Security for Hackers and Developers: Overview

INTRODUCTION

Dr. Jared DeMott
SECURITY RESEARCHER AND ENGINEER
@jareddemott

Course Overview

Introduction

Understanding the Security Development Lifecycle - SDL

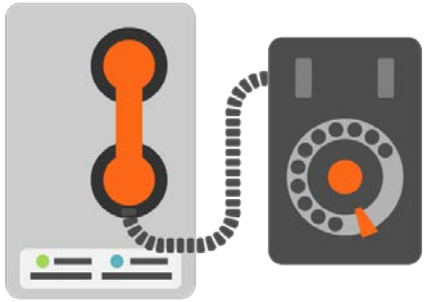
Uncovering Security Bugs

Using Security Techniques

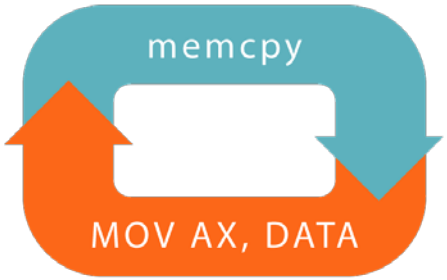
Learning from a Case Study

Four Pillars of AppSec

Code Auditing



Fuzzing



Reverse Engineering

Exploitation

Introduction

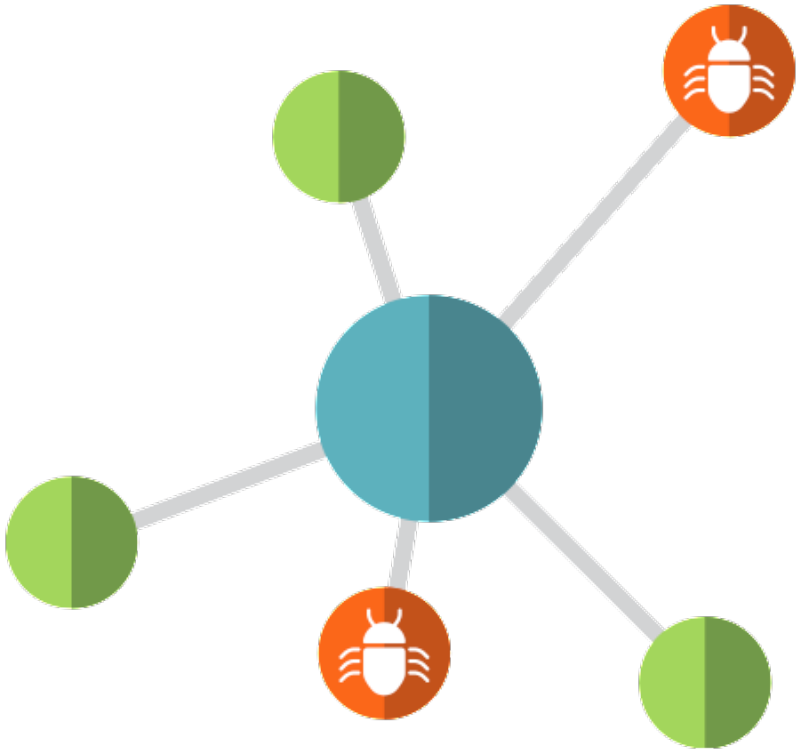
Why do bugs happen?

Ongoing process considerations

Why Do Bugs Happen?

Coding mistakes

- Tight deadlines
- Lack of proper testing

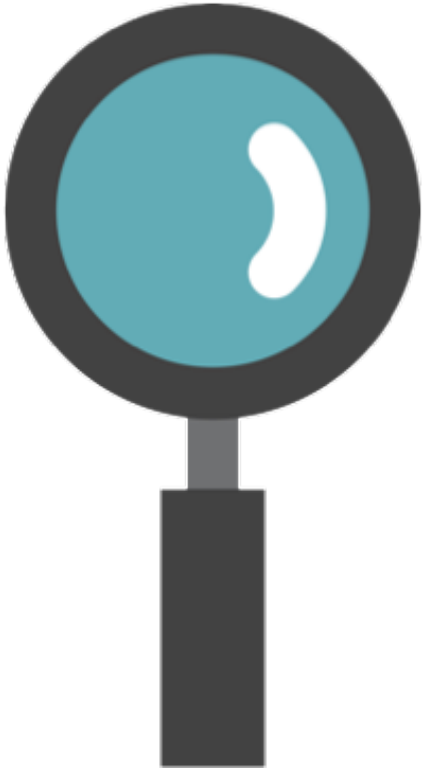


Imported bugs

- Existing bugs in libraries

Or other dependencies

- But do not develop in-house crypto

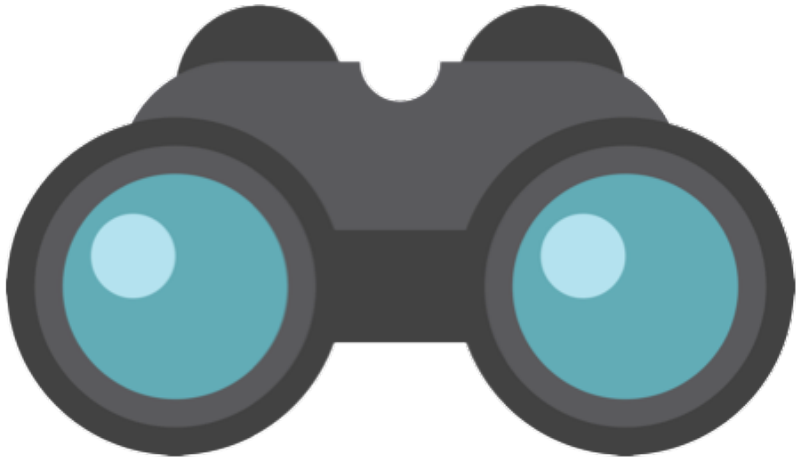


Lack of clarity

Confusing API calls

Responsibility gaps

Failure to train coders on security issues



Design issues

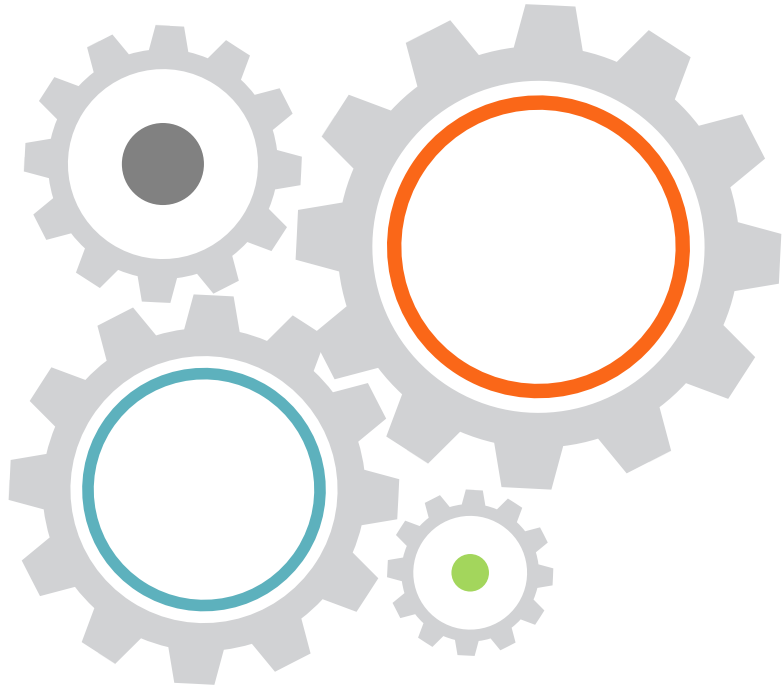
Complex Code

Not thinking about the future

Overly complex design or implementation

- Keep it simple!

Ongoing Process Considerations



Vulnerability fundamentals

- Security policy
  - Safer APIs
  - Review procedures
- Best Practices
  - Pre-audited code patterns

Design / logic review

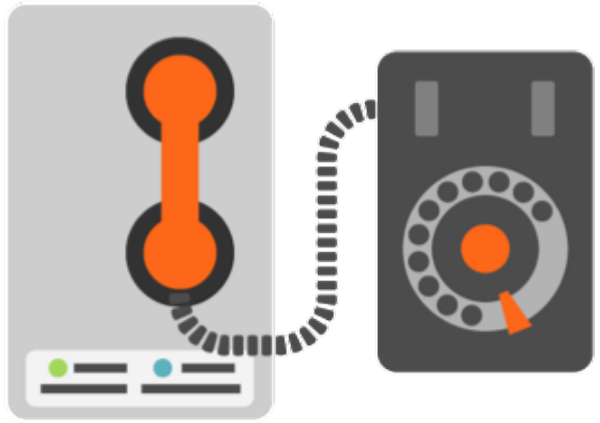
- Trust
- Algorithms
- Storing data
- Encryption

Operational review

- Systems change
  - Privileges and files
  - Process creation

Static testing

- Automation

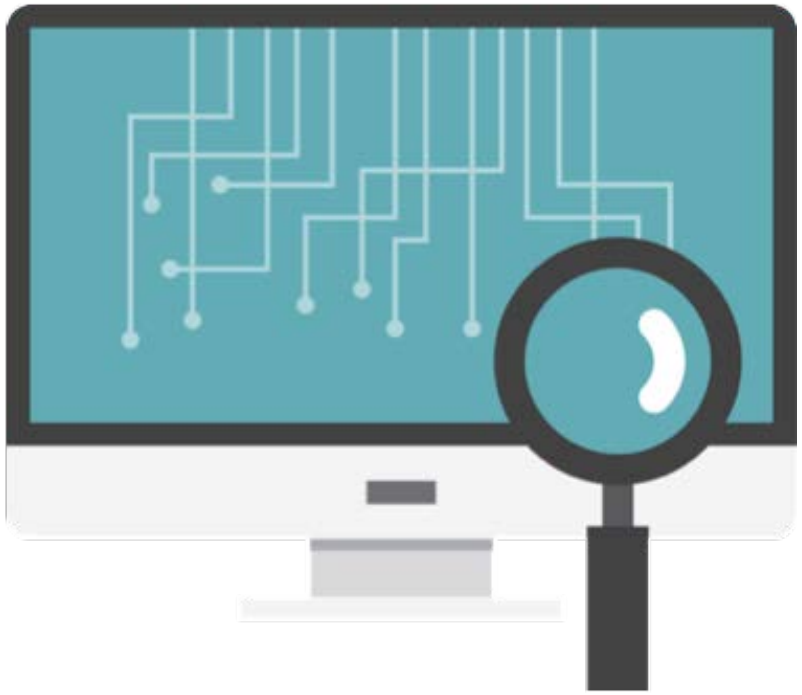


Dynamic security testing

- Fuzzing

Code review

- Tools
- Techniques



Release review

- Reverse engineering Protection
- What to do about exploits?
- Must be patchable
  - Obfuscation?

Summary

Course outline

Why do bugs happen?

Ongoing process Considerations