NDC { London }

Niels Tanis

Reducing Third-Party Security Risk in .NET Core Applications

Who am I?

- Niels Tanis
  - Security Researcher @ Veracode
  - Background in .NET Development
  - Application Security Consultancy
  - Pen-testing & Ethical Hacking
  - ISC^2 CSSLP

Reducing Third-Party Security Risk in .NET Core Applications

@nielstanis

Agenda

- Third-Party Security Risks
- ASP.NET Core MVC Demo App
- Compartmentalization of functionality
- Security review of framework and API’s
- Conclusion
- Q&A

Third-Party Security Risks

- .NET Core applications build on dependencies
- Developer needs to understand library and use as intended
- What do this projects do for security?
- Analyze dependencies for security issues and keep up-to-date
- What about transitive dependencies?

State of Software Security

- Volume 9 – 2018 edition
- 85.7% of the analyzed .NET applications contains at least 1 vulnerable component
- 87.5% for Java and 92% for C++
- https://veracode.com/soss

Equifax data-breach

- September 2017
- Apache Struts

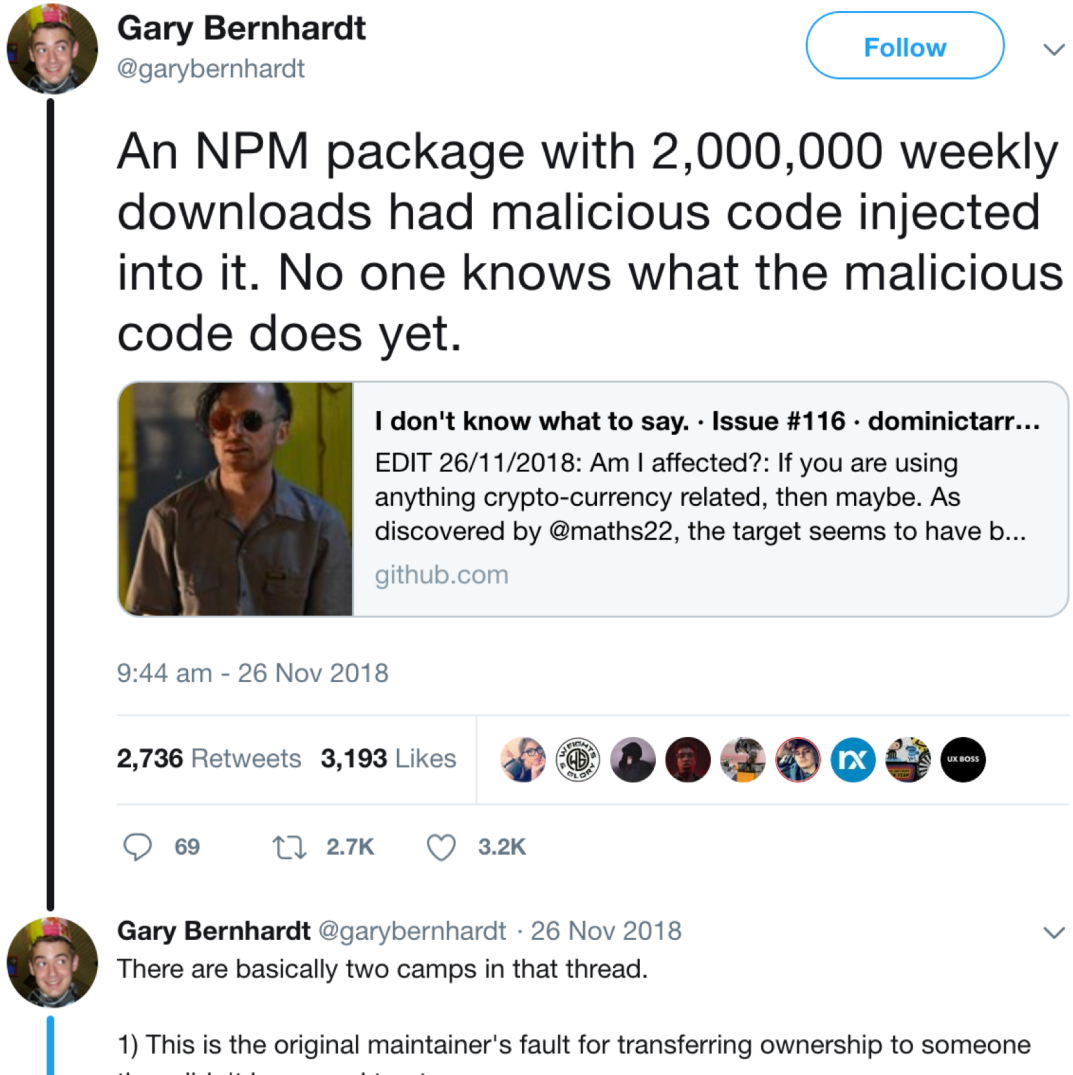
  CVE-2017-5638

Supply Chain Attacks

- Build pipelines, infrastructure as code
- Malicious functionality added for example malware/miners

EvenStream NPM

- November 2018
- Is transitive dependency of 2000 other libraries



Supply Chain Attacks

- What about signed packages?
  - Phil Haack on NuGet signing
    - https://haacked.com/archive/2019/04/03/nuget-package-signing/
  - Alternatively trust NuGet owners
    - https://haacked.com/archive/2019/05/10/friend-signing-packages/
- Central governance of CI/CD pipelines with observability

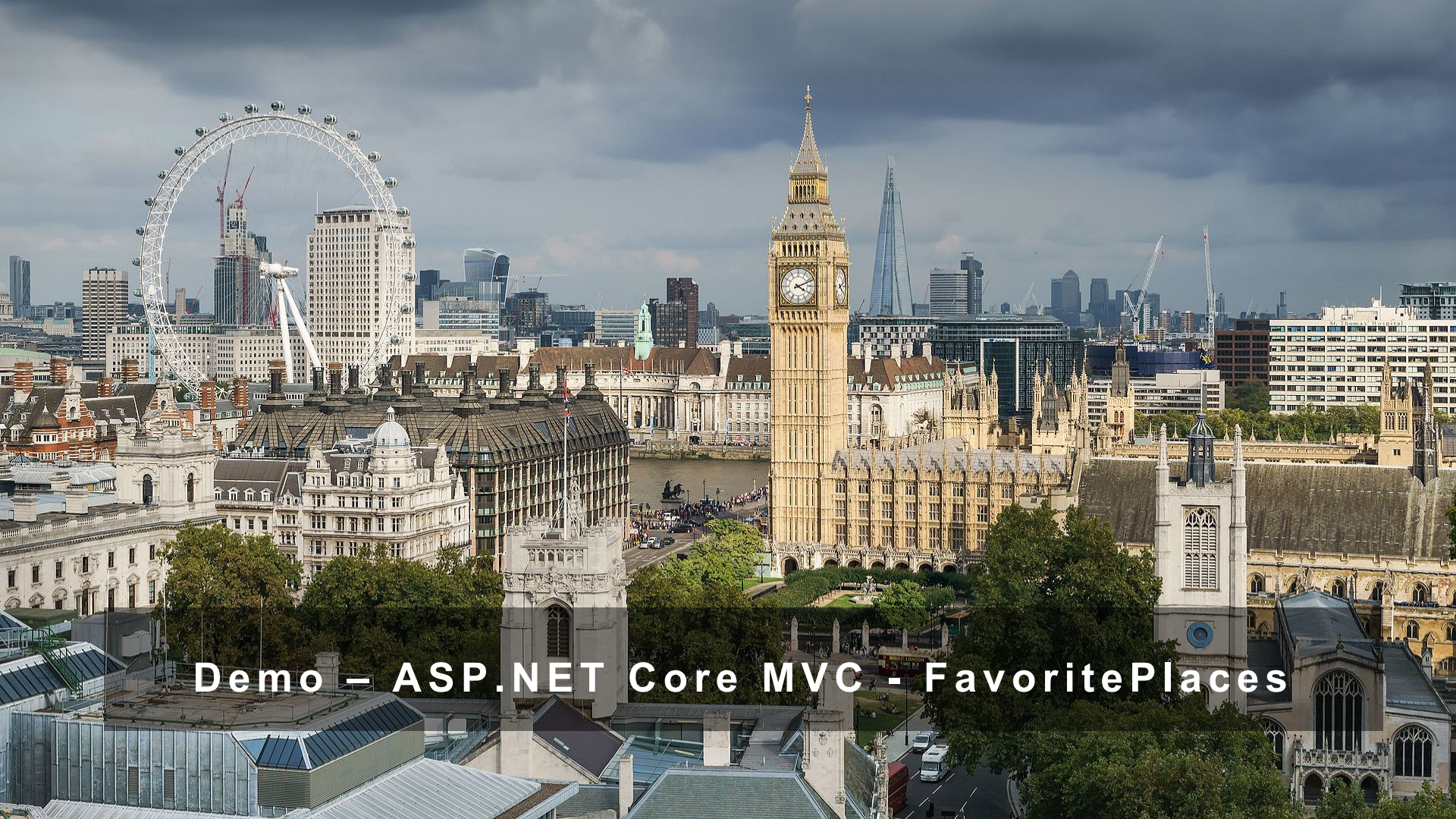
ASUS Live Update

- March 2019
- Kaspersky Labs identified malicious functionality

Costin Raiu 
@craiu

Asus Live Updater was used in a big supply chain attack we dubbed Operation #ShadowHammer. We estimate this may have affected over 1 million computer users between June and Nov 2018.

Hackers Hijacked ASUS Software Updates to Install Backdoors on T...



Demo – ASP.NET Core MVC - FavoritePlaces

Security vulnerabilities in FavoritePlaces

- Path traversal in report PDF (CWE 73)
  - Path dynamically created on user (controlled) input
  - Ability to write output PDF files to other location
- Consider following input for **Title**
  - /PDF/**Amersfoort**.PDF
  - /PDF/**../Amersfoort**.PDF
  - /PDF/**../wwwroot/Amersfoort**.PDF

Security vulnerabilities in FavoritePlaces

- Server-Side Request Forgery (SSRF) in PDF image (CWE 918)
- Information disclosure
  - file://etc/passwd
  - http://localhost:3306
  - http://169.254.169.254/metadata/attested/document?api-version=2018-10-01
  - Capital One data breach, August 2019
- ‘Blind’ variant, timing can be used as side-channel

Compartmentalization of functionality

- Are we able to isolate the functionality?
- Within the same process, using C# code
- Using the libraries as is without changing?
- Maybe we can create a sandbox for the PDF logic?
- A sandbox with limited capabilities

System.AppDomain

- Create isolation boundary for security, reliability, and versioning, and for unloading assemblies
- Code Access Security (CAS)
  - ASP.NET and Trust-Levels
  - As from 2017 advised not to be used as security boundary
- Not fully supported in .NET Core

System.Runtime.Loader.AssemblyLoadContext

- Create a scope for loading, resolving, and potentially unloading a set of assemblies.
- Multiple versions of the same assembly in same process
- Self-Contained Deployment (SCD)
- Inspired by .NET Core Plugins library - Nate McMaster

Demo - System.Runtime.Loader.AssemblyLoadContext

System.Runtime.Loader.AssemblyLoadContext

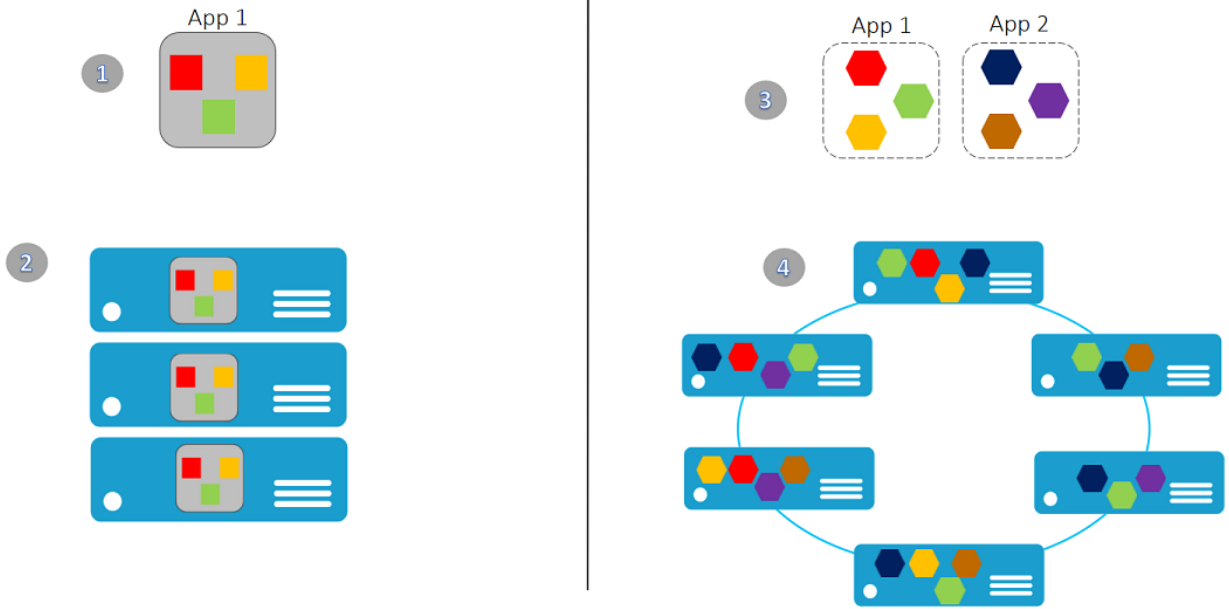
- Library **MyPDFLibrary** implementing shared interface **IPDFService**
- Self-Contained Deployment of **MyPDFLibrary**
- Use **IsolatedLoadContext** in DI to create instance of **IPDFService**
- Removal/replacement of **System.Net.Http.dll**
  - Extend functionality of .NET IL Linker (.NET Core 3.0)
  - Harmony (transpiler) - https://github.com/pardeike/Harmony

Inter-process Communication

- Create a separate host process for **MyPDFLibrary**
- Shared interface for named-pipe (**System.IO.Pipes**) communication or use gRPC as from .NET Core 3.0
- Improvement: Use **byte[]** or **Span<byte>** for image input and output PDF
- Demo is in GitHub project

Changes in Software Architecture

- Monolith → Microservices → Serverless



Review Third-Party Libraries

- Review internals and intent of library/framework
- Components vs Frameworks
- **Make sure to comply with license of component!**

What's this?

```
namespace WebApp.Controllers
{
    public class DataController
    {
        public string GetFileData(string input)
        {
            return System.IO.File.ReadAllText(input);
        }
    }
}
```


Demo – Review of AwesomePDF component

Review of AwesomePDF

- Pretty minimal implementation, uses **iText7** library
- Path problem best to be fixed with process isolation
- Two spots of usage **System.Net.Http.HttpClient**

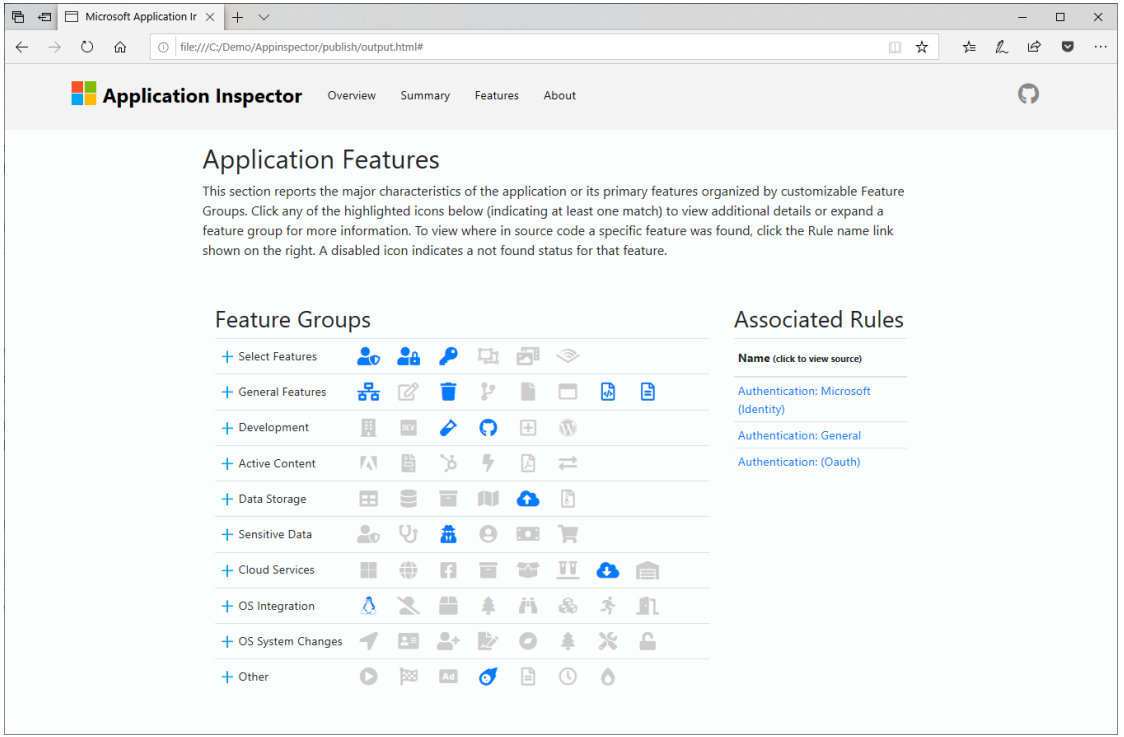
Fennec.NetCore

- DotNetCLI tool v0.4.0 on NuGet
- https://fennec.dev
- Diff on API's versions of Assemblies
- Skim for dangerous API usage

Microsoft Application Inspector

- Microsoft Application Inspector is a software source code analysis tool that helps identify and surface well-known features and other interesting characteristics of source code to aid in determining **what the software is** or **what it does**.
- https://github.com/microsoft/ApplicationInspector

Microsoft Application Inspector



Conclusion

- Start reviewing used third-party frameworks & components!
- Compartmentalize logic in code or by process to reduce risk
- Integrate security into your development processes

Thanks! Questions?

VERACODE

You change the world, we'll secure it.

https://github.com/nielstanis/ndclondon2020

ntanis at veracode.com

@nielstanis on Twitter