SECURITY BOOT CAMP FOR .NET DEVELOPERS

Philippe Arteau
Security Researcher for GoSecure

12/03/2018
WHO AM I?

▪ Philippe Arteau
▪ Security Researcher at GoSecure
▪ Open-source developer
  ▪ .NET Security Guard Security Code Scan (Roslyn – Static Analysis for .NET)
  ▪ Find Security Bugs (SpotBugs - Static Analysis for Java)
  ▪ Burp and ZAP Plugins (Retire.js, CSP Auditor)

▪ Volunteer for the nsec conference and former trainer
• Introduction
• Vulnerabilities in .NET Context
  • Path Traversal
  • XSS
  • Cryptography
  • Hardcoded secret
• Automating Checks
  • Visual Studio / MsBuild
• Recent Trends
  • Deserialization
  • Double Parsing
• Methodology for Code Review
• Conclusion
INTRODUCTION
Security Code Review

▪ Code review is the systematic examination of source code[1] with the specific goal of findings security bugs.

▪ Security Bugs?
  ▪ Injections
  ▪ XSS
  ▪ Cryptographic weakness
  ▪ Logic flaw
  ▪ And many more...

Why Security Code Review?

- Complementary to dynamic techniques (penetration testing, fuzzing, etc.)
- Every technique has its advantages

Code review advantages:
- Coverage
- Finding all instances of a vulnerability
- Accessible activity for developer
- Excellent for doing defense in depth
VULNERABILITIES IN .NET CONTEXT
Path Traversal

- SQL injection are *easy* to manage with *prepare statement*
- **Path traversal** is a source of injection that is often overlooked

- **When does it matter?**
  - File upload (writing to filesystem)
  - Document loading (reading from filesystem)
  - Can be applied in rare cases to URL\[1\]

\[1\] Example: [https://sakurity.com/blog/2015/03/15/authy_bypass.html](https://sakurity.com/blog/2015/03/15/authy_bypass.html)
- Encoding with Razor template is usually secure
  - HTML entities are escaped by default

Special cases
- Use of `@Html.Raw()`
- Placing values in JavaScript `!/\`
- JavaScript client-side template
.NET Framework provided symmetric encryption primitive

- Include the mode: CBC, ECB, OFB, ...
- Does not provided integrity
- Password
- Service account
- API keys
- Store value in configuration
- Encrypt the value

```csharp
Identity Server
new Client
{
    ClientId = "client",
    AllowedGrantTypes = GrantTypes.ClientCredentials,
    ClientSecrets =
    {
        new Secret("secret".Sha256())
    },
    AllowedScopes = { "api1" }
}
```
AUTOMATING CHECKS
Identifying bugs and vulnerabilities is nice but...
- Remediation is even better!

- Some vulnerabilities require high-level understanding of the application
Security Code Scan
Demo

roslyn + Visual Studio
RECENT TRENDS
JSON DESERIALISATION

- History repeats itself
  - 2016: Numerous Java application were found vulnerable to native deserialization
  - 2017: Researchers [1] found issues in .NET JSON serializer
    - Some libraries have issued updates
    - The vulnerability was called: JSON Friday 13th

- Two ingredients needed for a successful attack
  - Gadgets
  - Unsafe deserialization

[1] Alvaro Muñoz, Oleksandr Mirosh and James Forshaw
Affected libraries

- FastJSON
- Json.NET (use of TypeNameHandling.All)
- FSPickler
- Sweet.Jayson
- JavascriptSerializer
-DataContractJsonSerializer

What if the system validating and using the value was not the same
When parsing the following URL, what is the host?

Reference: A New Era of SSRF - Exploiting URL Parser in Trending Programming Languages!
Double Parsing: URLs

- Less likely to happen in .NET
  - Small numbers of URI parser
- High probability when interacting in other systems
- DNS rebinding needed to be considered for host whitelisting

Conclusion
- Do not trust validated input that was parsed differently
Pseudocode is highly simplified

```json
{
    "username": "philippe",
    "username": "yannlarrivee",
    "fullname": "hihihi",
    "newPassword": "C0nf00"
}
```

```csharp
{
    "username": "philippe",
    "username": "yannlarrivee",
    "fullname": "hihihi",
    "newPassword": "C0nf00"
}
```

```csharp
using Newtonsoft.Json;
{
    "username": "philippe",
    "username": "yannlarrivee",
    "fullname": "hihihi",
    "newPassword": "C0nf00"
}
```
METHODOLOGY FOR CODE REVIEW
First thing first, code review is ONE of the security activities that need to be integrated in the development lifecycle.
1. Threat modeling [1]
2. Analysis
3. Reporting (Document or Opening ticket) *
4. Bug fixing *

* Not covered in this presentation
Identify assets to protect
- Personal information
- Documents
- Passwords

Identify entry points
- MVC Controller
- Web Services
- Forms

Identify external dependencies

Imagine possible threats (STRIDE: Spoofing, Tampering, Information Disclosure, Denial of Service, Elevation of privilege)
1. Tools configuration
2. Automate scan
   ▪ Review potential issues
3. Manual review

Static analysis tools can also be run in parallel with the manual review.
- Mapping between inputs and APIs
- Categories of bugs
  - Injection
  - Path traversal
  - Static IV (Cryptography)
  - Deserialization
▪ APIs are not vulnerable by default
▪ APIs are designed to be used in a certain context
▪ Categories of bugs
  ▪ Random number generation
  ▪ Oracle Padding Attack or any other active attack
  ▪ Control based on Host header
  ▪ Insecure communication (internal communication vs network communication)
  ▪ Configuration files vs Upload files
### Checklist

- Intended for baseline verifications
- Guidelines
- Reproducibility

GOOD RESOURCES

- Code Review Guide
- Development Lifecycle
- Verification List
Conclusion

- Code review is a powerful technique to find security bugs
  - But don’t forget to do dynamic tests as well

- Use tools when possible
  - Build or extends tools when needed

- Recent trends affecting .NET
  - Angular XSS (Client-side template injection)
  - Deserialization vulnerabilities
  - Double parsing
▪ OWASP .NET Project
https://www.owasp.org/index.php/Category:OWASP_.NET_Project

▪ .NET Security Cheat Sheet

▪ Security Code Scan
https://security-code-scan.github.io/

https://www.youtube.com/watch?v=Ip6wrpYFHhE

Roslyn Wiki

https://github.com/dotnet/roslyn/wiki

Learn Roslyn Now: Part 10 Introduction to Analyzers by Josh Varty

https://joshvarty.wordpress.com/2015/04/30/learn-roslyn-now-part-10-introduction-to-analyzers/

.NET Compiler Platform SDK

- Ysoserial.net: Payload generator
  https://github.com/pwntester/ysoserial.net
- Friday The 13th JSON Attack - White Paper
QUESTIONS ?

Contact
✉ parteau@gosecure.ca
🌐 gosecure.net/blog/
🐦 @h3xStream @GoSecure_Inc