
VERACODE
Reviewing NuGet Packages security easily using OpenSSF Scorecard
Niels Tanis
Sr. Principal Security Researcher
elia group

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

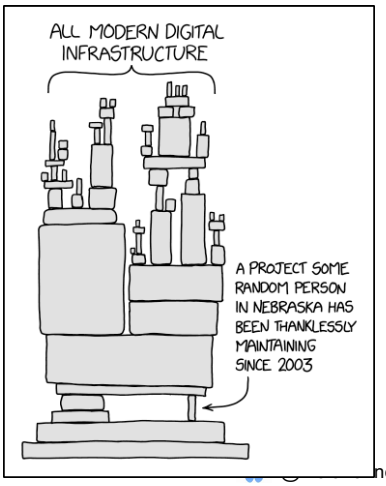
VERACODE

MVP Microsoft® Most Valuable Professional

elia group

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

Modern Application Architecture
XKCD 2347



elia group

nec.dev @nielstanis@infosec.exchange

https://xkcd.com/2347/

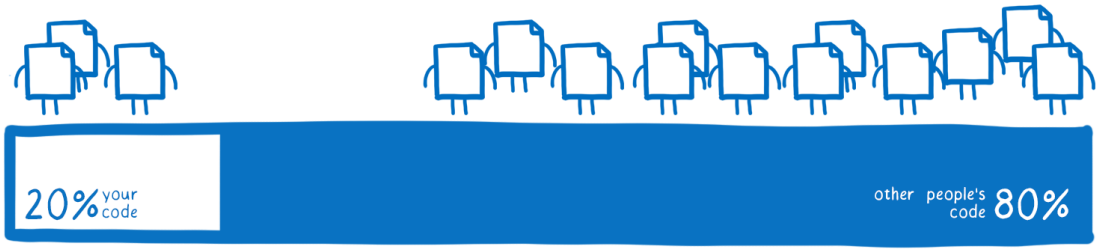
Agenda

- 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



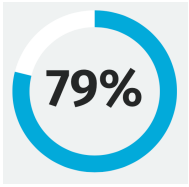
elia group

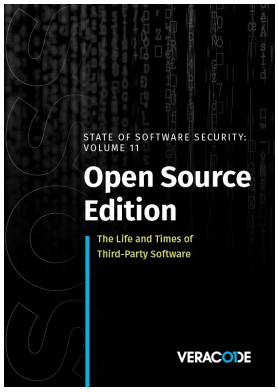
@niels.fennec.dev @nielstanis@infosec.exchange

https://hacks.mozilla.org/2019/11/announcing-the-bytecode-alliance/

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

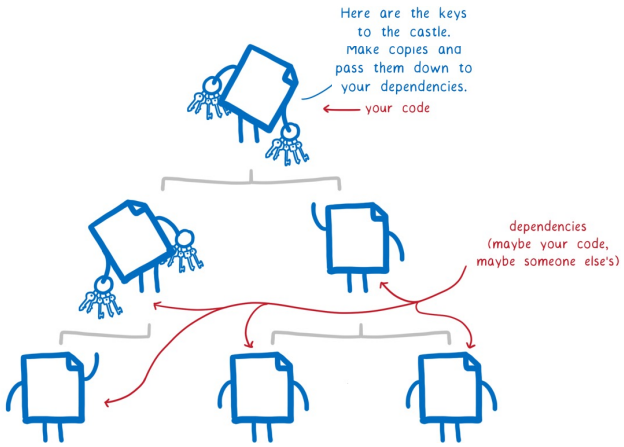
- Analysed our data August-November 2023
  - Total set of almost 39K unique applications scanned
- 2.8% run version vulnerable to Log4Shell
- 3.8% run version patched but vulnerable to other CVE
- 32% rely on a version that's end-of-life and have no support for any patches.

elia group

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

https://www.veracode.com/blog/research/state-log4j-vulnerabilities-how-much-did-log4shell-change

Average codebase composition

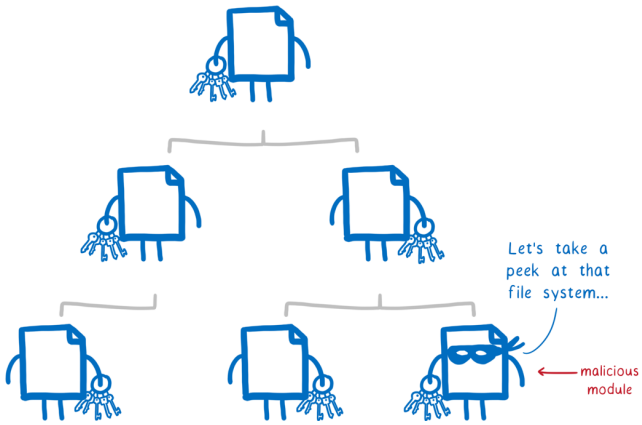


elia group

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

https://hacks.mozilla.org/2019/11/announcing-the-bytecode-alliance/

Malicious Assembly

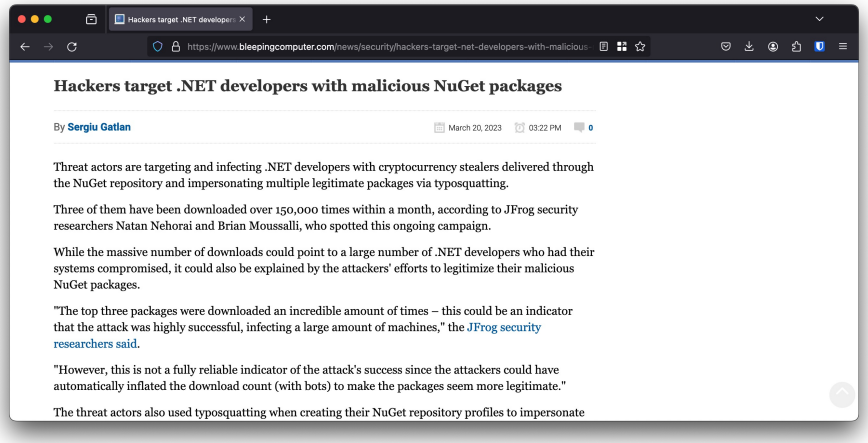


elia group

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

https://hacks.mozilla.org/2019/11/announcing-the-bytecode-alliance/

Malicious Package



Hackers target .NET developers with malicious NuGet packages

By Sergiu Gatlan — March 20, 2023 03:22 PM 0

Threat actors are targeting and infecting .NET developers with cryptocurrency stealers delivered through the NuGet repository and impersonating multiple legitimate packages via typosquatting.

Three of them have been downloaded over 150,000 times within a month, according to JFrog security researchers Natan Nehorai and Brian Moussalli, who spotted this ongoing campaign.

While the massive number of downloads could point to a large number of .NET developers who had their systems compromised, it could also be explained by the attackers' efforts to legitimize their malicious NuGet packages.

"The top three packages were downloaded an incredible amount of times – this could be an indicator that the attack was highly successful, infecting a large amount of machines," the JFrog security researchers said.

"However, this is not a fully reliable indicator of the attack's success since the attackers could have automatically inflated the download count (with bots) to make the packages seem more legitimate."

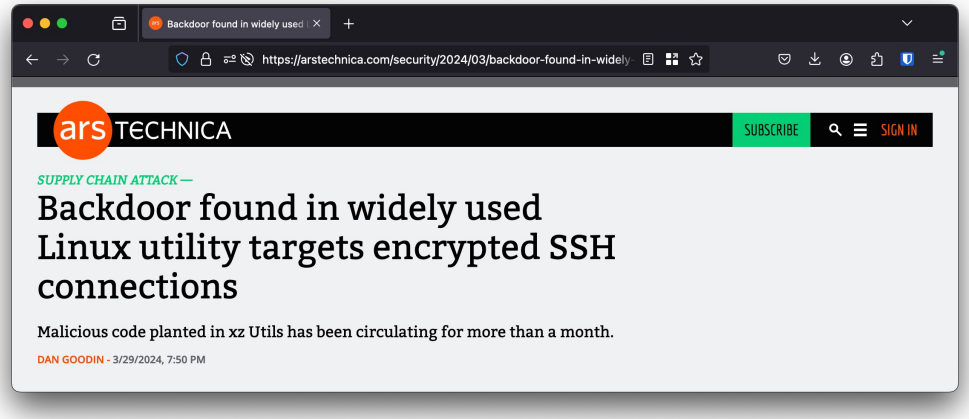
The threat actors also used typosquatting when creating their NuGet repository profiles to impersonate

elia group

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

https://www.bleepingcomputer.com/news/security/hackers-target-net-developers-with-malicious-nuget-packages//

XZ Backdoor

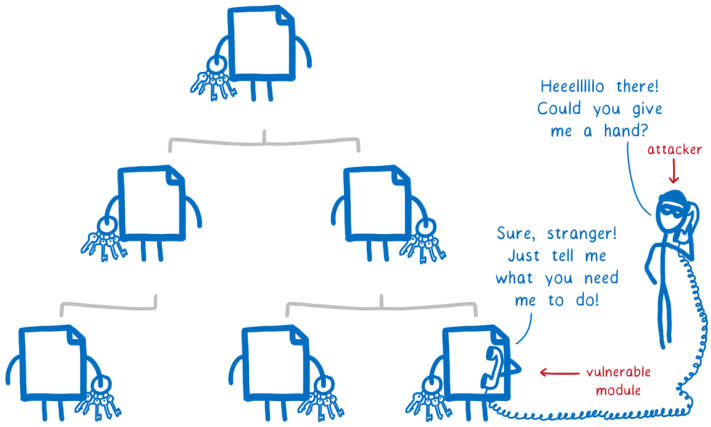


elia group

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

https://arstechnica.com/security/2024/03/backdoor-found-in-widely-used-linux-utility-breaks-encrypted-ssh-connections/

Vulnerable Assembly

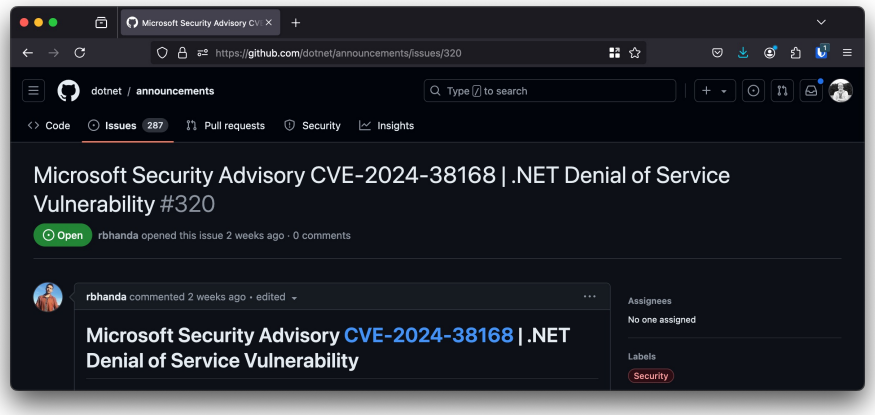


elia group

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

https://hacks.mozilla.org/2019/11/announcing-the-bytecode-alliance/

Vulnerabilities in Libraries

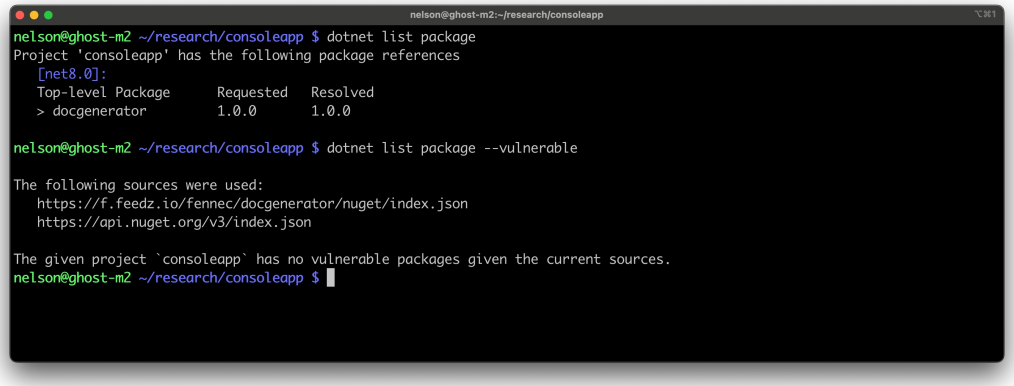


elia group

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

https://github.com/dotnet/announcements/issues/320

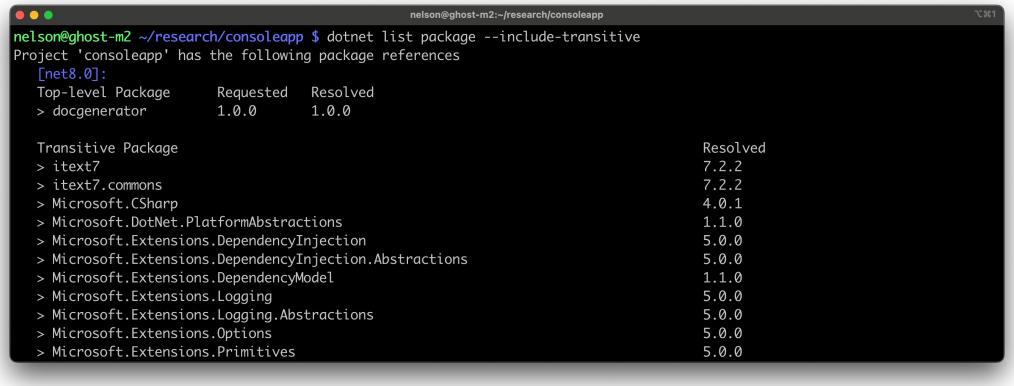
DotNet CLI



elia group

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

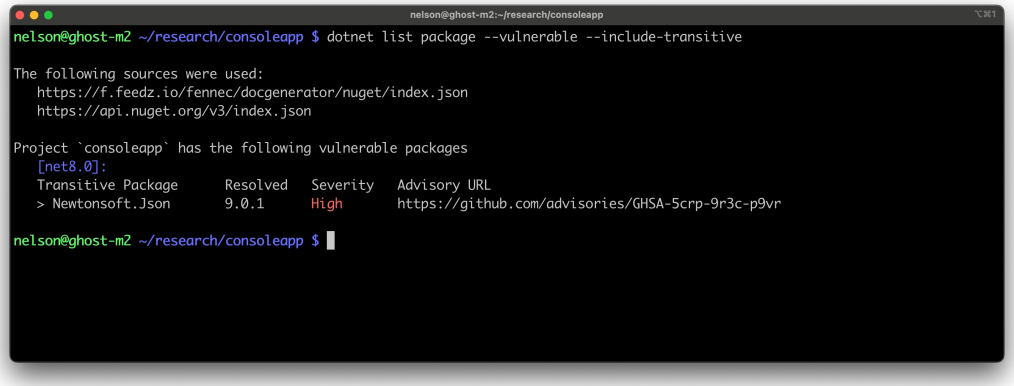
DotNet CLI



elia group

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

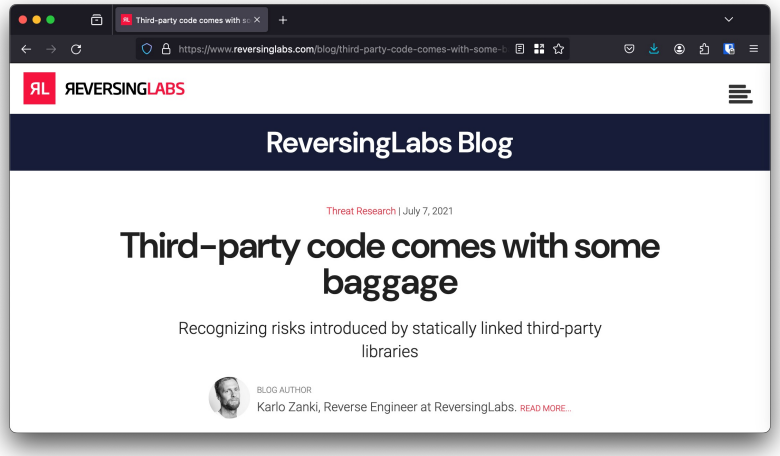
DotNet CLI



elia group

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

Do you know what's inside?



elia group

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

https://www.reversinglabs.com/blog/third-party-code-comes-with-some-baggage

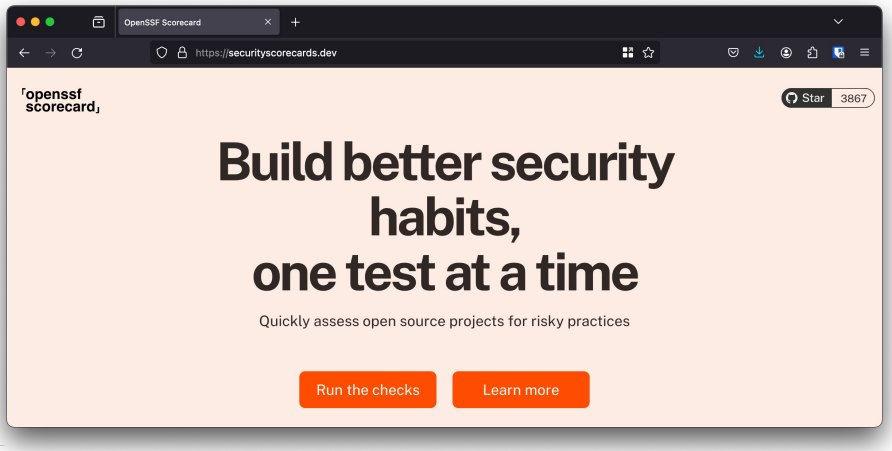
Nutrion Label for Software?

elia group

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

https://securityscorecards.dev/

OpenSSF Scorecards

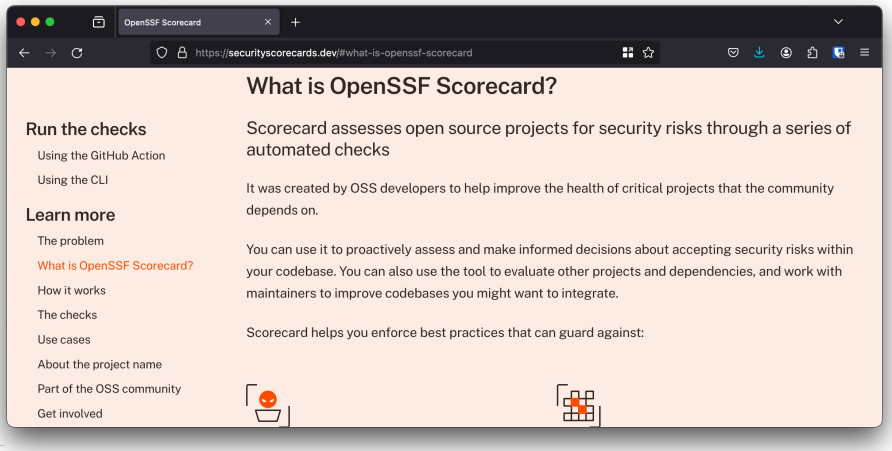


elia group

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

https://securityscorecards.dev/

OpenSSF Security Scorecards

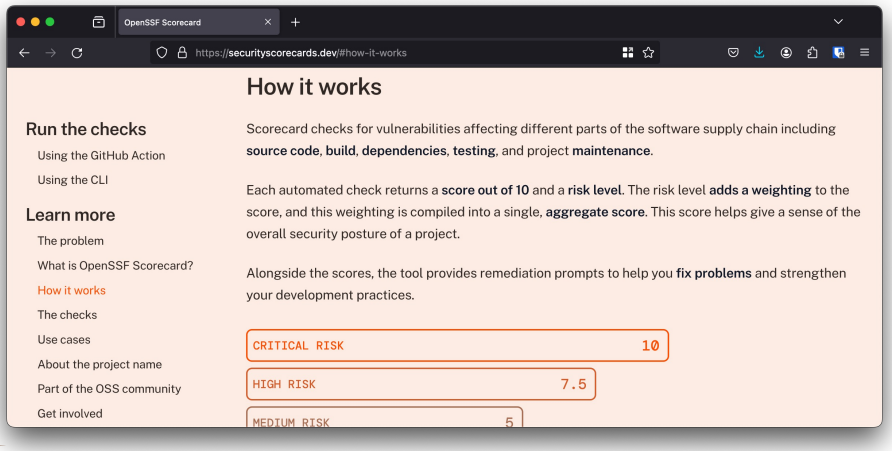


elia group

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

https://securityscorecards.dev/

OpenSSF Security Scorecards

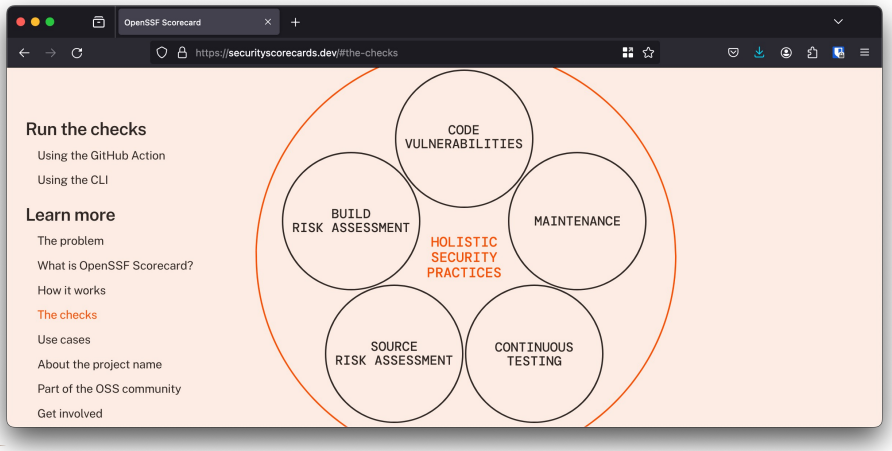


elia group

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

https://securityscorecards.dev/

OpenSSF Security Scorecards



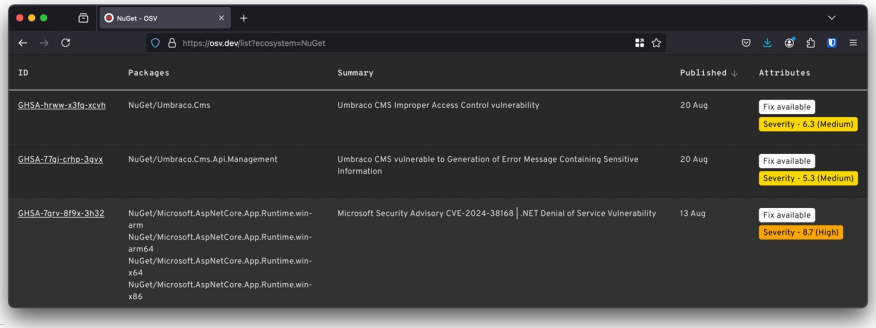
elia group

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

https://securityscorecards.dev/

Code Vulnerabilities (High)

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



elia group

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

https://osv.dev/list?ecosystem=NuGet

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

https://www.bestpractices.dev/en/criteria/0

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).

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)

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.

elia group

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

https://securitylab.github.com/research/github-actions-preventing-pwn-requests/

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

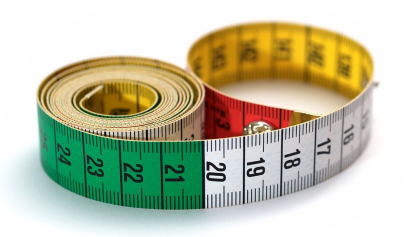
Demo OpenSSF Scorecard
Fennec CLI

```
Running checks
```

elia group

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

Measure?

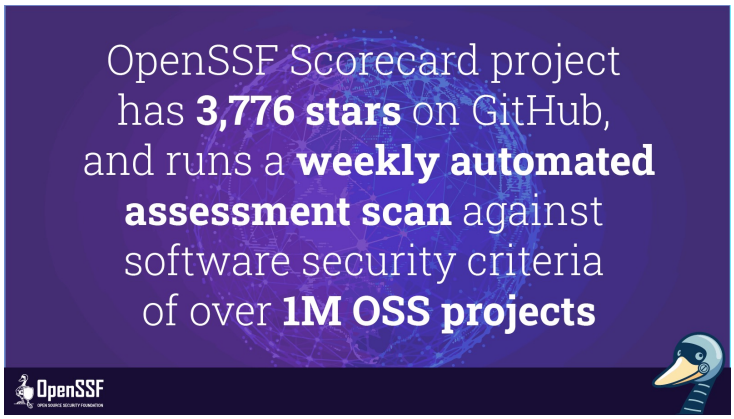


elia group

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

https://www.bestpractices.dev/en/criteria/0

OpenSSF Annual Report 2023



elia group

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

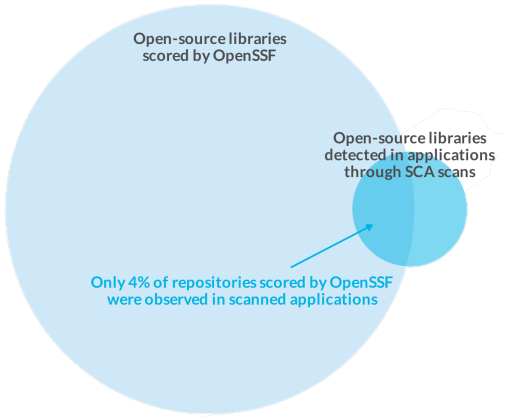
https://openssf.org/download-the-2023-openssf-annual-report/

SOSS & OpenSSF Scorecard

elia group

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

SOSS & OpenSSF Scorecard

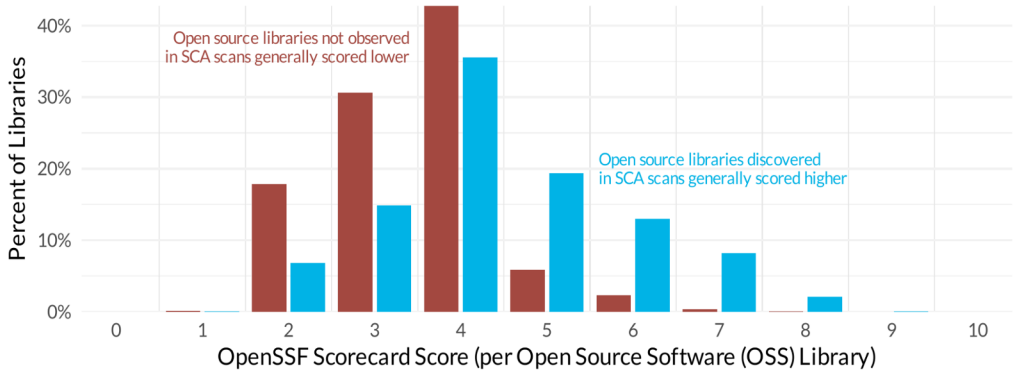


elia group

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

https://www.rsaconference.com/Library/presentation/usa/2024/quantifying%20the%20probability%20of%20flaws%20in%20open%20source

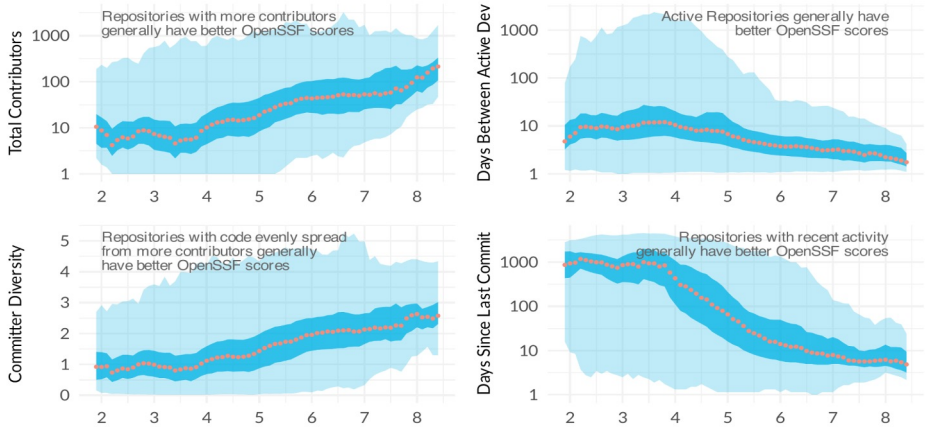
Correlation between SOSS



elia group

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

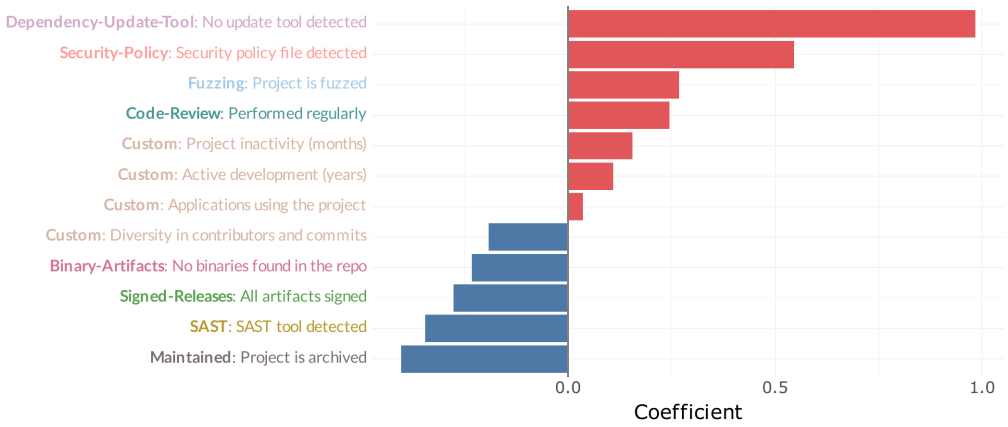
Github commits vs OpenSSF



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

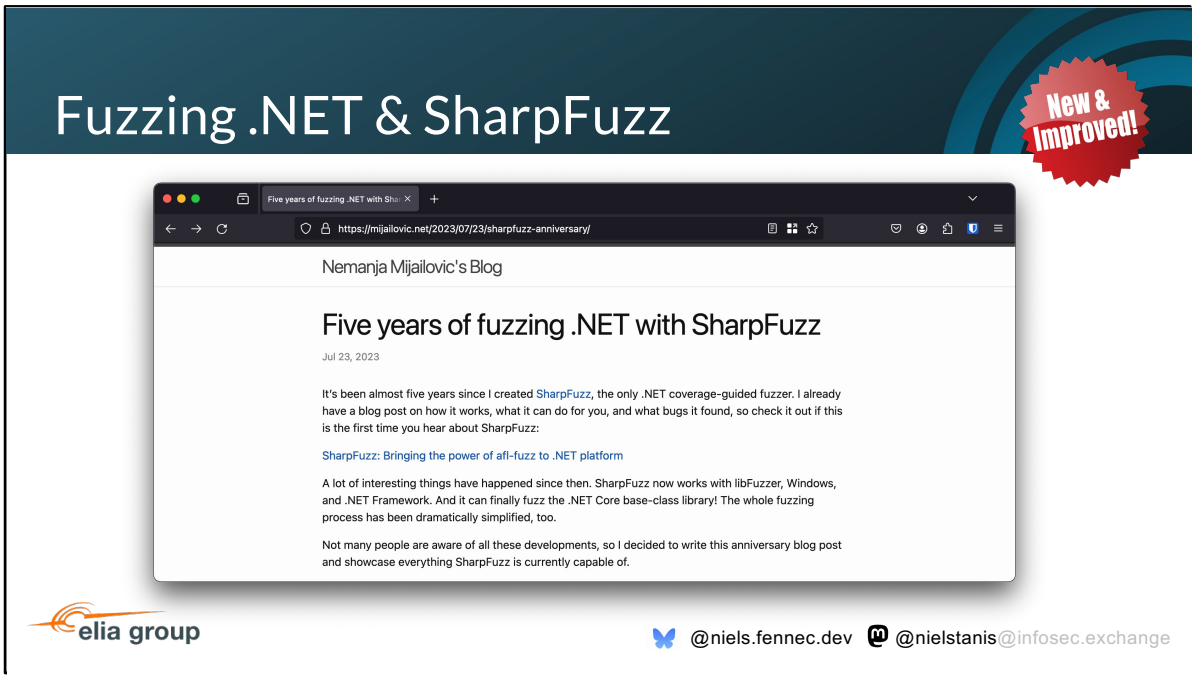
https://www.bestpractices.dev/en/criteria/0

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?

https://www.bestpractices.dev/en/criteria/0

Fuzzing .NET & SharpFuzz



elia group

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

https://mijailovic.net/2023/07/23/sharpfuzz-anniversary/

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

https://github.com/google/fuzzing/blob/master/docs/structure-aware-fuzzing.md

Static Code Analysis (SAST)


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

elia group

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

https://www.bestpractices.dev/en/criteria/0

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

https://www.bestpractices.dev/en/criteria/0

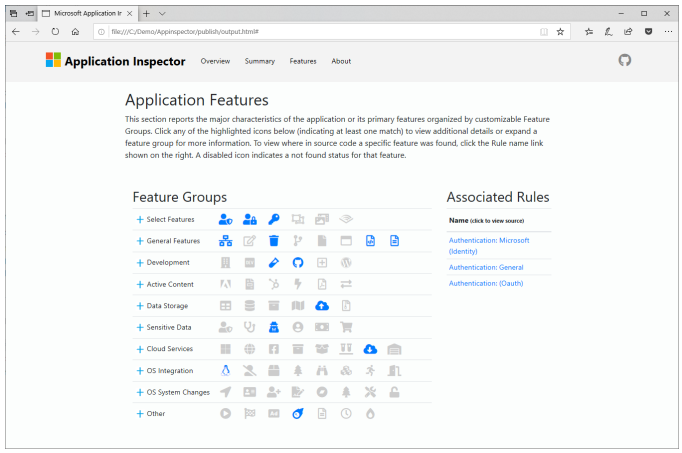
.NET Reproducibility

- 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)

elia group

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

Application Inspector

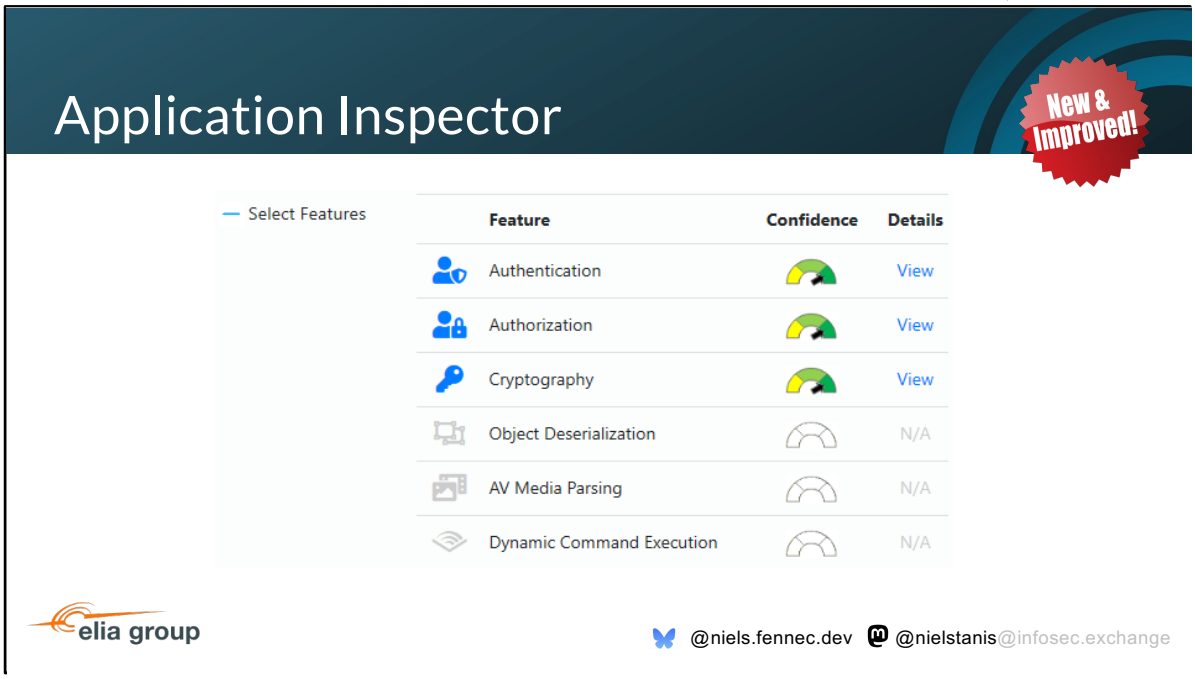


elia group

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

https://github.com/microsoft/ApplicationInspector

Application Inspector

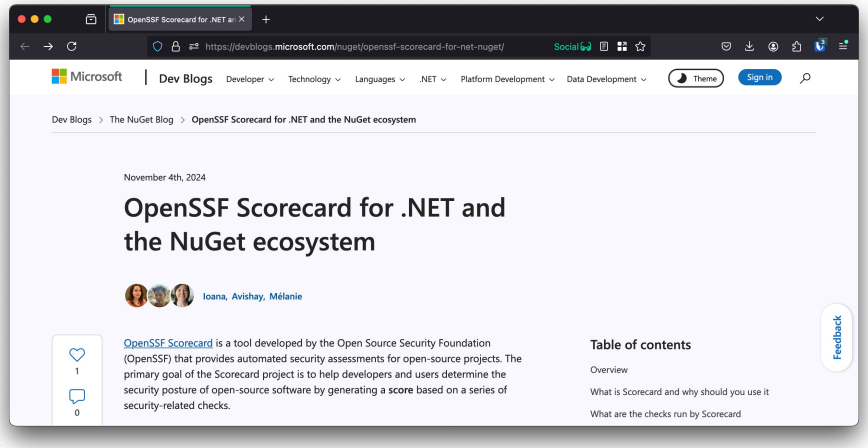


elia group

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

https://github.com/microsoft/ApplicationInspector

NuGet Blog



elia group

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

https://devblogs.microsoft.com/nuget/openssf-scorecard-for-net-nuget/

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!

elia group

@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

- d

Danke, Merci, Bedankt!

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

elia group

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