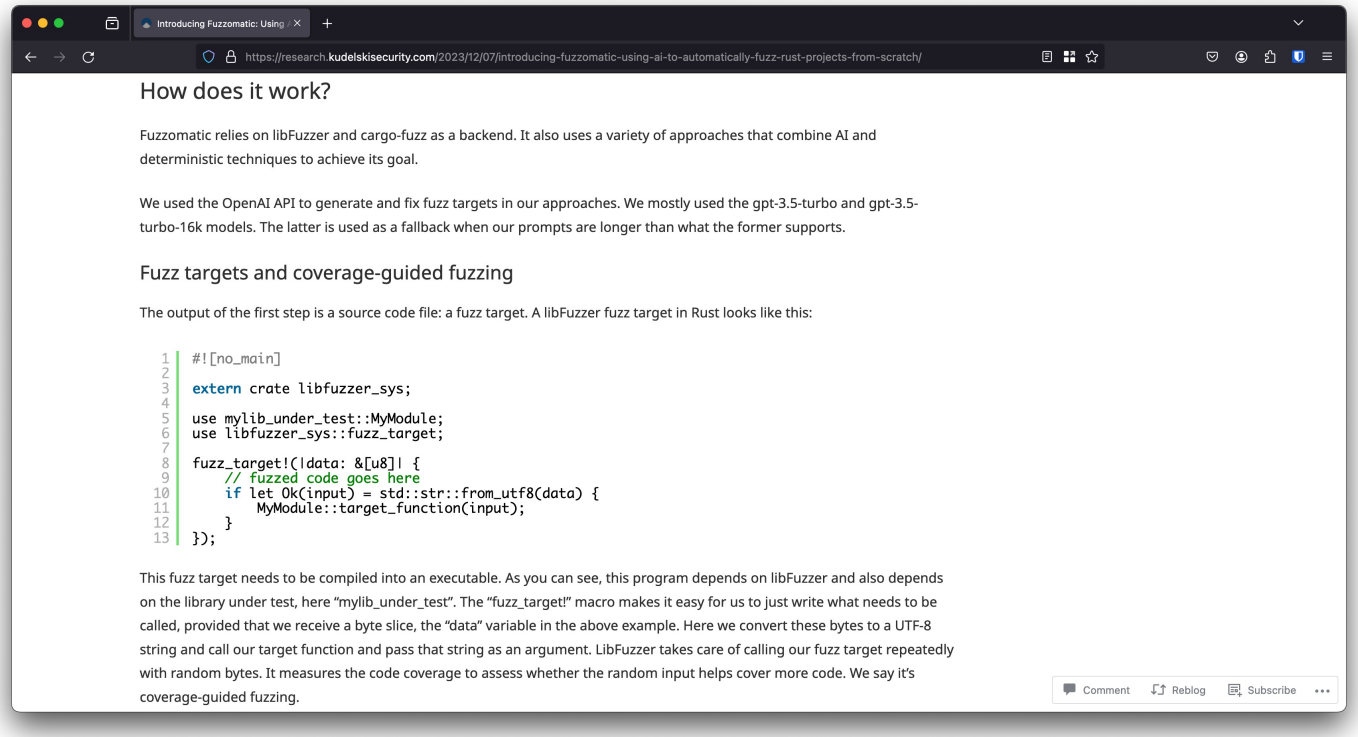
@niels.fennec.dev @nielstanis@infosec.exchange

Fuzzing .NET – Jil JSON Serializer

```
public static void Main(string[] args)
{
      SharpFuzz.Fuzzer.OutOfProcess.Run(stream => {
      try
      {
            using (var reader = new
                  System.IO.StreamReader(stream))
            JSON.DeserializeDynamic(reader);
      }
      catch (DeserializationException) { }
      });
}
```


@niels.fennec.dev @nielstanis@infosec.exchange

Fuzzomatic: Using AI to Fuzz Rust



@niels.fennec.dev @nielstanis@infosec.exchange

Future Idea – Dashboard & Feedback

- Local dashboard for project
- Remote dashboard for community
- Ability to do review and share security related data of project
- MCP server with embedded prompt resources to help security review
- Roslyn analyzer that can help guide the developer

@niels.fennec.dev @nielstanis@infosec.exchange

Conclusion

- Doing software security is hard and Software Security Professionals don’t scale!
- Package eco-systems have been wild west lately
- Goal is to have Fennec Labs help with reviewing and sharing security details for all audiences
- Allowing everyone to make better informed decisions and can act if needed.

@niels.fennec.dev @nielstanis@infosec.exchange

Thank you!

- ntanis at Veracode.com
- https://github.com/nielstanis/ndcmanchester2025
- Questions?

@niels.fennec.dev @nielstanis@infosec.exchange