VERACODE

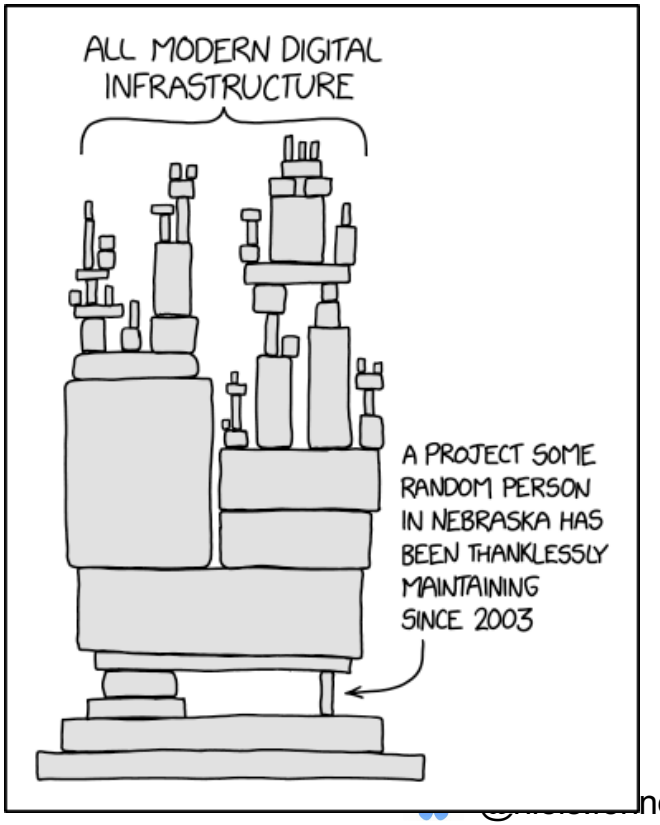
Reviewing NuGet Packages security easily using OpenSSF Scorecard

Niels Tanis
Sr. Principal Security Researcher

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

Modern Application Architecture
XKCD 2347



elia group

nec.dev @nielstanis@infosec.exchange

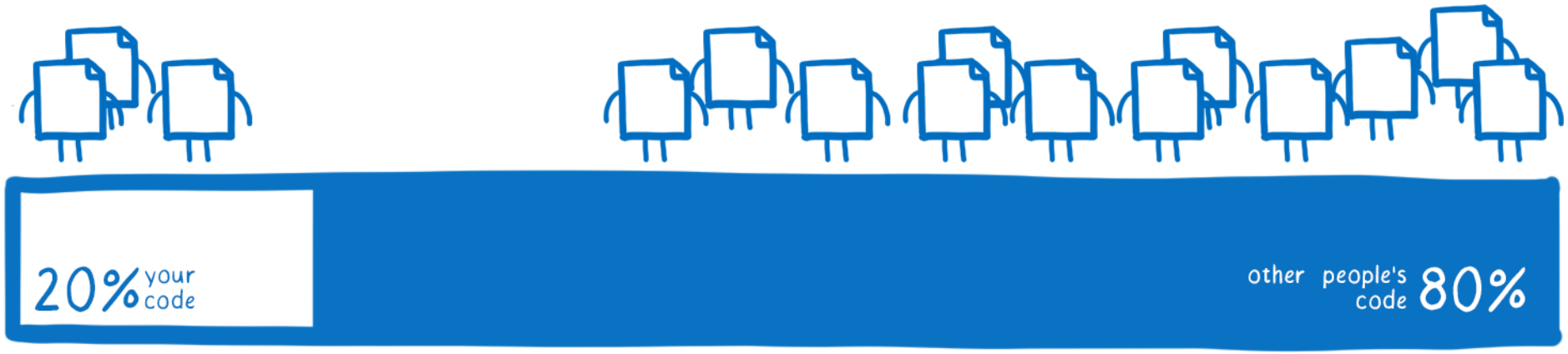
Agenda

- Risks in 3rd party NuGet Packages
- OpenSFF Scorecard
- Measure, New & Improved
- Conclusion - Q&A

elia group

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

Average codebase composition

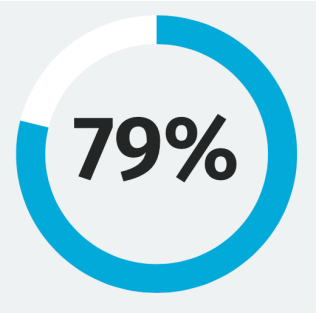


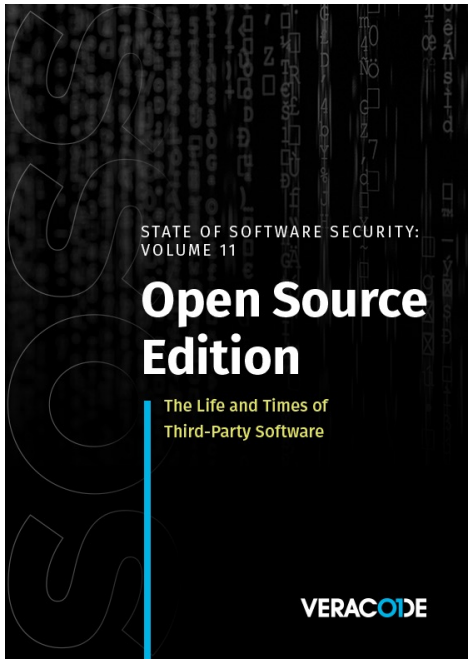
elia group

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

State of Software Security v11

”Despite this dynamic landscape, 79 percent of the time, developers never update third-party libraries after including them in a codebase.”





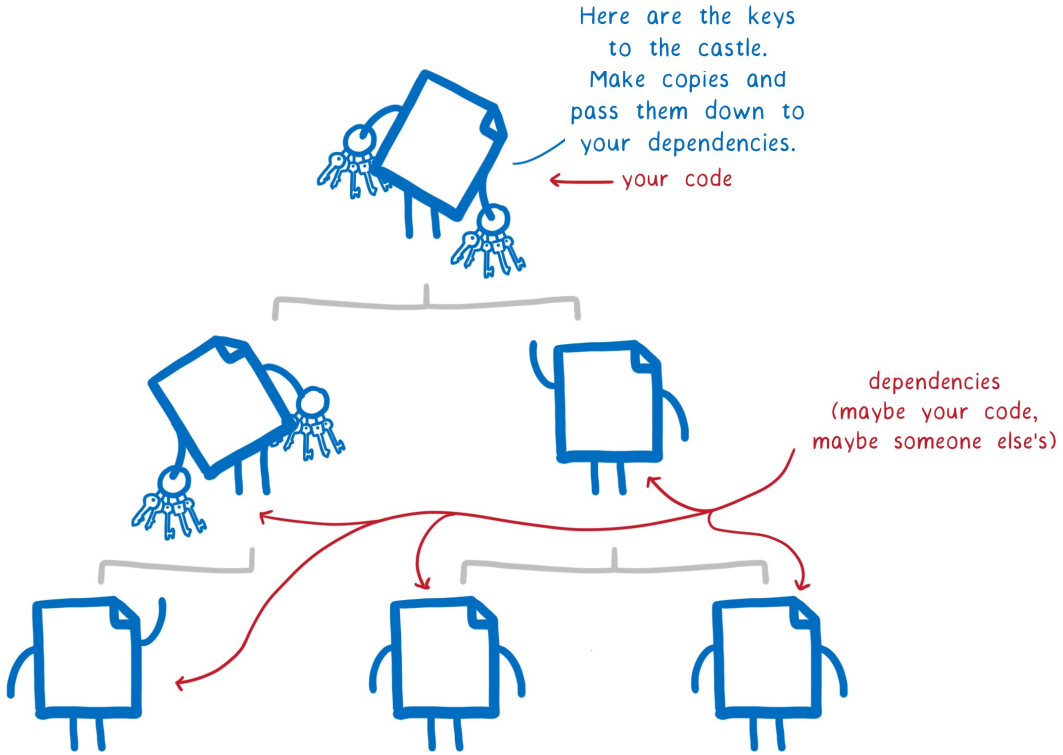
elia group

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

State of Log4j – 2 years later

- Analysed our data August-November 2023
  - Total set of almost 39K unique applications scanned
- 2.8% run version vulnerable to Log4Shell
- 3.8% run version patched but vulnerable to other CVE
- 32% rely on a version that's end-of-life and have no support for any patches.

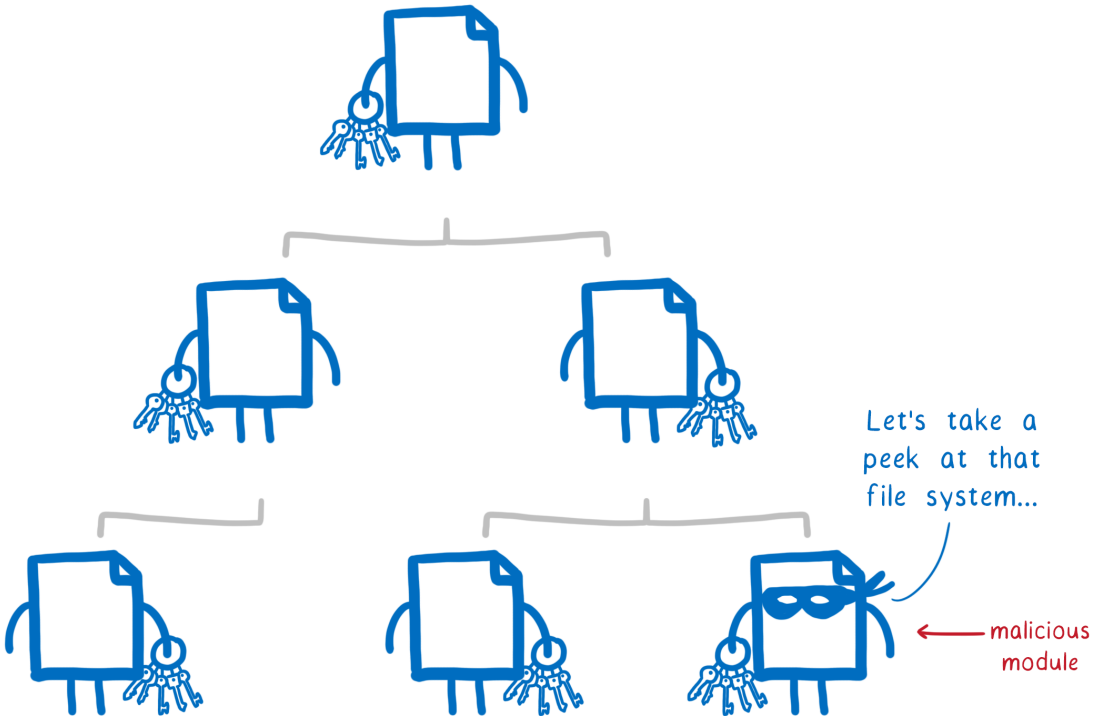
Average codebase composition



elia group

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

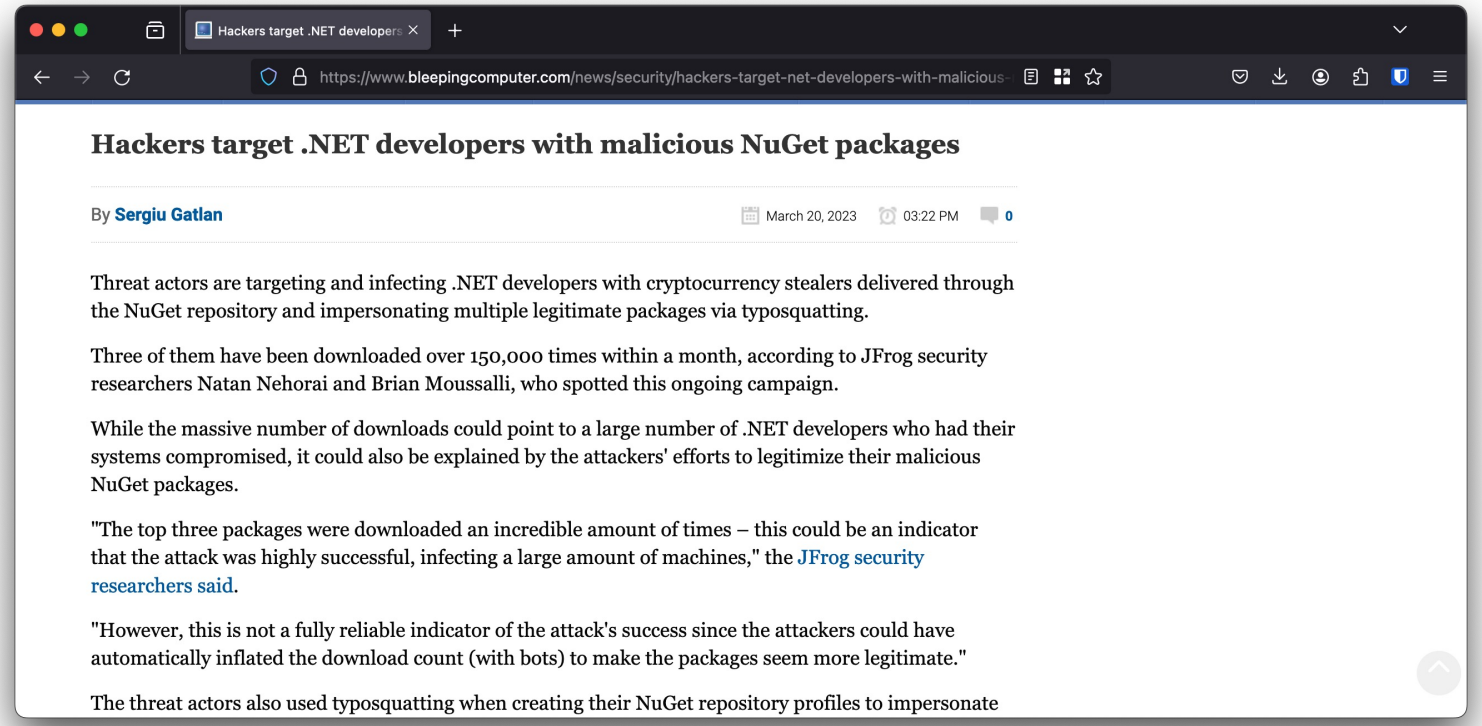
Malicious Assembly



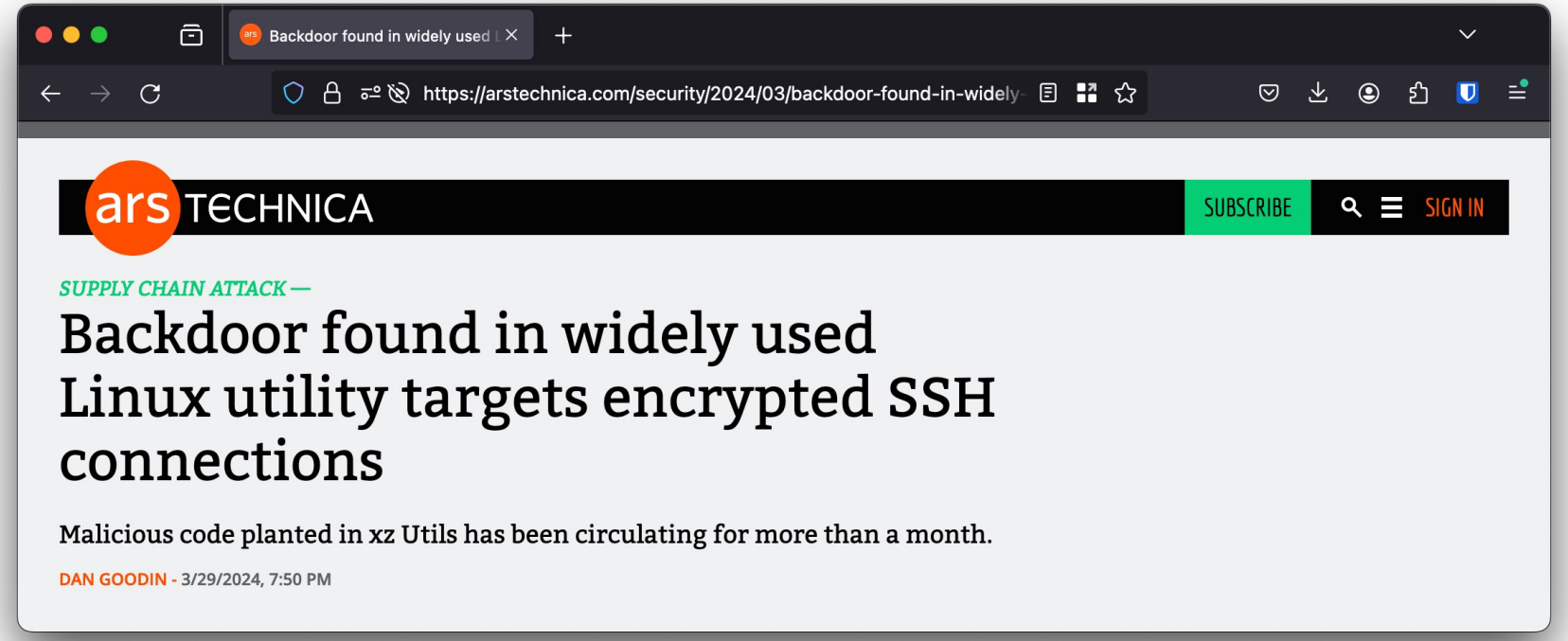
elia group

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

Malicious Package



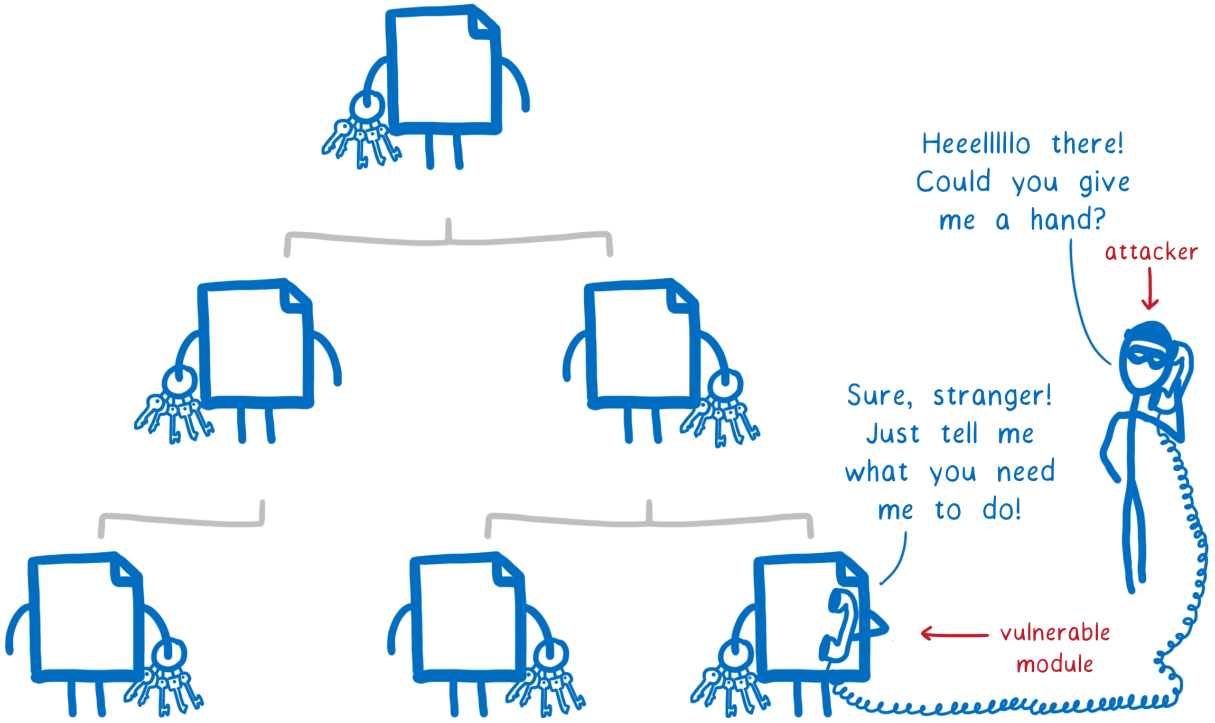
XZ Backdoor



elia group

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

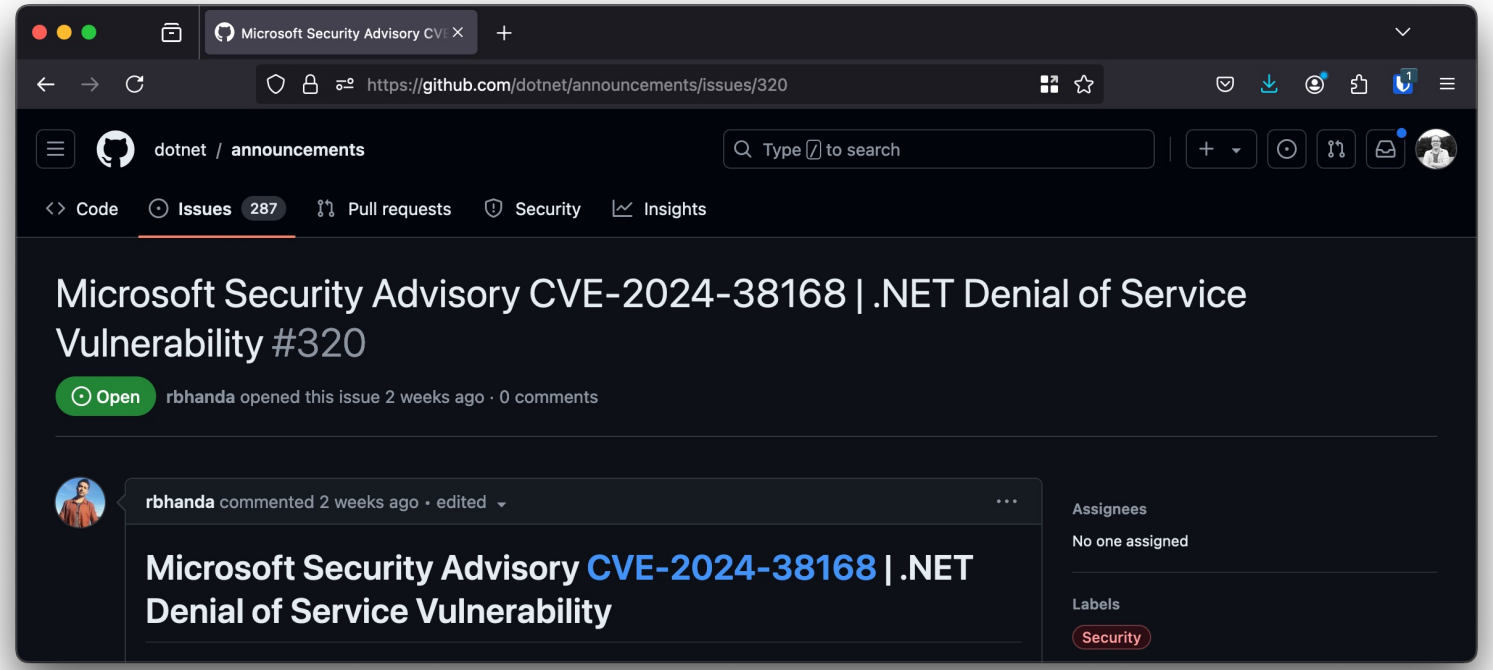
Vulnerable Assembly



elia group

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

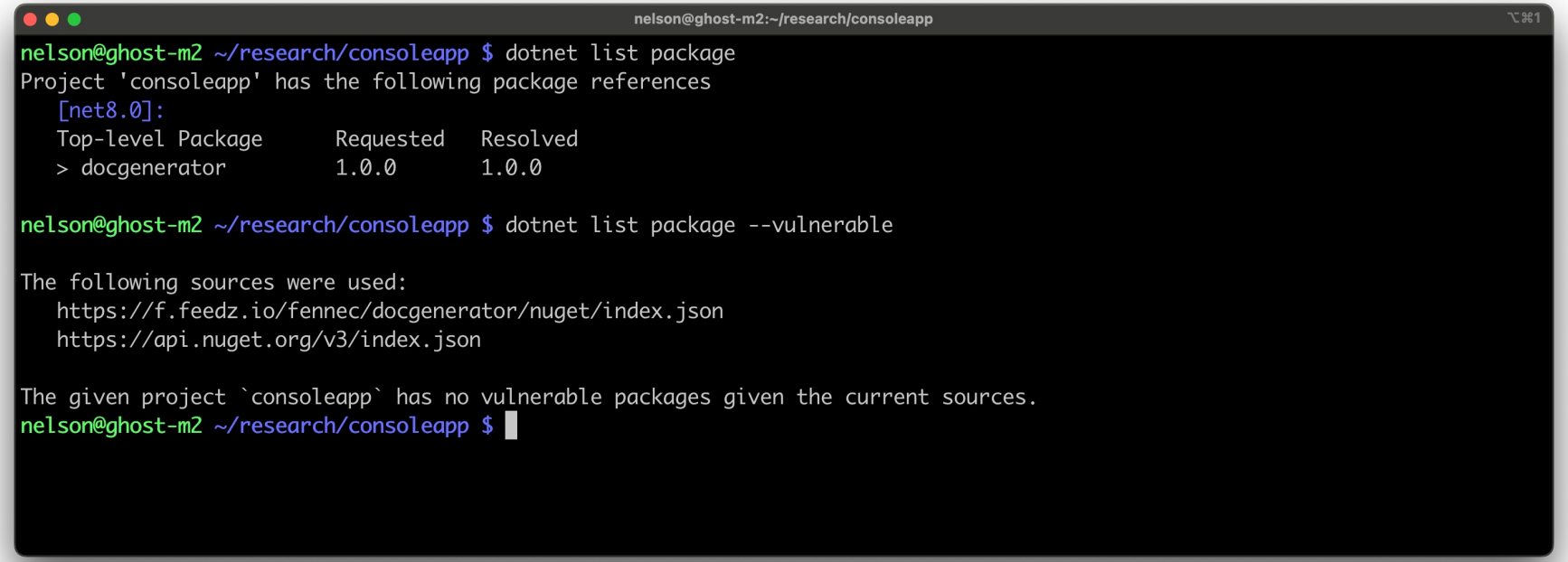
Vulnerabilities in Libraries



elia group

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

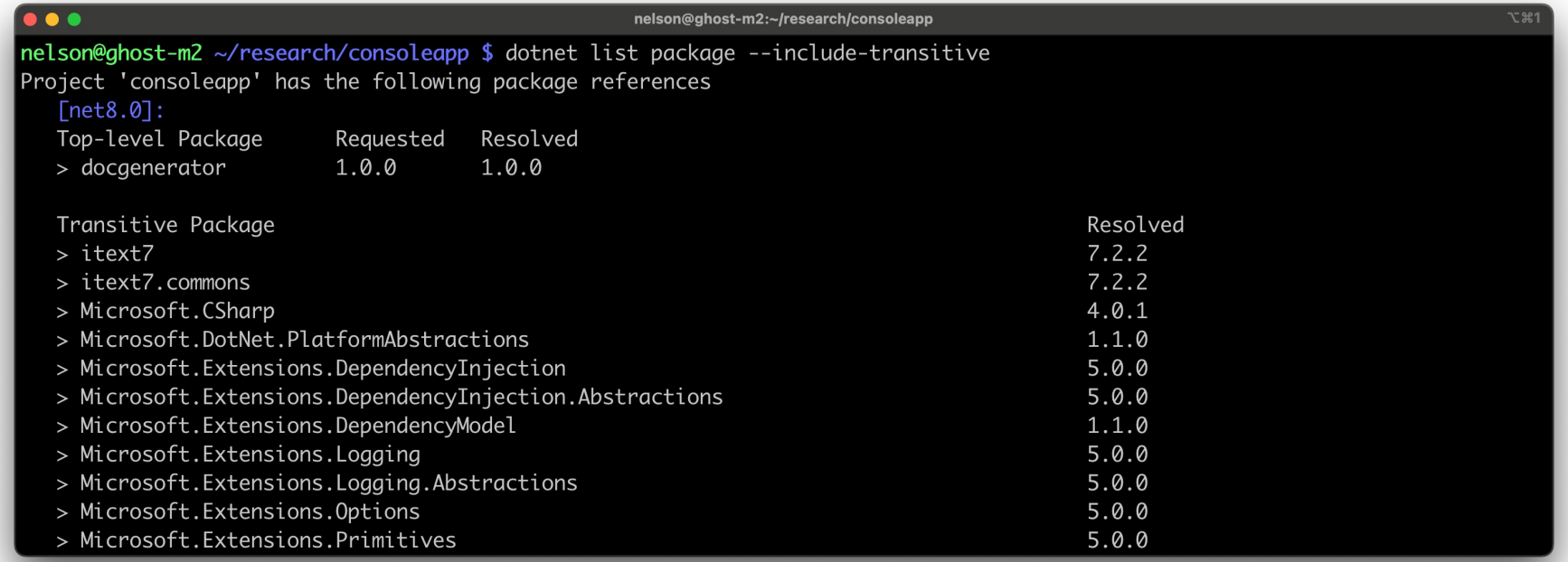
DotNet CLI



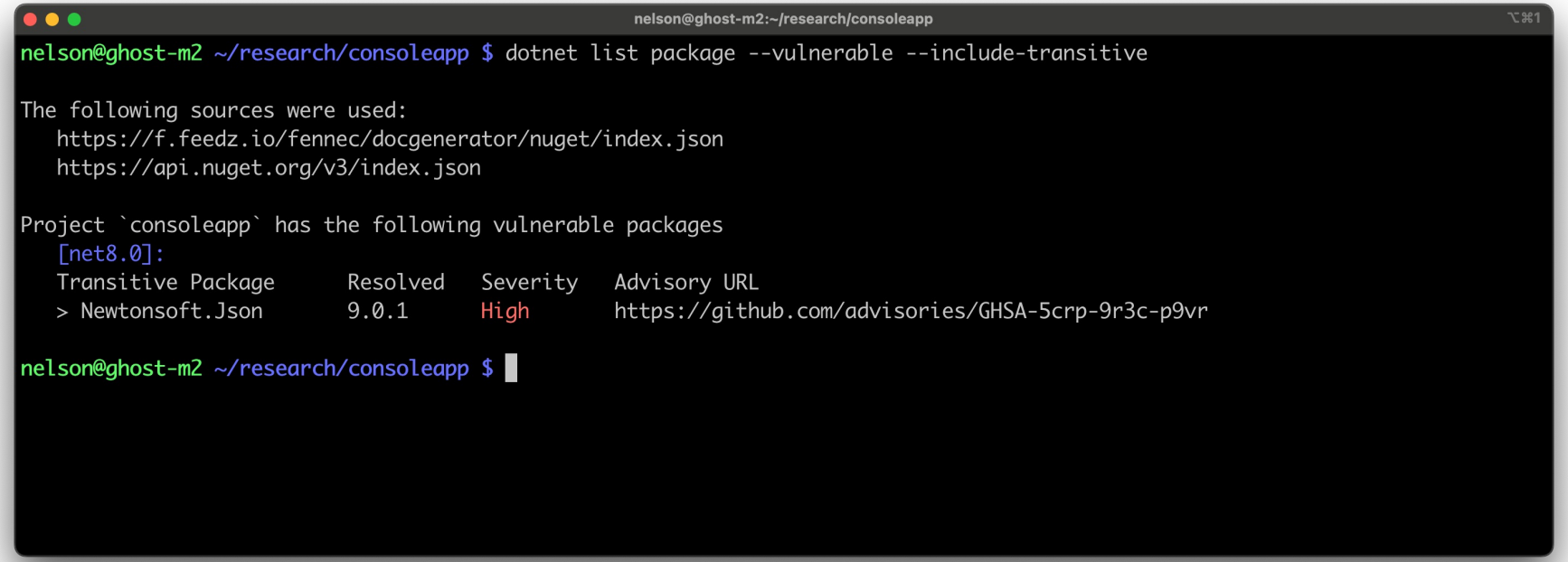
elia group

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

DotNet CLI


nelson@ghost-m2:~/research/consoleapp
nelson@ghost-m2 ~/research/consoleapp $ dotnet list package --include-transitive
Project 'consoleapp' has the following package references
[net8.0]:
Top-level Package Requested Resolved
> docgenerator 1.0.0 1.0.0
Transitive Package Resolved
> itext7 7.2.2
> itext7.commons 7.2.2
> Microsoft.CSharp 4.0.1
> Microsoft.DotNet.PlatformAbstractions 1.1.0
> Microsoft.Extensions.DependencyInjection 5.0.0
> Microsoft.Extensions.DependencyInjection.Abstractions 5.0.0
> Microsoft.Extensions.DependencyModel 1.1.0
> Microsoft.Extensions.Logging 5.0.0
> Microsoft.Extensions.Logging.Abstractions 5.0.0
> Microsoft.Extensions.Options 5.0.0
> Microsoft.Extensions.Primitives 5.0.0

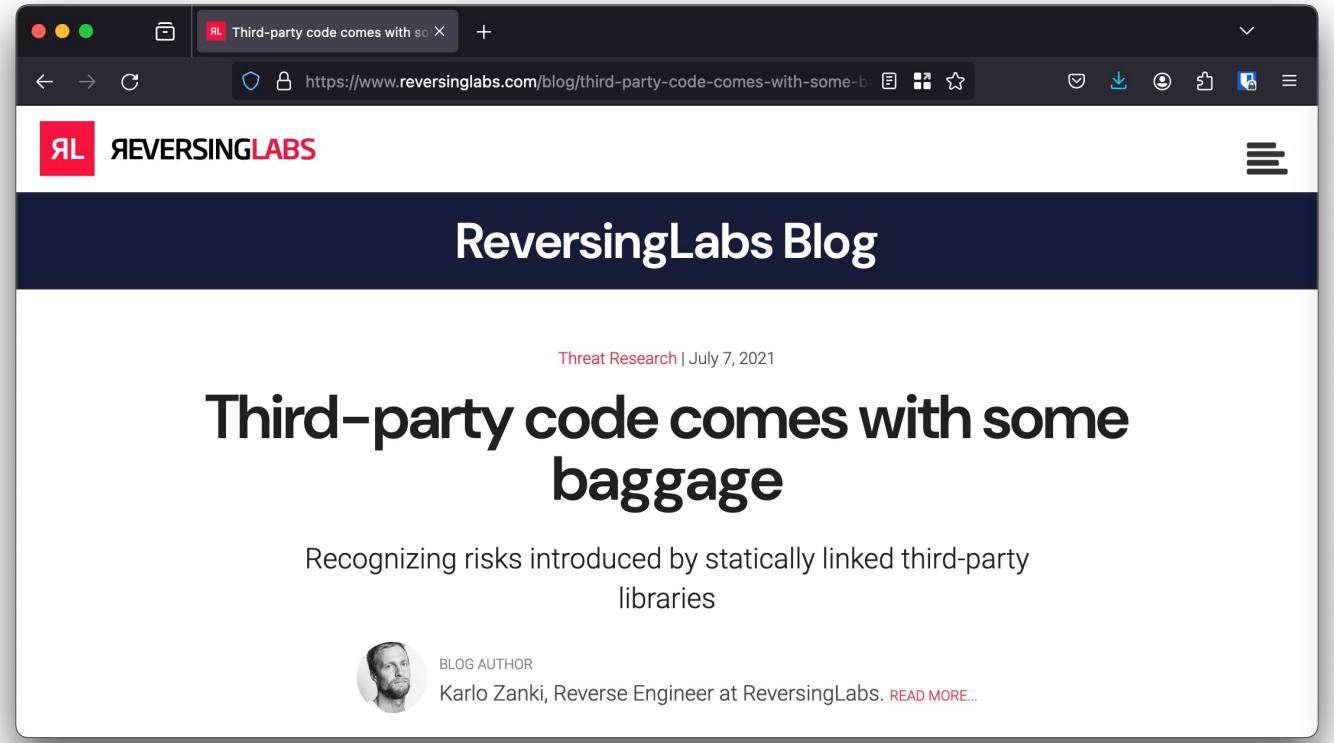
DotNet CLI



elia group

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

Do you know what's inside?



elia group

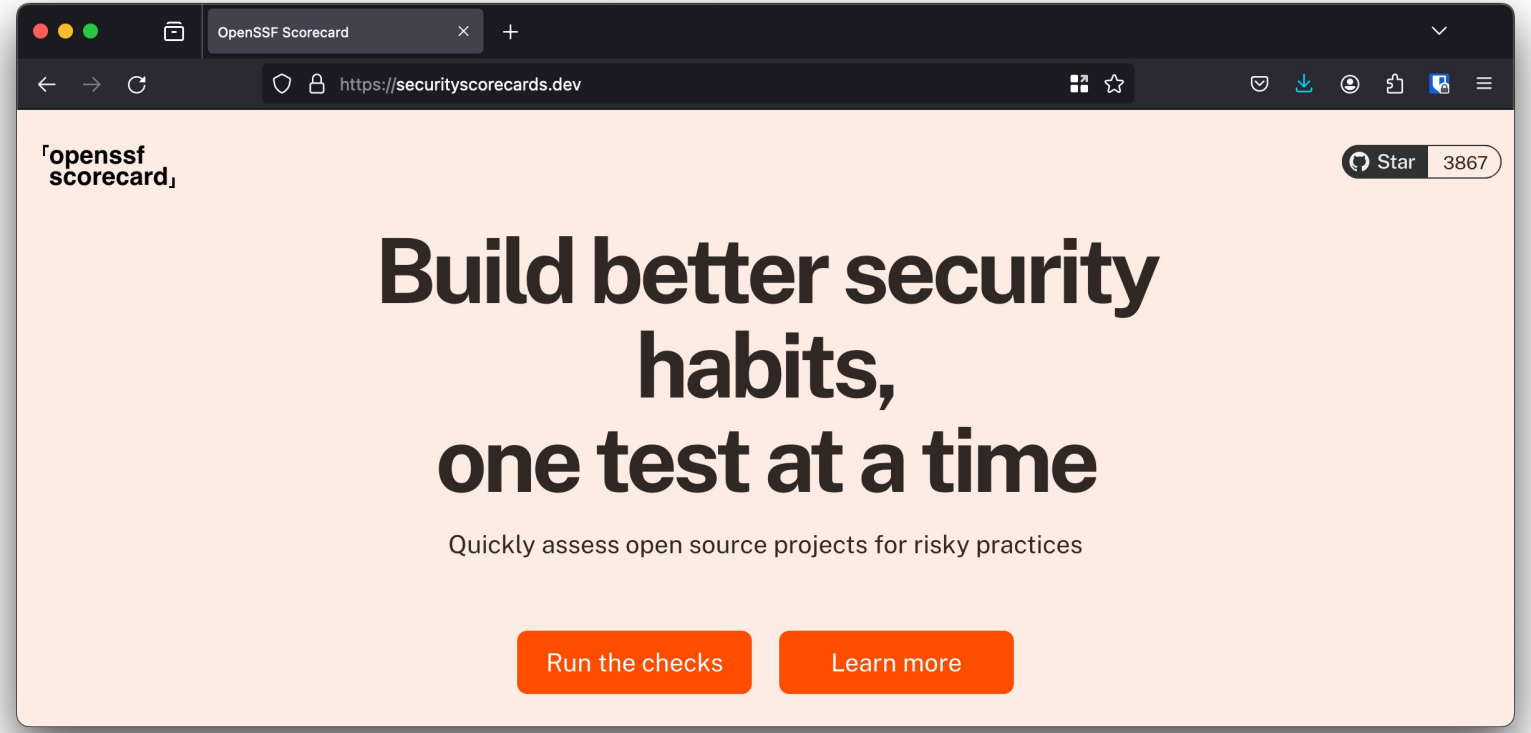
@niels.fennec.dev @nielstanis@infosec.exchange

Nutrion Label for Software?

elia group

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

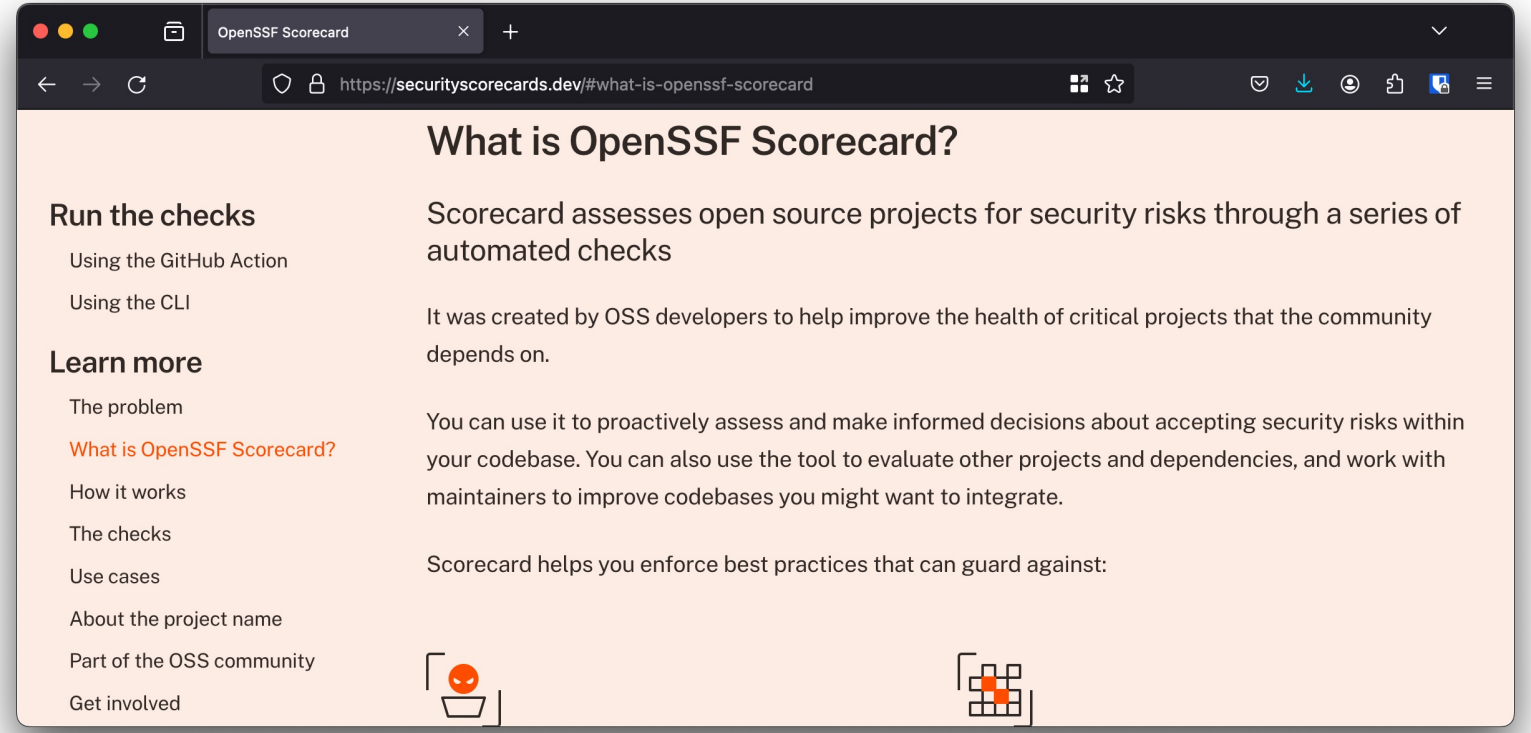
OpenSSF Scorecards



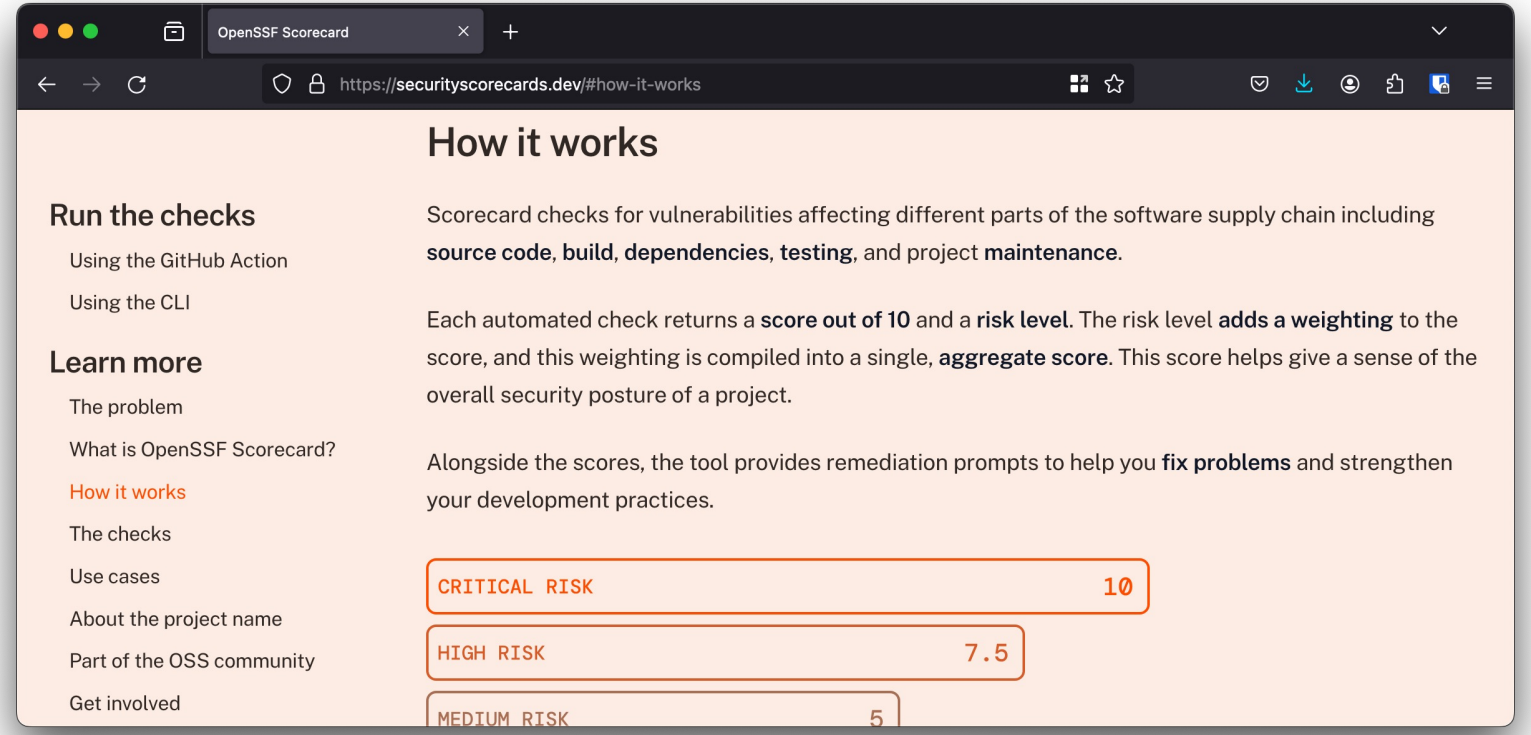
elia group

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

OpenSSF Security Scorecards



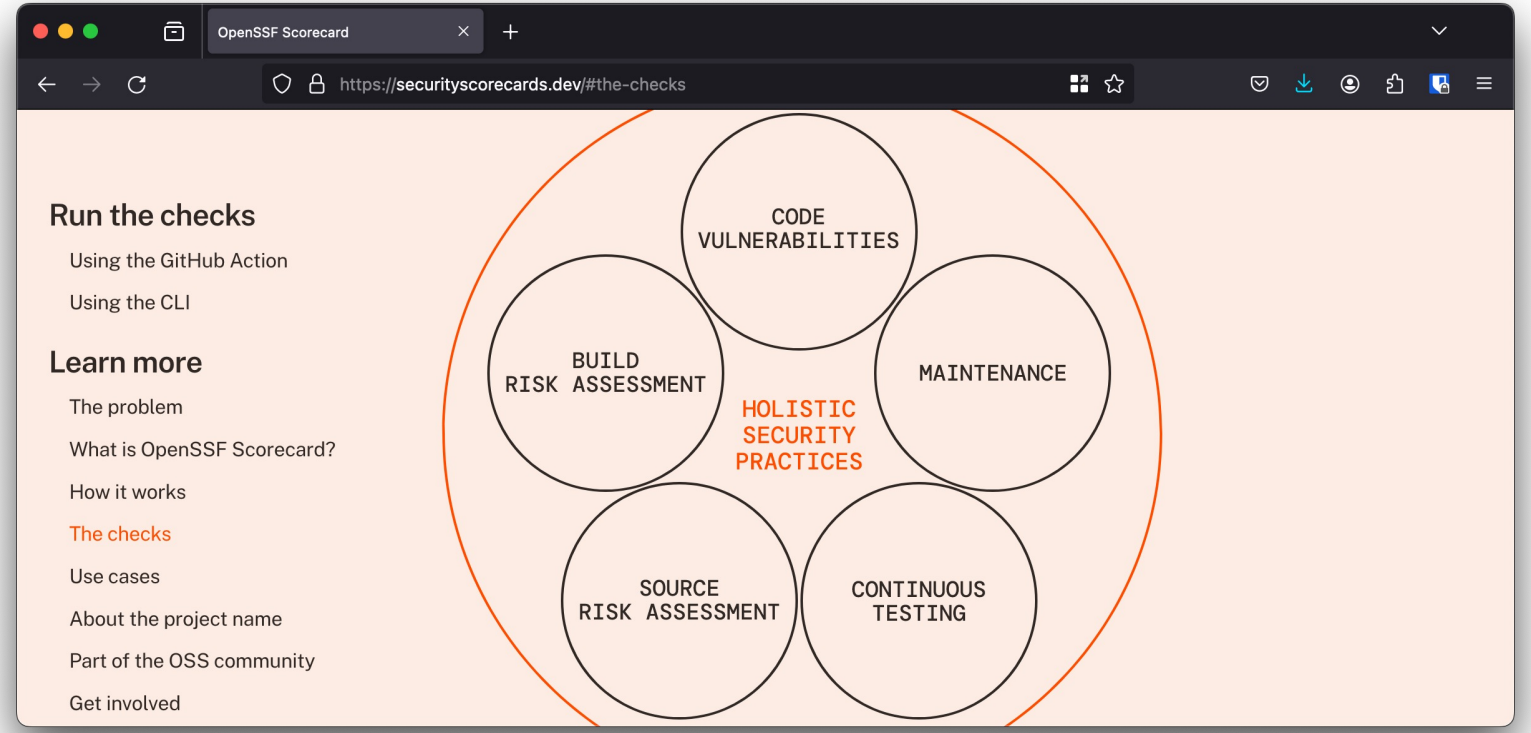
OpenSSF Security Scorecards



elia group

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

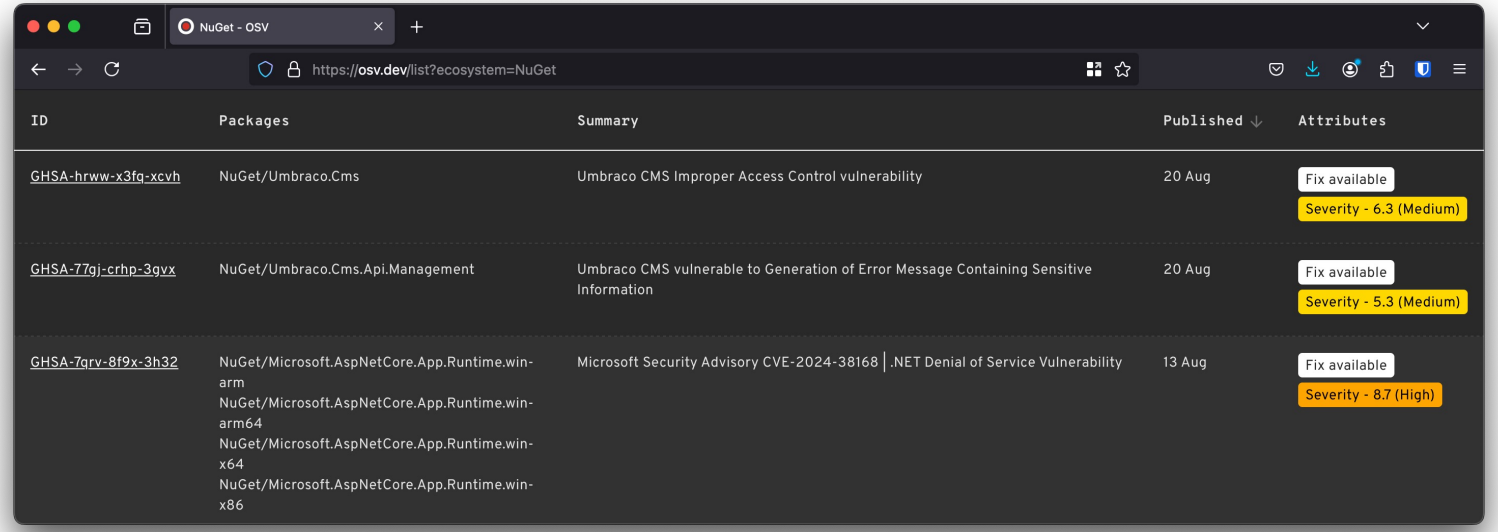
OpenSSF Security Scorecards



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

Code Vulnerabilities (High)

- Does the project have unfixed vulnerabilities? Uses the OSV service.



ID	Packages	Summary	Published	Attributes
GHSA-hrww-x3fq-xcvh	NuGet/Umbraco.Cms	Umbraco CMS Improper Access Control vulnerability	20 Aug	Fix available Severity - 6.3 (Medium)
GHSA-77gj-crhp-3gvx	NuGet/Umbraco.Cms.Api.Management	Umbraco CMS vulnerable to Generation of Error Message Containing Sensitive Information	20 Aug	Fix available Severity - 5.3 (Medium)
GHSA-7qrv-8f9x-3h32	NuGet/Microsoft.AspNetCore.App.Runtime.win-arm NuGet/Microsoft.AspNetCore.App.Runtime.win-arm64 NuGet/Microsoft.AspNetCore.App.Runtime.win-x64 NuGet/Microsoft.AspNetCore.App.Runtime.win-x86	Microsoft Security Advisory CVE-2024-38168 | .NET Denial of Service Vulnerability	13 Aug	Fix available Severity - 8.7 (High)

Maintenance
Dependency-Update-Tool (High)

- Does the project use a dependency update tool? For example Dependabot or Renovate bot?
- Out-of-date dependencies make a project vulnerable to known flaws and prone to attacks.

Maintenance Security Policy (Medium)

- Does project have published security policy?
- E.g. a file named **`SECURITY.md`** (case-insensitive) in a few well-known directories.
- A security policy can give users information about what constitutes a vulnerability and how to report one securely so that information about a bug is not publicly visible.

Maintenance License (Low)

- Does project have license published?
- A license can give users information about how the source code may or may not be used.
- The lack of a license will impede any kind of security review or audit and creates a legal risk for potential users.

Maintenance
CII Best Practices (Low)

- OpenSSF Best Practices Badge Program
- Way for Open Source Software projects to show that they follow best practices.
- Projects can voluntarily self-certify, at no cost, by using this web application to explain how they follow each best practice.

openssf best practices | passing

elia group

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

Continuous testing
CI Tests (Low)

- Does the project run tests before pull requests are merged?
- The check works by looking for a set of CI-system names in GitHub CheckRuns and Statuses among the recent commits (~30).

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

Continuous testing
Fuzzing (Medium)

- This check tries to determine if the project uses fuzzing by checking:
  - Added to OSS-Fuzz project.
  - If ClusterFuzzLite is deployed in the repository;
- Does it make sense to do fuzzing on .NET projects?

Continuous testing
Static Code Analysis (Medium)

- This check tries to determine if the project uses Static Application Security Testing (SAST), also known as static code analysis. It is currently limited to repositories hosted on GitHub.
  - CodeQL
  - SonarCloud
- Definitely room for improvement!

Source Risk Assesement
Binary Artifacts (High)

- This check determines whether the project has generated executable (binary) artifacts in the source repository.
- Binary artifacts cannot be reviewed, allowing possible obsolete or maliciously subverted executables.
- There is need for **reproducible** builds!

Source Risk Assesement
Branch Protection (High)

- This check determines whether a project's default and release branches are protected with GitHub's branch protection or repository rules settings.
  - Requiring code review
  - Prevent force push, in case of public branch all is lost!

Source Risk Assesement
Dangerous Workflow (Critical)

- This check determines whether the project's GitHub Action workflows has dangerous code patterns.
  - Untrusted Code Checkout with certain triggers
  - Script Injection with Untrusted Context Variables
- https://securitylab.github.com/research/github-actions-preventing-pwn-requests/

Source Risk Assesement
Code Review (Low)

- This check determines whether the project requires human code review before pull requests are merged.
- The check determines whether the most recent changes (over the last ~30 commits) have an approval on GitHub and merger!=committer (implicit review)

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

Source Risk Assesement Contributors (Low)

- This check tries to determine if the project has recent contributors from multiple organizations (e.g., companies).
- Relying on single contributor is a risk for sure!
- But is a large list of contributors good?

Build Risk Assesement
Pinned Dependencies (High)

- Does the project pin dependencies used during its build and release process.
- **RestorePackagesWithLockFile** in MSBuild results in **packages.lock.json** file containing versioned dependency tree with hashes
- If Workflow is present what about the Actions used?

Build Risk Assesement
Token Permission (High)

- This check determines whether the project's automated workflows tokens follow the principle of least privilege.
- This is important because attackers may use a compromised token with write access to, for example, push malicious code into the project.

Build Risk Assesement Packaging (Medium)

- This check tries to determine if the project is published as a package.
- Packages give users of a project an easy way to download, install, update, and uninstall the software by a package manager.

Build Risk Assessment
Signed Releases (High)

- This check tries to determine if the project cryptographically signs release artifacts.
  - Signed release packages
  - Signed build provenance

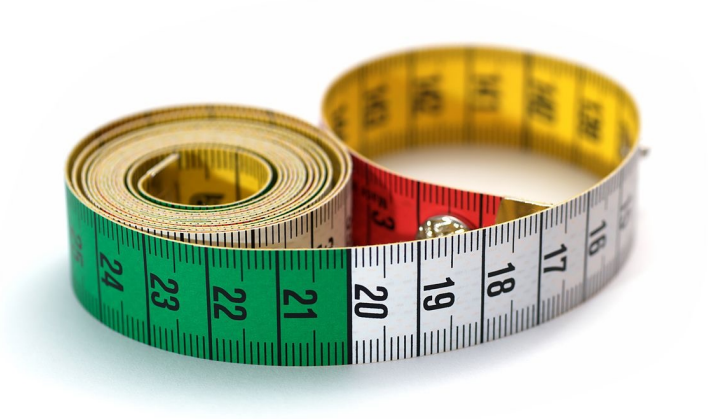
Demo OpenSSF Scorecard
Fennec CLI

```
Running checks
```

elia group

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

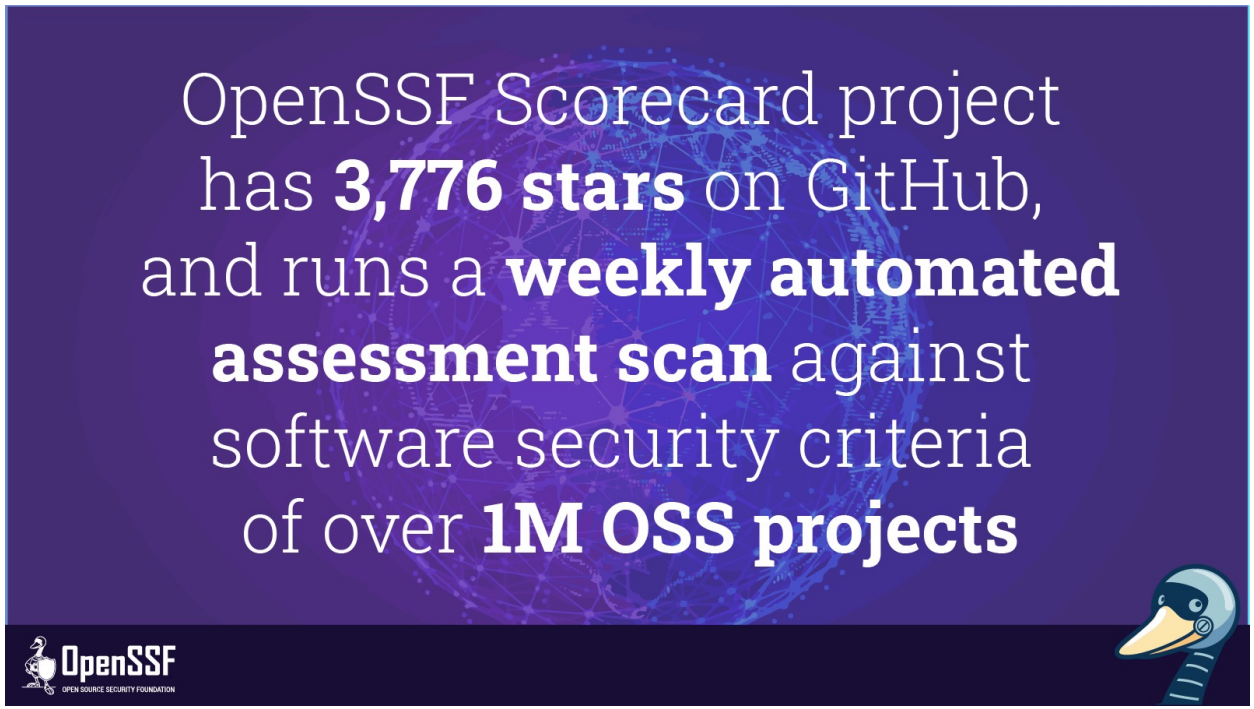
Measure?



elia group

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

OpenSSF Annual Report 2023



elia group

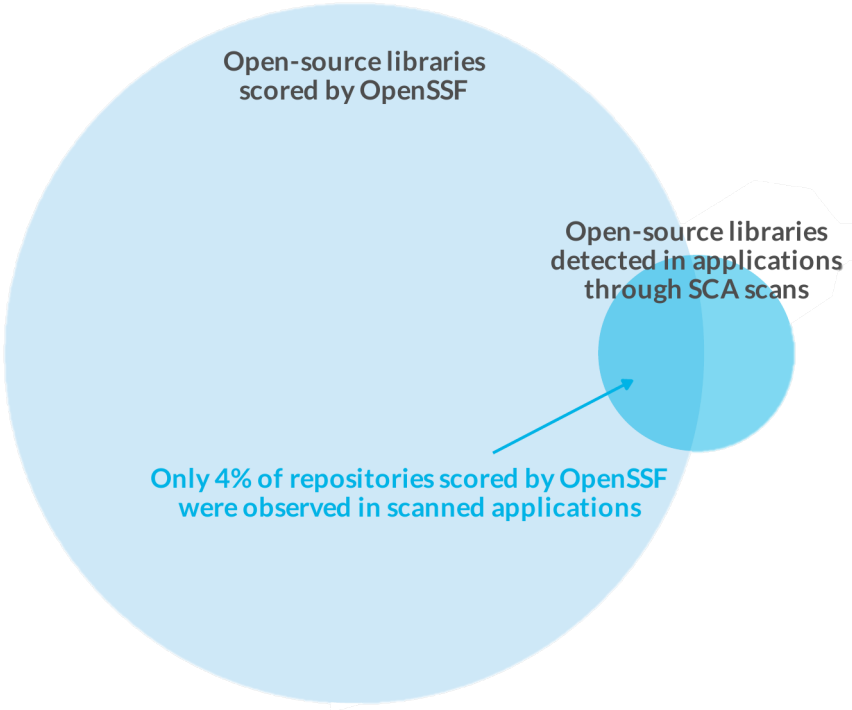
@niels.fennec.dev @nielstanis@infosec.exchange

SOSS & OpenSSF Scorecard

elia group

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

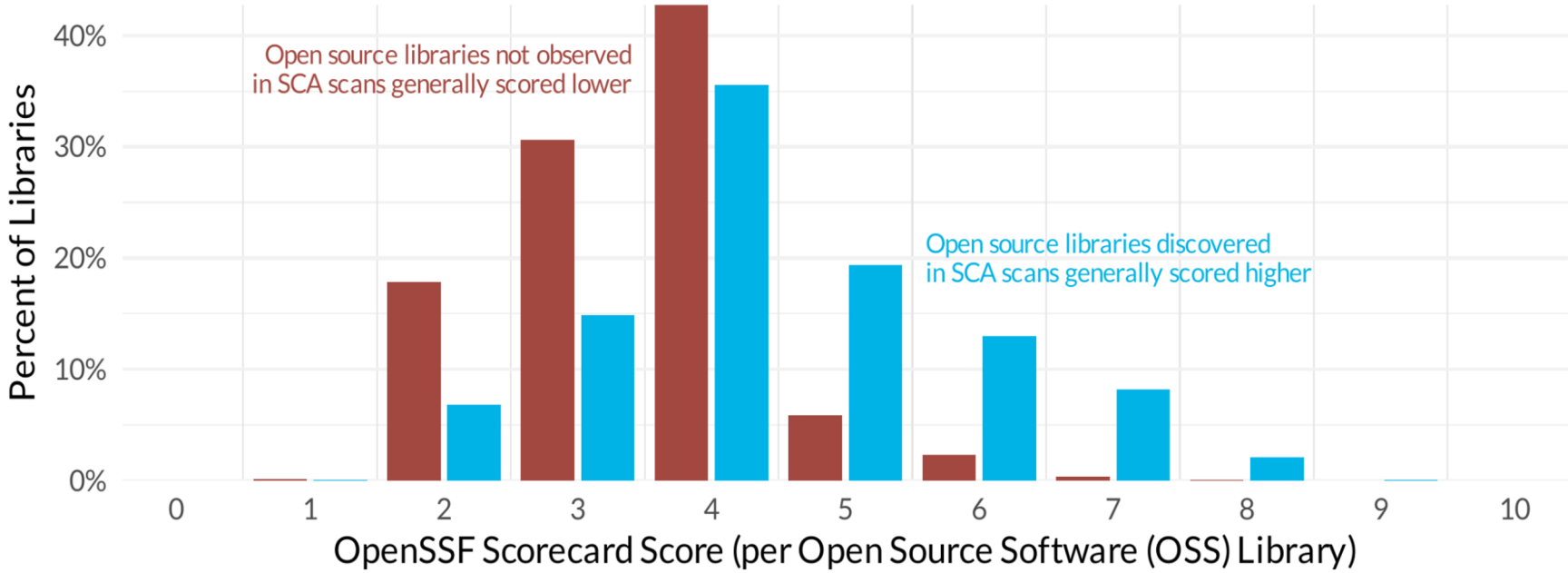
SOSS & OpenSSF Scorecard



elia group

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

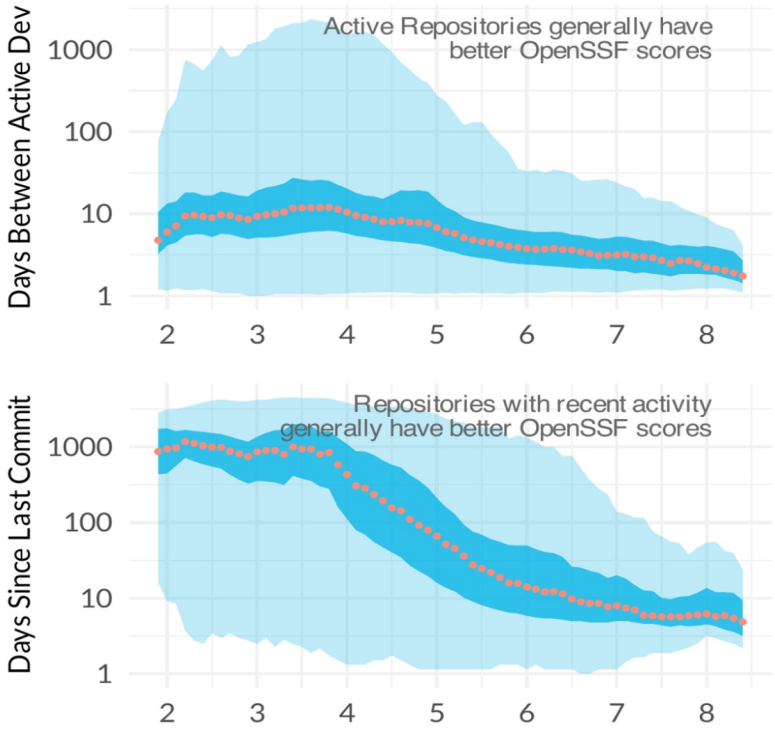
Correlation between SOSS



elia group

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

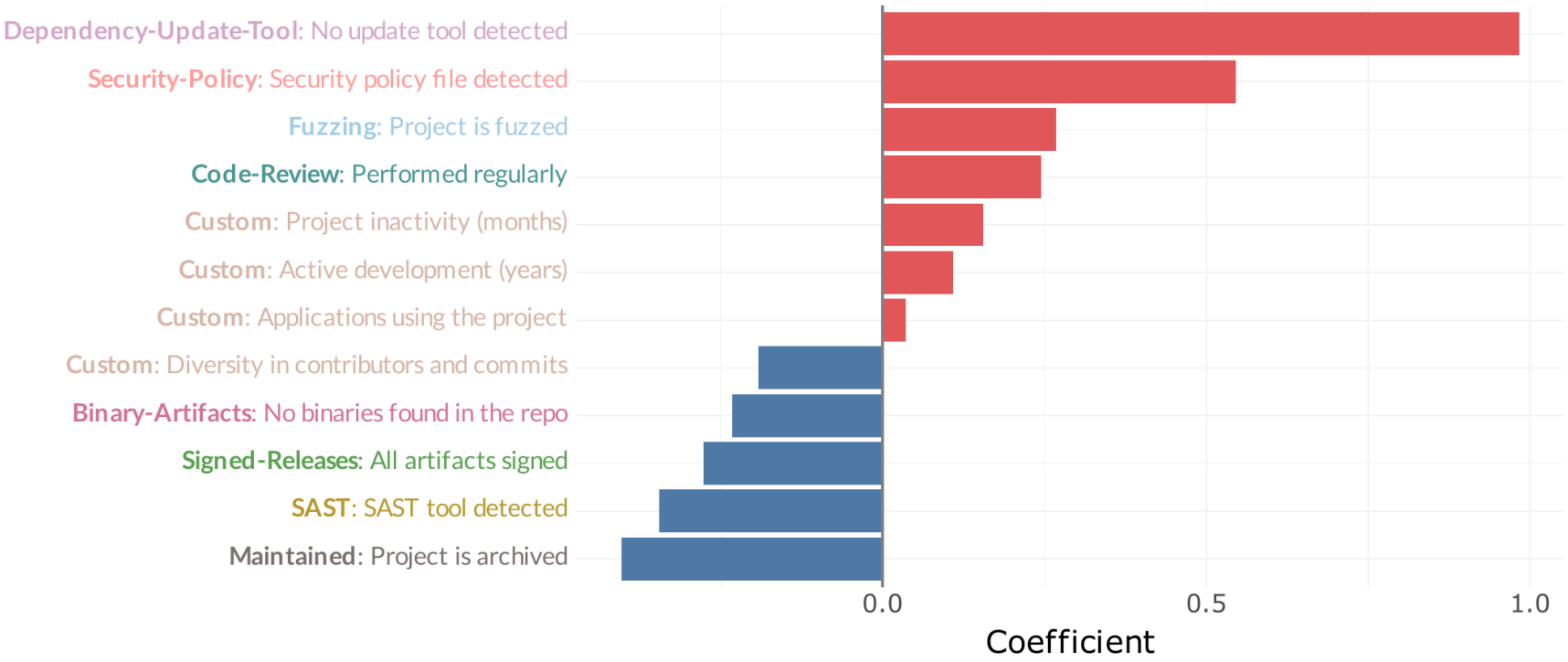
Github commits vs OpenSSF



elia group

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

What really contributes to OSS Security?



elia group

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

What can we improve?

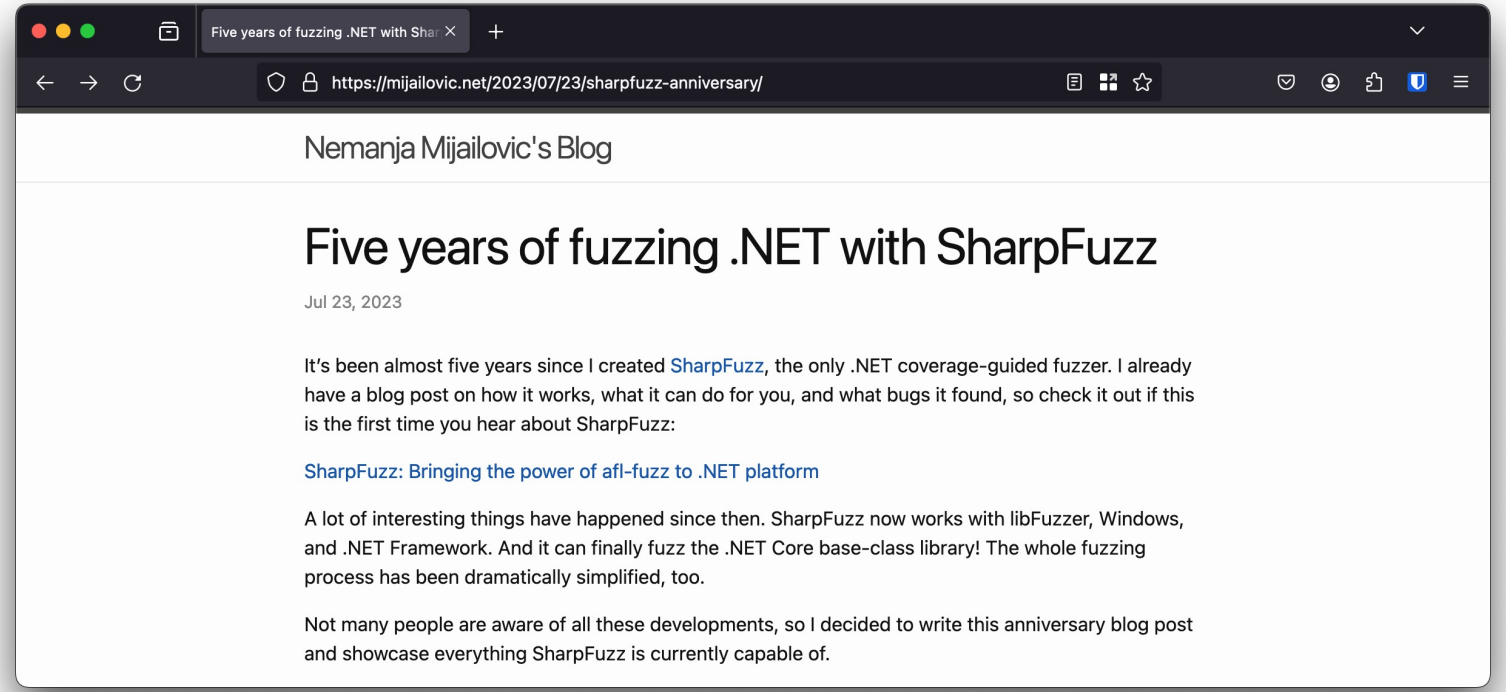
elia group

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

Fuzzing .NET

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

Fuzzing .NET & SharpFuzz



elia group

@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) { }
        });
}
```

elia group

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

Static Code Analysis (SAST)


```
public byte[] CreateHash(string password)
{
  var b = Encoding.UTF8.GetBytes(password);
  return SHA1.HashData(b);
}
```

Static Code Analysis (SAST)


```
public class CustomerController : Controller
{
    public IActionResult GenerateCustomerReport(string customerID)
    {
        var data = Reporting.GenerateCustomerReportOverview(customerID)
        return View(data);
    }
}
```

```
public static class Reporting
{
    public static byte[] GenerateCustomerReportOverview(string ID)
    {
        return System.IO.File.ReadAllBytes($"./data/{ID}.pdf");
    }
}
```

elia group

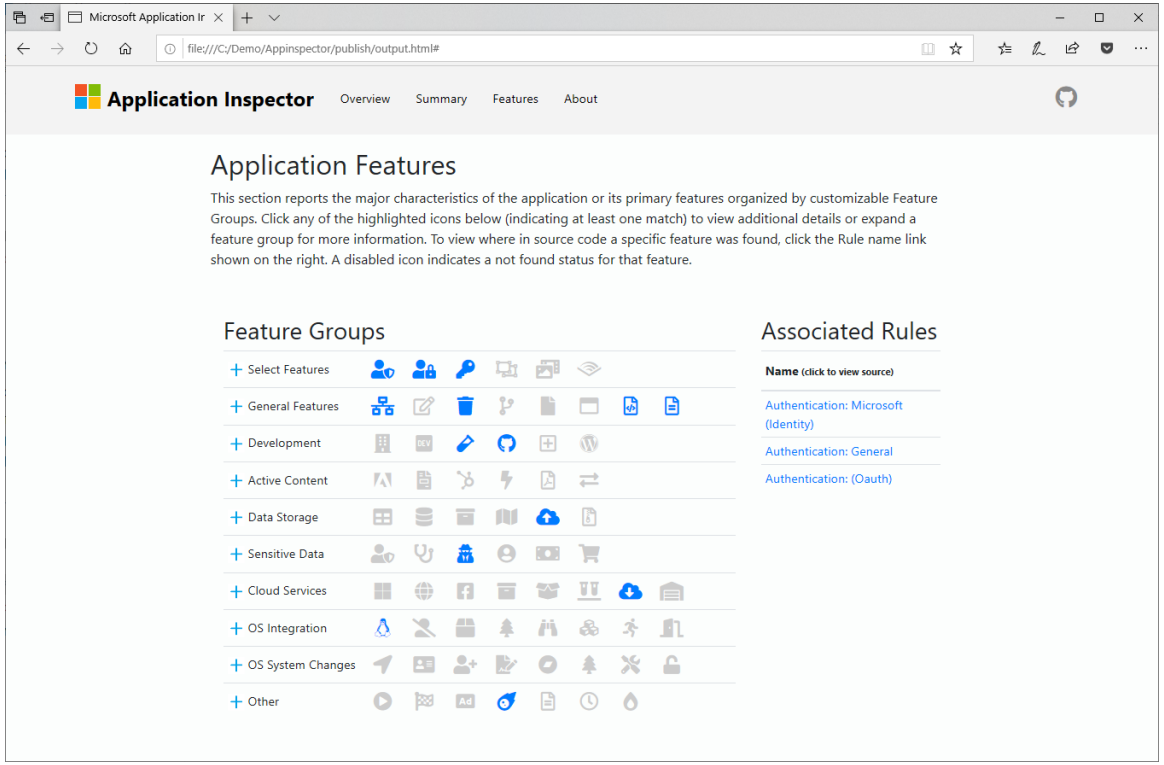
@niels.fennec.dev @nielstanis@infosec.exchange

.NET Reproducibility

- Reproducible builds → independently-verifiable path from source to binary code.
- .NET Roslyn Deterministic Inputs
- How reproducible is a simple console app?
- Fennec Diff (work-in-progress)

Application Inspector

New & Improved!

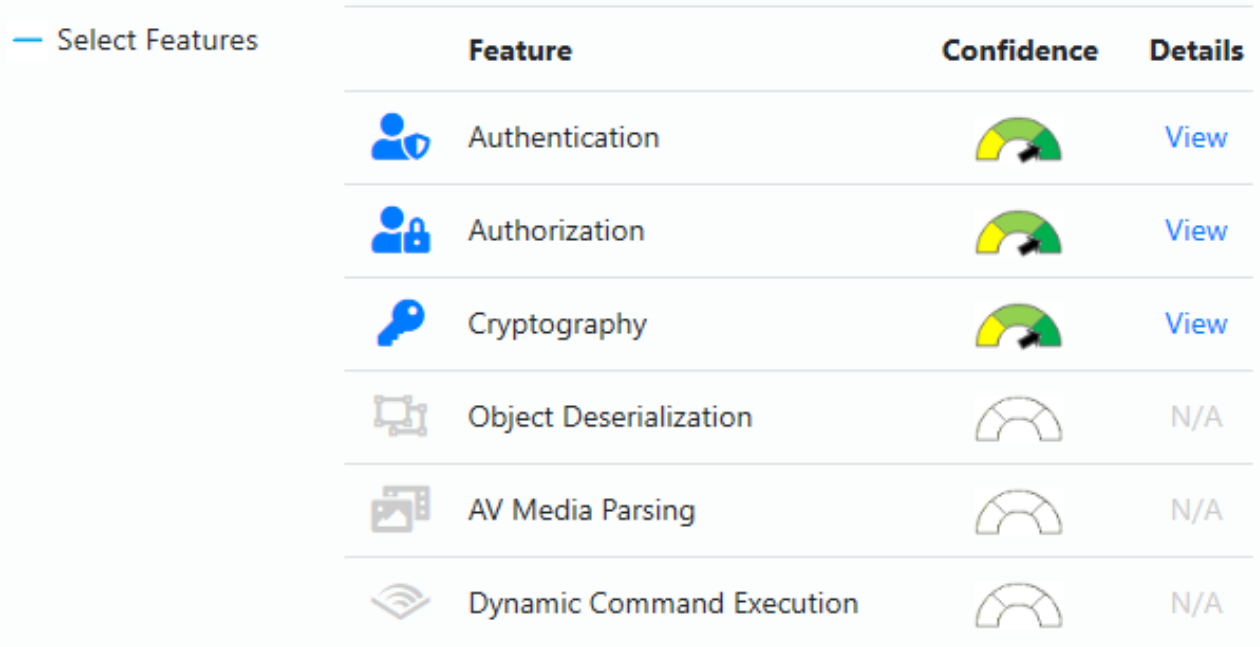


elia group

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

Application Inspector

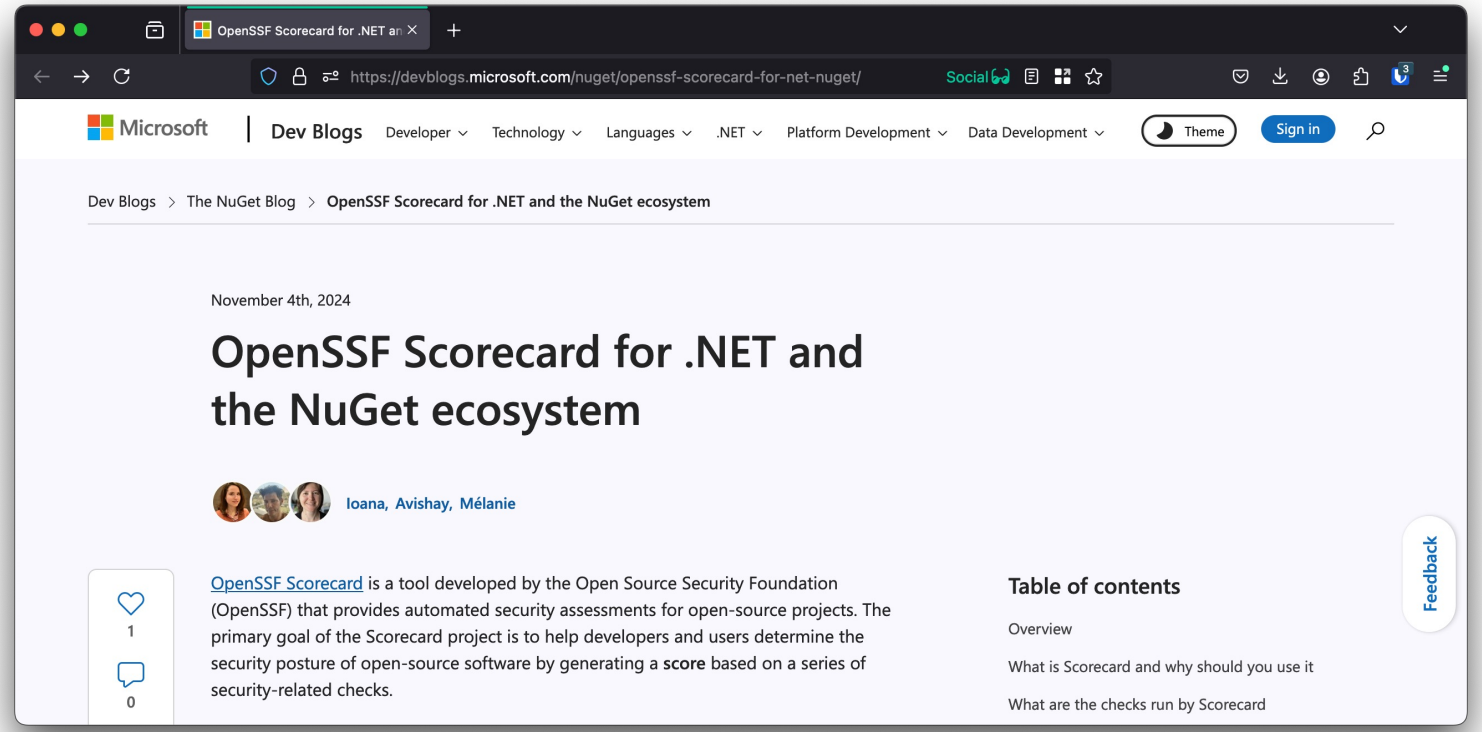
New & Improved!



elia group

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

NuGet Blog



elia group

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

Conclusion

- Scorecard helps security reviewing a NuGet Package
- Better understand what's inside, how it’s build/maintained and what are the risks!
- Scorecard should not be a goal on its own!

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

Conclusion

- Room for .NET specific improvements with Fennec CLI
  - Tools (diff, insights)
  - Transitive Trust Graph
  - Contribute back to OpenSSF Scorecard

```
dotnet tool install -g fennec --prerelease
```

elia group

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

Danke, Merci, Bedankt!

- https://github.com/nielstanis/eliadevcon2024/
- ntanis at Veracode.com
- @nielstanis@infosec.exchange
- https://www.fennec.dev
  https://blog.fennec.dev