How your .NET software supply chain is open to attack

and how to fix it

Andrei EPURE

Engineering Manager at 

Software supply chain

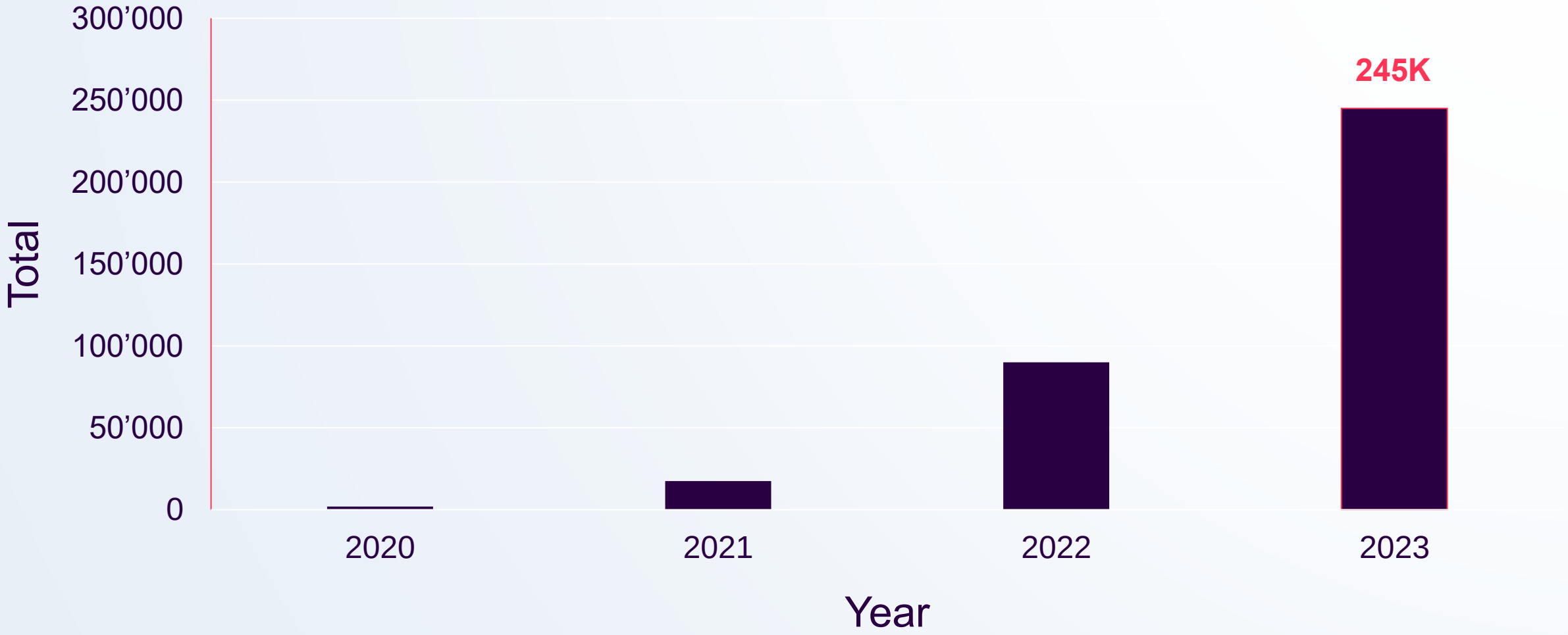
Source code

Libraries

Package managers (NuGet, NPM, PyPI, Maven)

Build tools, CI/CD, etc

Number of malicious packages discovered in major package managers



Source: Sonatype’s “9th State of the Software Supply Chain report” (2023)

What are the risks?

Remote code execution to steal via exfiltration:

- Source code
- Production secrets
- Credentials

More about attack tactics & techniques:
- Initial Access MITRE TA0001
- Code execution: MITRE TA0002
- Data exfiltration: MITRE TA0010

Andrei Epure

Engineering Manager

feedback form & slides

AndreiEpure.ro

sonar

home of {clean code}

Booth A20, level 0, near stage 8

Free - IDE Plugin

sonarlint

Free - On Premise

sonarqube

Free 4 OSS - SaaS

sonarcloud

30+ languages

DevOps integration

clean code throughout the development workflow

www.sonarsource.com

sonar

Agenda

Supply chain attacks:

1. Typosquatting
2. Dependency confusion

Attack & Defense in NuGet

Typosquatting

G	O	O	G	L	E
G	O	G	L	E	

dotnet add package **Gogle**.Protobuf -v 5.3.2

NUGET.ORG

Google.Protobuf
5.3.2

Gogle.Protobuf
5.3.2

Typosquatting (2016) - Nikolai Tschacher

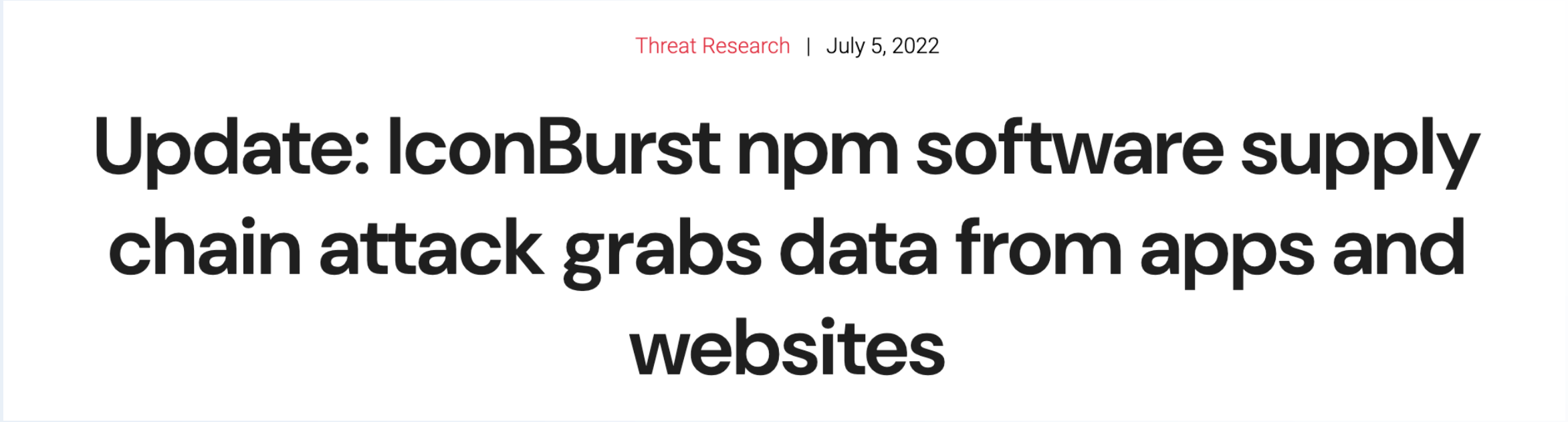
NPM and PyPi

17K infected PCs

50% admin privilege

Source: incolumitas.com

Typosquatting (2022)



27K downloads

Source: ReversingLabs

Typosquatting (2023)

Analyzing Impala Stealer – Payload of the first NuGet attack campaign

Part two of series "First NuGet malicious packages campaign"

By Ori Hollander, JFrog Security Research | April 10, 2023
9 min read

SHARE:

Source: JFrog

BLOG HOME >

Attackers are starting to target .NET developers with packages

By Natan Nehorai, Application Security Res
Research Team Leader | March 20, 2023
12 min read

Source: JFrog

Threat Research | July 11, 2024

Malicious NuGet campaign uses homoglyphs and IL weaving to fool devs

Malware authors upped their game, using homoglyphs to impersonate a protected NuGet prefix and IL weaving to inject malicious code, RL researchers found.

BLOG AUTHOR
Karlo Zanki, Reverse Engineer at ReversingLabs. READ MORE...

Source: ReversingLabs

Oct 12, 20

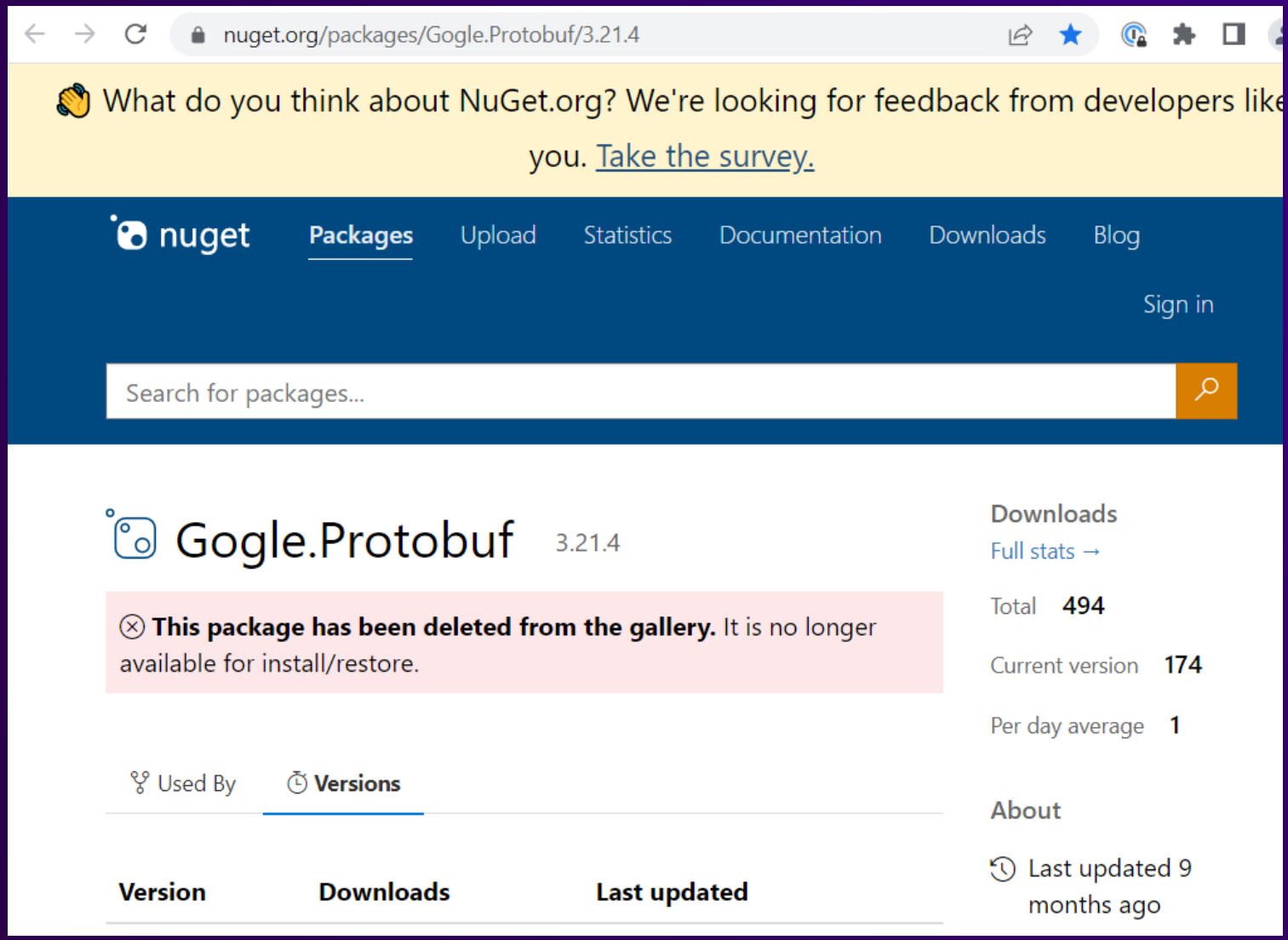
Phylum Disco[illegible] Typosquatted NuGet Package

Source: Phylum

Typosquatting DEMO

Mr. Evil Hacker

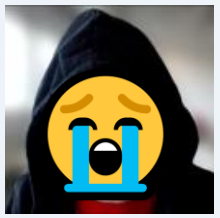
In production

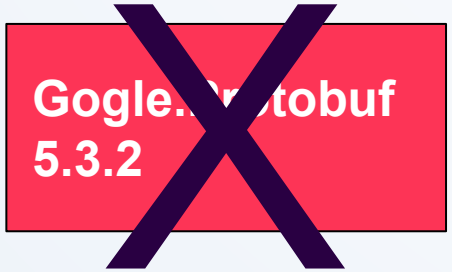


☝ Typosquatting defense

Enable `signatureValidationMode`

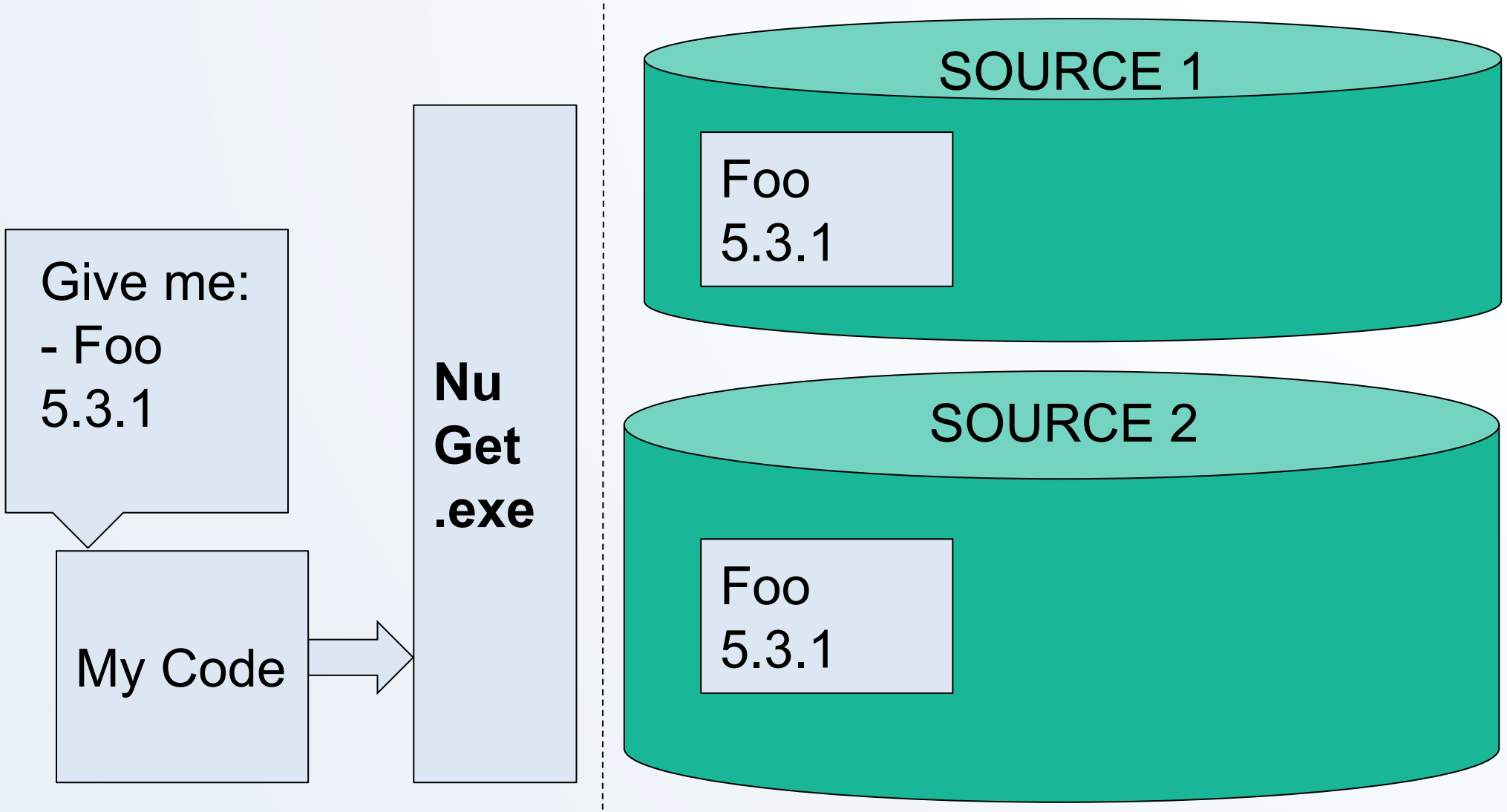
Declare `owners` you trust under `trustedSigners`



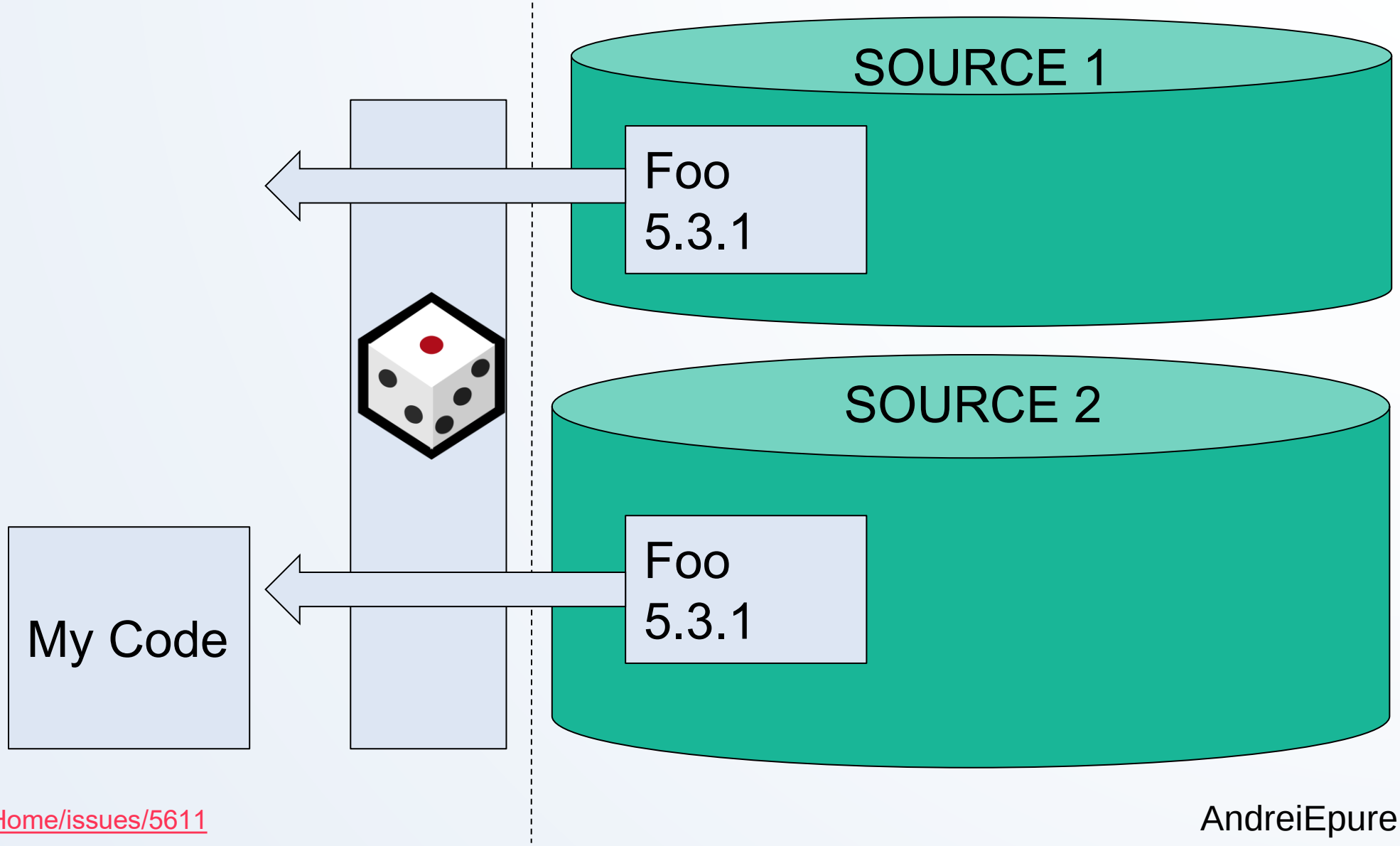


Dependency confusion

NuGet Package Resolution



NuGet Package Resolution

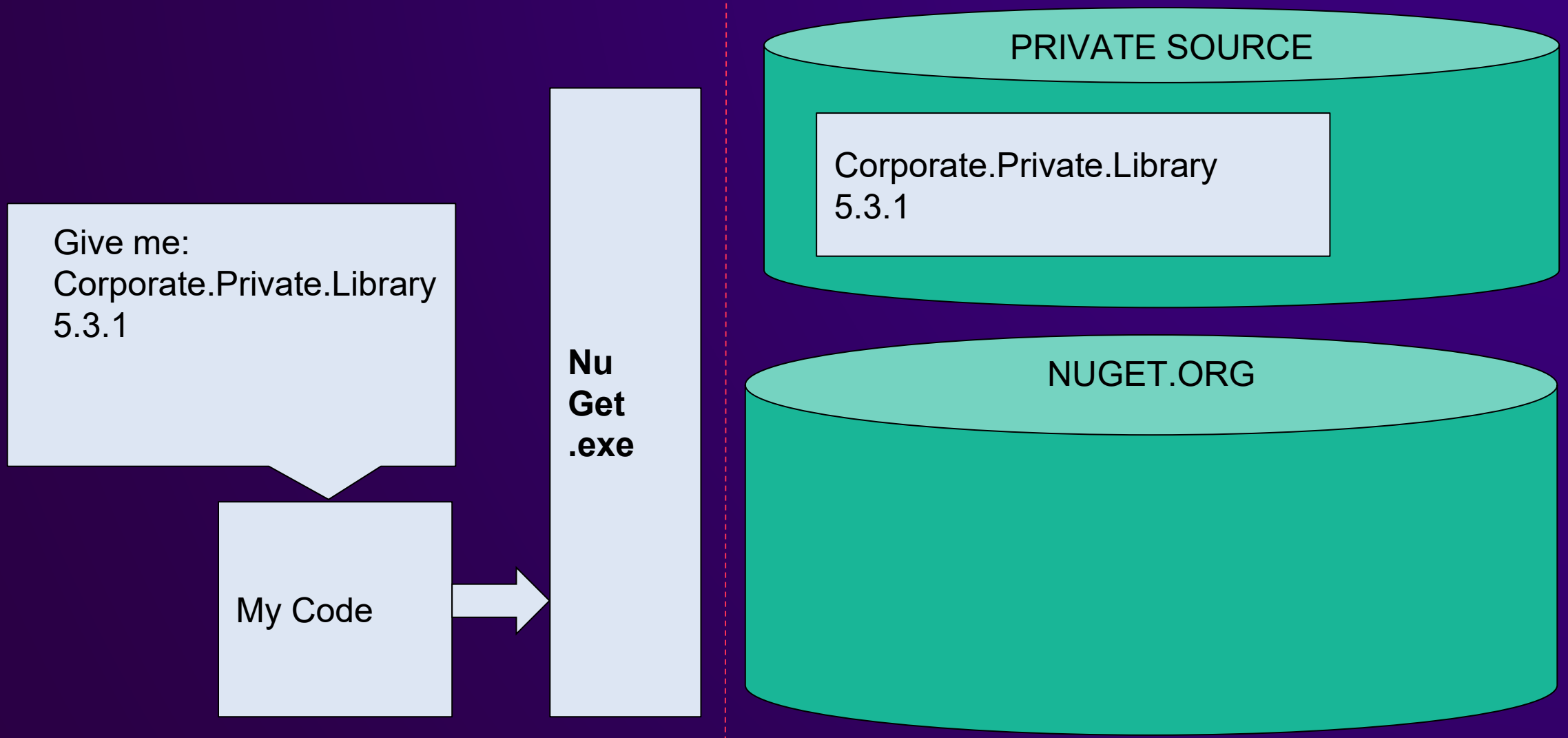


By design: NuGet/Home/issues/5611

Dependency Confusion

Mr. Evil Hacker

Dependency confusion



Dependency confusion

Give me:
Corporate.Private.Library
5.3.1

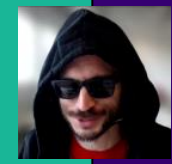
My Code

Nu Get .exe

PRIVATE SOURCE

Corporate.Private.Library
5.3.1

NUGET.ORG



Dependency confusion

Give me:
Corporate.Private.Library
5.3.1

My Code

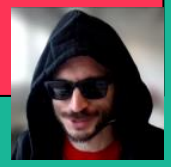
Nu Get .exe

PRIVATE SOURCE

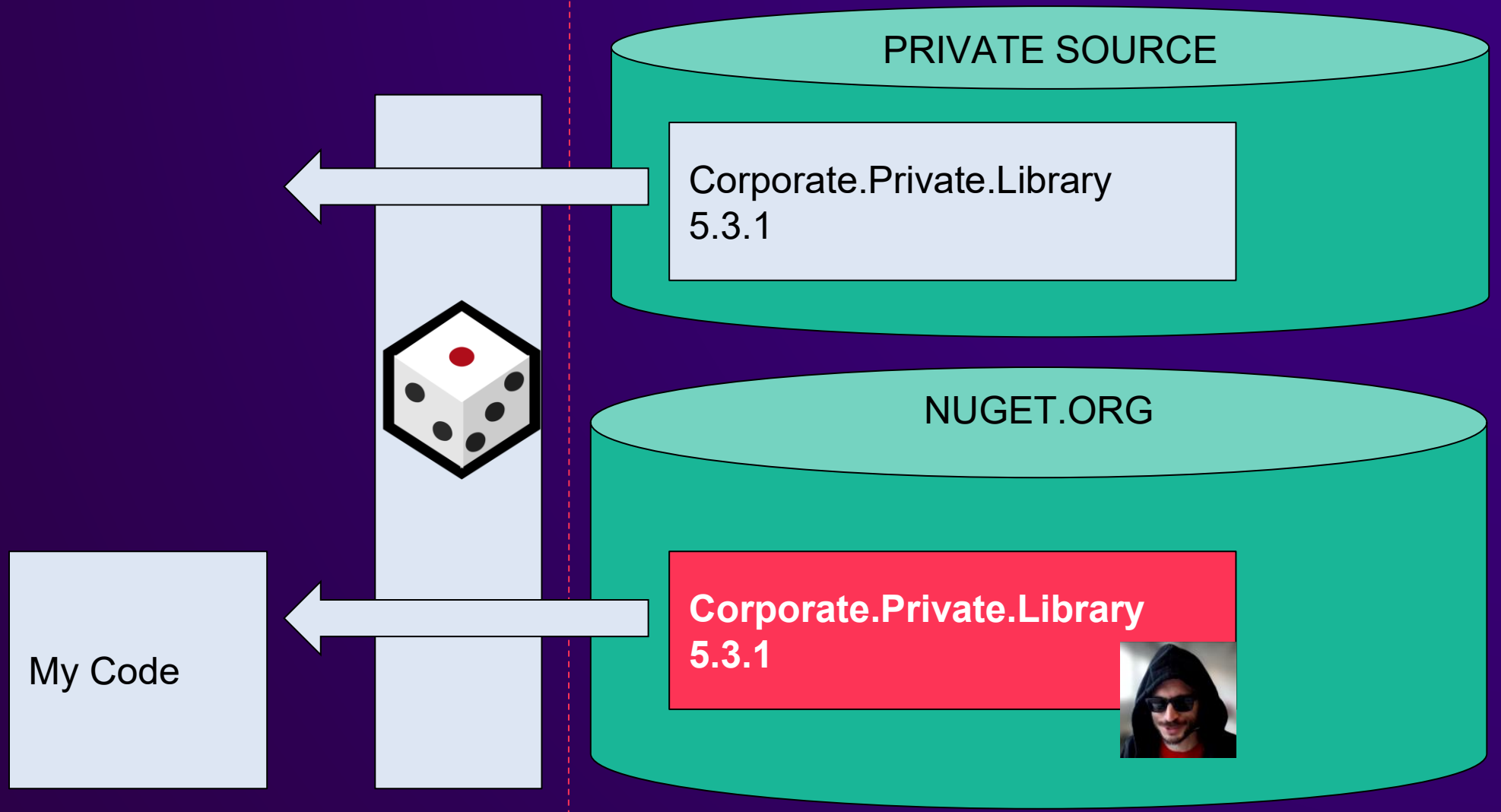
Corporate.Private.Library
5.3.1

NUGET.ORG

Corporate.Private.Library
5.3.1



Dependency confusion



Dependency confusion

Give me:
Corporate.Private.Library

5.3.*

My Code

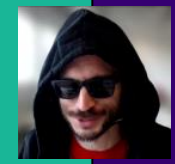
Nu Get .exe

PRIVATE SOURCE

Corpo
5.3.1

Corporate.Private.Library
5.3.2

NUGET.ORG



Dependency confusion

Give me:
Corporate.Private.Library

5.3.*

My Code

Nu Get .exe

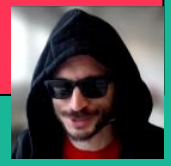
PRIVATE SOURCE

Corpo
5.3.1

Corporate.Private.Library
5.3.2

NUGET.ORG

Corporate.Private.Library
5.3.999



Dependency confusion

Here you are:

Corporate.Private.Library
5.3.999

Nu Get .exe

My Code

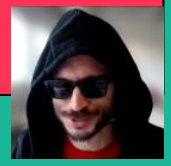
PRIVATE SOURCE

Corpo
5.3.1

Corporate.Private.Library
5.3.2

NUGET.ORG

Corporate.Private.Library
5.3.999



Dependency confusion

In 2020, security researcher Alex Bîrsan 🇷🇴 used

dependency confusion to hack into:

- Microsoft — $ 40K
- Apple — $ 30K
- Shopify — $ 30K
- Paypal — $ 30K
- … and another 31 big companies

Bug bounties

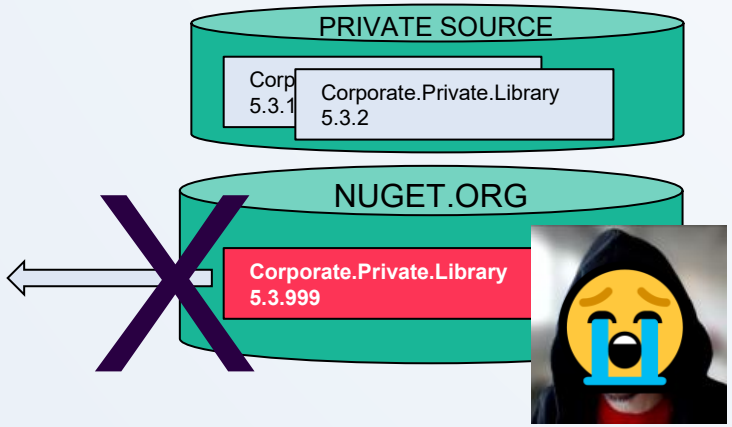
Source: medium.com/@alex.birsan

Dependency confusion DEMO

☝ Dependency confusion defense

Use `packageSourceMapping`

Map `package` name patterns to `packageSource`



☝Summary

Attacks

Goal: code execution, data exfiltration.

Typosquatting Humans introduce typos.

Attackers publish packages with typos on public repositories (e.g. gogle vs google).

Dependency confusion Attackers publish malicious packages with the same names as private ones on public repositories.

Defense

Enable `signatureValidationMode`.

Declare `owners` you trust under `trustedSigners`.

Use `packageSourceMapping`

or use a single package source.

`<clear />` to avoid system defaults.

Reserve prefixes on nuget.org for both your public and private packages.

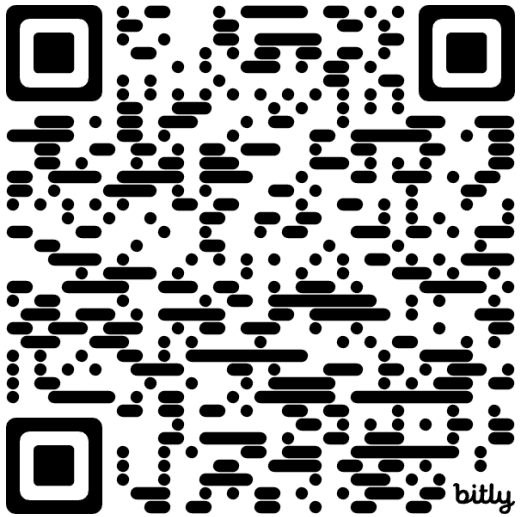
Sign your packages.

Use precise versions for dependencies.

Inspect dependencies before usage.

More details in: How to secure your NuGet supply chain

Q & A



feedback form & slides on

AndreiEpure.ro