ENPM 809W
Introduction to Secure Software Engineering

Gananand Kini

Lecture 13

Debug Code

MITRE | SOLVING PROBLEMS FOR A SAFER WORLD™

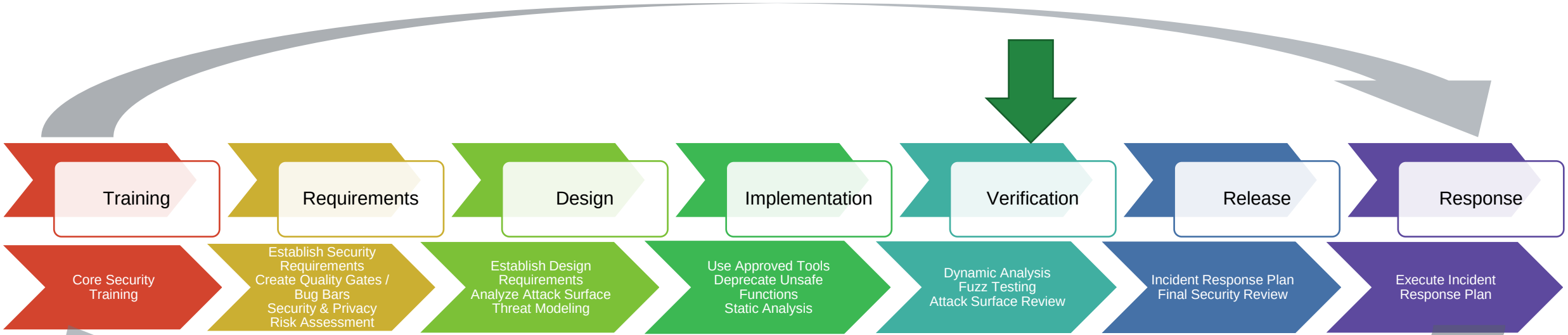
Outline

- Debug Code
- Release Proofing

Debug Code

Debug Code

- **The Verification phase is when you should perform the secure code reviews.**
- **Findings from the secure code review can include issues related to Debug Code.**



Sources:

1. https://en.wikipedia.org/wiki/Npm_(software)#Notable_breakages

Playstation 5 Debug menu anyone?

- **Forget getting your hands on one of the consoles, someone already managed to get access to the ‘Debug settings’ menu on the console.**
- **Original vendor probably meant to have authorized users access it for troubleshooting or debugging console problems for warranty purposes.**

The crucial thing here, as Wololo.net points out, is that he managed to activate that menu on a retail console rather that a PS5 testkit.

Nguyen said that he has no plans to disclose how he managed to access the debug menu, while Fail0verflow notably waited for Sony to patch the PS4 before publishing the details of its hack. So these achievements may not go anywhere, despite being fairly big news.

Source:
Tom Pritchard. PS5 Hacks could bring us one step closer to a full jailbreak. https://www.tomsguide.com/news/ps5-hacks-could-bring-us-one-step-closer-to-a-full-jailbreak. Retrieved Nov. 14, 2021.

CWE-489: Active Debug Code

- **Description: The application is deployed to unauthorized actors with debugging code still enabled or active, which can create unintended entry points or expose sensitive information.**
- **A common development practice is to add "back door" code specifically designed for debugging or testing purposes that is not intended to be shipped or deployed with the application.**
- **These back door entry points create security risks because they are not considered during design or testing and fall outside of the expected operating conditions of the application.**

Sources:
1. CWE-489: Active Debug Code. MITRE CWE. https://cwe.mitre.org/data/definitions/489.html. Retrieved July 20, 2021.

Demotivational Poster of the Day…

Put something here later …

LAZINESS

Testing Code

- **Is the following test code ok to be included in the binary?**

```
protected readonly IHostingEnvironment HostingEnvironment;

public TestController(IConfiguration configuration, IHostingEnvironment hostingEnv){
    this.Configuration = configuration;
    this.HostingEnvironment = hostingEnv;
}

[HttpGet]
public IActionResult Test(){
    if(this.HostingEnvironment.IsDevelopment()){
        System.Diagnostics.Debug.WriteLine("Checking Test() controller method");
        // Set env configuration
    }

    return View();
}
```

Don’t leave debug code around …

- Might be tempting to just submit code for code review leaving debug code in the code base…
- DON’T DO IT!
- The code review process should highlight these code instances even if the developer tells a reviewer to ignore those.
- At the very least, they should be included as an INFO level finding saying that these should be removed for the final secure code review that is performed in the Release phase.

Android Qualcomm Innovation Center (QuIC) patch enables logging sensitive information.

- A boolean flag allows for sensitive information to be logged to the System log which is accessible by any user.
- In this case the sensitive information was the disk encryption password.
- Logging and debug issues can often go hand in hand…

Logging of potentially sensitive information via NativeDaemonConnector (CVE-2013-2599)

Release Date:
July 3, 2013

Affected Projects:
Android for MSM, QRD Android

Advisory ID:
QCIR-2013-00003-1

CVE ID(s):
CVE-2013-2599

Summary:
The following security vulnerability has been identified in the NativeDaemonConnector class.

CVE-2013-2599:
Due to the state of a boolean variable within the NativeDaemonConnector class, messages passed to its log method will be logged in the system log. In some cases this can result in unwanted logging of potentially sensitive information such as the disk encryption password when MountService is instantiating NativeDaemonConnector to pass and log communication to vold. The messages from the system log can be accessed by an adversary, e.g., through the logcat functionality.

AccessVector: local
Security Risk: high

Source:
1. CVE-2013-2599. https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2013-2599.

Remember CWE-532: Insertion of Sensitive Information into Log File?

- **Description: Information written to log files can be of a sensitive nature and give valuable guidance to an attacker or expose sensitive user information.**
- **While logging all information may be helpful during development stages, it is important that logging levels be set appropriately before a product ships so that sensitive user data and system information are not accidentally exposed to potential attackers.**
- **Different log files may be produced and stored for:**
  - Server log files (e.g. server.log). This can give information on whatever application left the file. Usually this can give full path names and system information, and sometimes usernames and passwords.
  - log files that are used for debugging

Sources:
1. CWE-532: Insertion of Sensitive Information into Log File. MITRE CWE. https://cwe.mitre.org/data/definitions/532.html. Retrieved July 20, 2021.

Android Logcat and ADB

- ADB is the Android Debugger which allows anybody to connect to an unlocked Android device over USB.
- Logcat is the logging system in Android.
- You are able to view all log messages on an unlocked Android phone by connecting over USB.
- At one point, no developer mode was needed on Android to access Logcat messages.
- This allowed anyone with a USB connection to view any and all log messages.

CWE-1295: Debug Messages Revealing Unnecessary Information

- **Description: The product fails to adequately prevent the revealing of unnecessary and potentially sensitive system information within debugging messages.**
- **Debug messages are messages that help troubleshoot an issue by revealing the internal state of the system.**
- **For example, debug data in design can be exposed through internal memory array dumps or boot logs through interfaces like UART via TAP commands, scan chain, etc.**
- **The more information contained in a debug message, the easier it is to debug. However, there is also the risk of revealing information that could help an attacker either decipher a vulnerability, and/or gain a better understanding of the system.**
- **This extra information could lower the "security by obscurity" factor. While "security by obscurity" alone is insufficient, it can help as a part of "Defense-in-depth".**

Sources:
1. CWE-1295: Debug Messages Revealing Unnecessary Information. MITRE CWE. https://cwe.mitre.org/data/definitions/1295.html. Retrieved July 20, 2021.

Real World Example – Information Leak

Server Error in '/' Application.

Invalid object name 'user_acc'.

Description: An unhandled exception occurred during the execution of the current web request. Please review the stack trace for more information about the error and where it originated in the code.

Exception Details: System.Data.SqlClient.SqlException: Invalid object name 'user_acc'.

Source Error:
Line 135: sqlcmd = new SqlCommand("select uid, password from user_acc where uid='" + uidd + "'", hookup);
Line 136: hookup.Open();
Line 137: reader=sqlcmd.ExecuteReader();
Line 138:
Line 139: while (reader.Read())
Source File: c:\inetpub\vhosts\cactusindia.com\httpdocs\Default.aspx.cs Line: 137

Stack Trace:
[SqlException (0x80131904): Invalid object name 'user_acc'.]
…

Version Information: Microsoft .NET Framework Version:2.0.50727.4952; ASP.NET Version:2.0.50727.4955

Jating2. (2010, November 15). *Server error in '/' application.* Retrieved February 21, 2011, from http://forums.asp.net/p/1623626/4169865.aspx

Debug Messages can also leak information …

- As we have seen before, debug messages can also leak information to adversaries and users alike.
- Turning off Debug mode for compilation and removing debug and debug logging code is necessary before shipping a product or putting it into production…

CWE-11: ASP.NET Misconfiguration: Creating Debug Binary

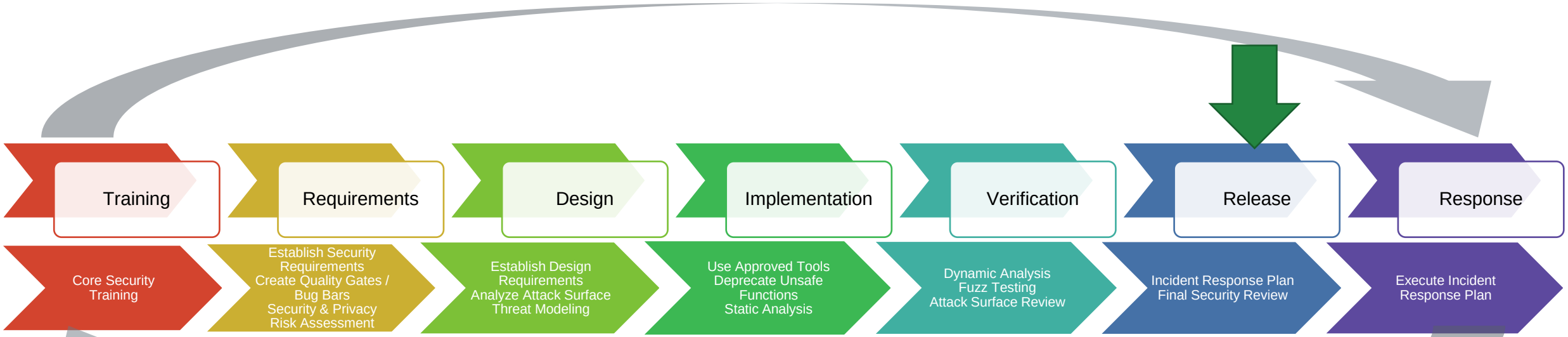
- **Description: Debugging messages help attackers learn about the system and plan a form of attack.**
- **Old ASP .NET applications can be configured to produce debug binaries. These binaries give detailed debugging messages and should not be used in production environments.**
- **Debug binaries are meant to be used in a development or testing environment and can pose a security risk if they are deployed to production.**

Sources:
1. CWE-11: ASP.NET Misconfiguration: Creating Debug Binary. MITRE CWE. https://cwe.mitre.org/data/definitions/11.html. Retrieved July 20, 2021.

Release Proofing

Release Proofing

- **The Release phase is when you should perform the final secure code review.**



Sources:

1. https://en.wikipedia.org/wiki/Npm_(software)#Notable_breakages

Release Process

- Plan the release process
- Build a release candidate
- Test the release candidate (usually User Acceptance Testing)
- Deploy the release
- Archive the release

Planning the release

- Usually involves going through all the steps (a pipeline) necessary for a release to occur with the necessary approvals etc.
- Involves checking policy requirements (release type, standards, and governance requirements) that need to be met for the release.
- A deployment plan – what should happen when and what should happen when things go wrong.
- What will be included in a release “unit”.
- What are major and minor version numbers for the release.
- How will the release be packaged?
- Who will be the release manager (responsible for ensuring the release occurs correctly)?
- And many more …

Building a Release candidate

- Most platforms and languages have a way to build a release candidate.
- Removes all debugging symbols, extra metadata etc.
- Can include using minification where all symbols in the javascript are reduced into letter identifiers using the fewest characters to try to reduce the file size.
- In .NET platform, change Build type to ‘Release’.
- In Visual Studio can follow strategy here to minify:
  - https://www.meziantou.net/minify-css-and-javascript-files-with-visual-studio-and-asp-net-core.htm
- Microsoft recommends using WebOptimizer and using it in conjunction with JS packaging tools like Gulp or WebPack. See link: https://docs.microsoft.com/en-us/aspnet/core/client-side/bundling-and-minification?view=aspnetcore-6.0.
- Use <environment/> tag in the CSHTML pages that use the normal JS libraries to specify use in development environments only.

Release proofing

- Final Code Review is performed during this step.
- If the code review passes, only then is the release package deployed into production.
- Good time to check configuration files and whether log files are being generated at the appropriate logging levels and in the correct locations.
- Also good time to finalize the documentation for the software system and include as part of the release package.

Deployment and Archival

- **After code is deployed and working:**
  - Create an archive of the release package and save it into a backup location. (Something like JFrog's Artifactory can be used)
  - Start post-release process for tracking errors, capturing user incidents and support tickets etc.