NuGet: Things You Didn’t Know

Claire Novotny

Senior Product Manager, NuGet

@clairernovotny

Package Authors vs Consumers

How many of you have authored or maintain a NuGet package?

- OSS or internal?

How many of you consume internal packages for work?

- Share common code with other teams?

How many of you are familiar with multi-targeting?

- What are some of the reasons you use it?

Authoring Best Practices

PackageIcon

ReadMe

License

Symbols

Source Link

Package Validation

Potential Issues

Breaking changes

Mismatching public API & impl

Incompatibilities across TFMs

Validations

Baseline version

Compatible runtime

Compatible framework

Multi-targeting

Create libraries that work on multiple platforms

- Might need native interop

Why not just .NET Standard 2.0?

- .NET Framework
- New API's: Performance, new features like `Span<T>`

Different dependencies based on target platform

Critical that public surface area is compatible

Versioning

Lots of opinions… we aim for **immutable, guaranteed reproducibility**

- Tags can be changed
- Build / env variables can be changed

NerdBank.GitVersioning

- Part of the .NET Foundation
- Version increment per commit, branch patterns for prerelease
- Version bumps in file
- CLI / MSBuild Task / Node.js support

Supply Chain Trust

Whole other talk!

Key points

- Increasingly sophisticated attackers
- Target dependencies (libraries, packages, build systems, repositories, etc)

Recent Incidents

- 2016 – left-pad (npm)
- 2017/2018 – Python Package Highjacking (PyPi)
- 2018 – event-stream (npm)
- 2019 – Account takeovers of popular packages (Ruby Gems)
- 2021 – Dependency confusion (all)
- 2021 – Log4J (maven)
- 2022 – node-ipc and peacenotwar (npm)

Think of it this way…

We download **code**
From the **internet**
Written by **unknown individuals**
That we **haven’t read**
That we **execute**
With **full permissions**
On our **trusted devices**
Where we keep our **most important data**

It’s a miracle this all works

Supply Chain Trust – What is NuGet doing?

Package Source Mapping
- Source mapping tool

Package Signing
- Tamper resistance
- Provenance: see what feed it came from, who created it
- Specify allowed publishers

CVE alerts / auditing
- Vulnerability alerts on CLI

Mandatory 2FA on NuGet.org

Prefix Reservations

Central Package Management

Manage common dependencies for many projects

`Directory.Packages.props`

- `PackageVersion` items, `Version` attribute

Overrides: `PackageOverride` attribute

Transitive pinning: `CentralPackageTransitivePinningEnabled`

Requires NuGet 6.2 / .NET SDK 6.0.300 / VS 17.2 p3

- Works with legacy projects, just needs newest tooling

Upcoming Plans - Highlights

Package Scoring

Machine readable dotnet list package commands

Package Source Mapping tooling

See transitive deps in Package Manager UI (PMUI)

Search by TFM

Latest on GitHub: https://aka.ms/nuget/roadmap

Questions?

Twitter: @nuget

Thank you!