



NDC { Oslo }

VERACODE

You change the world, we'll secure it.

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
 - ISC² CSSLP



Reducing Third-Party Security Risk in .NET Core Applications



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 % of the analyzed applications contains at least 1 vulnerable component
- Open Source Edition – May 2020
 - 70% of the analyzed applications contains at least 1 vulnerable component
- <https://veracode.com/soss>



Equifax data-breach

- September 2017
- Apache Struts

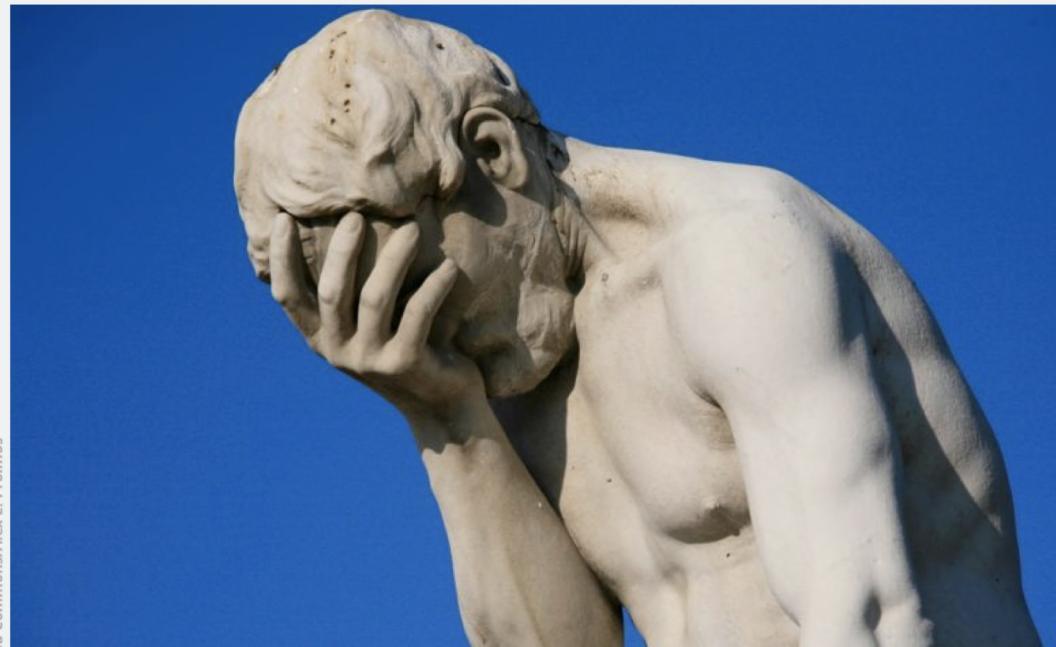
CVE-2017-5638

BIZ & IT —

Failure to patch two-month-old bug led to massive Equifax breach

Critical Apache Struts bug was fixed in March. In May, it bit ~143 million US consumers.

DAN GOODIN - 9/14/2017, 5:12 AM



Alex E. Proimos via Commons/Alex E. Proimos

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



Gary Bernhardt
@garybernhardt

Follow

An NPM package with 2,000,000 weekly downloads had malicious code injected into it. No one knows what the malicious code does yet.



I don't know what to say. · Issue #116 · dominictarr...

EDIT 26/11/2018: Am I affected?: If you are using anything crypto-currency related, then maybe. As discovered by @maths22, the target seems to have b...

github.com

9:44 am - 26 Nov 2018

2,736 Retweets 3,193 Likes



69 2.7K 3.2K



Gary Bernhardt @garybernhardt · 26 Nov 2018

There are basically two camps in that thread.

1) This is the original maintainer's fault for transferring ownership to someone they didn't know / trust.



Octopus Scanner

- May 2020
- GitHub Security

May 28, 2020

The Octopus Scanner Malware: Attacking the open source supply chain



Alvaro Muñoz

Securing the open source supply chain is an enormous task. It goes far beyond a security assessment or just patching for the latest CVEs. Supply chain security is about the integrity of the entire software development and delivery ecosystem. From the code commits themselves, to how they flow through the CI/CD pipeline, to the actual delivery of releases, there's the potential for loss of integrity and security concerns, throughout the entire lifecycle.

In the past few years the open source supply chain experienced a variety of attacks. From developer credential hijacks aimed at introducing backdoors, like in the [event stream incident](#), to a seemingly nonstop stream of typosquatting attacks against popular package managers such as [npm](#) and [pypi](#).

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
- Supply Chain Attacks much more complex!

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 report 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)
 - Mostly known 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)
- Initial version inspired by .NET Core Plugins library - Nate McMaster



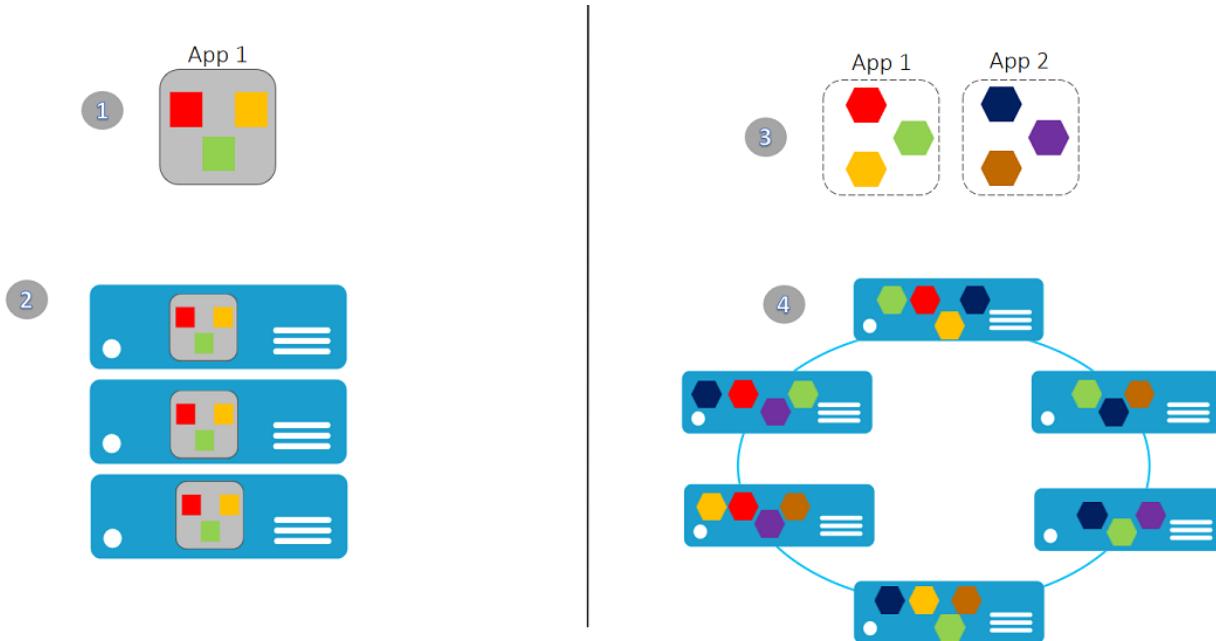
Demo - System.Runtime.Loader.AssemblyLoadContext

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)
 - Fennec.NetCore .NET CLI Patcher (future version)

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 can be fixed with micro-patch or separate process
- Two spots of usage **System.Net.Http.HttpClient**

Fennec.NetCore

- DotNetCLI tool v0.4 on NuGet
- <https://fennec.dev>
- Diff on API's versions of Assemblies
- Skim for dangerous API usage
- v0.7 → Micro Patching of Assemblies



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

The screenshot shows a browser window titled "Microsoft Application Inspector" displaying the "Application Features" page. The page includes a navigation bar with links for Overview, Summary, Features, and About, along with a search bar and a user profile icon.

Application Features

This section reports the major characteristics of the application or its primary features organized by customizable Feature Groups. Click any of the highlighted icons below (indicating at least one match) to view additional details or expand a feature group for more information. To view where in source code a specific feature was found, click the Rule name link shown on the right. A disabled icon indicates a not found status for that feature.

Feature Groups

Icon	Group
User	Select Features
Lock	General Features
Key	Development
Database	Active Content
Cloud Storage	Data Storage
Cloud	Sensitive Data
Cloud	Cloud Services
Cloud	OS Integration
Cloud	OS System Changes
Cloud	Other

Associated Rules

Name
Authentication: Microsoft (Identity)
Authentication: General
Authentication: (Oauth)

Conclusion

- Start reviewing used third-party frameworks & components!
- Compartmentalize logic in code and/or by process, consider the micro-patching of assemblies.
- Still need for input validation and defensive coding, integrate security into your development lifecycle.

A large, semi-transparent network graph composed of blue dots and connecting lines, forming a globe-like shape in the background.

Thanks! Questions?

VERACODE

You change the world, we'll secure it.

<https://github.com/nielstanis/ndcoslo2020>

ntanis at veracode.com

@nielstanis on Twitter