Uncovering Security Bugs

Dr. Jared DeMott
SECURITY RESEARCHER AND ENGINEER

@jareddemott

Overview

How do bugs manifest?

Where do they hide?

How to find them?

Data Sanitization

Many software issues result from improperly sanitized or parsed input/output data

Memory corruption from bad parsing

- File/network program in C/C++

Memory Corruption Example

```
char buf[100];
int x = atoi(argv[1]); //bug - we cannot trust user input
if ( x < 100 )
    strncpy(buf, argv[2], x);
```

Memory Corruption Example

```
char buf[100];
int x = strlen(argv[2]);  //fix - test it for ourselves
if ( x < 100 )
    strncpy(buf, argv[2], x);
```

Improperly cleansed or displayed user data

- Web bug such as SQL/XSS

Cross-site Script Example

```
url = params.get('details_url', None)
if url:
    #bug -- do not directly interpret user input as HTML
    markup = markup.replace
        (url, u'<a href="{0}">{0}</a>'.format(url))
```

Cross-site Script Example

```
url = params.get('details_url', None)
if url:
    #Fix - string will not be interpreted as script
    url = escape(url)
    markup = markup.replace
        (url, u'<a href="{0}">{0}</a>'.format(url))
```

Language vs. Typical Bug Type

Unmanaged Code

- C and C++

Scripting Languages

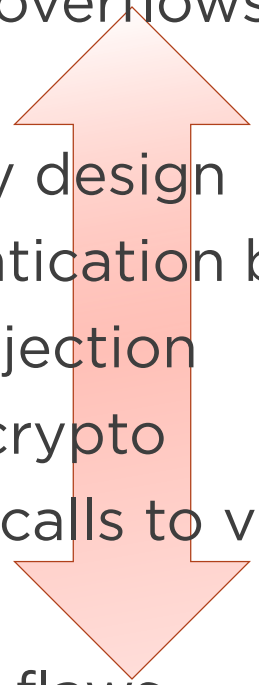
- Python, Perl, Ruby, etc.

Managed Code

- .NET (C#)

Web Application Code

- PHP, Java, ASP, etc.

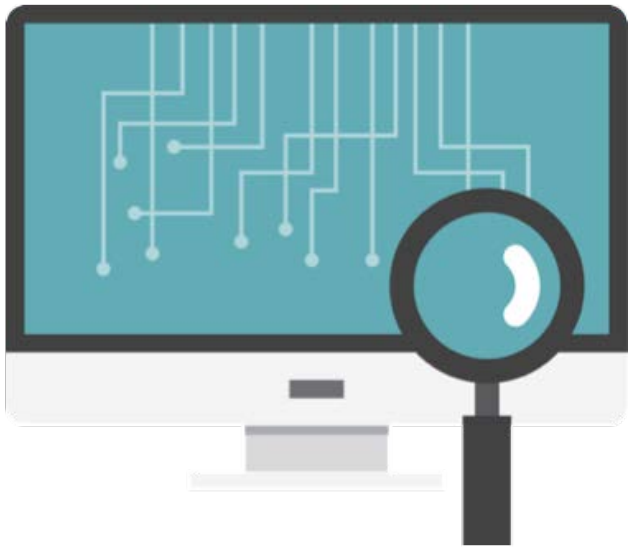


- Memory corruption bugs
  - Buffer overflows and the like
- Insecurity design
  - Authentication bypass
  - Cmd injection
  - Weak crypto
  - Native calls to vuln C
- Web app flaws
  - SQL injections, and XSS to name a couple

Software Bugs

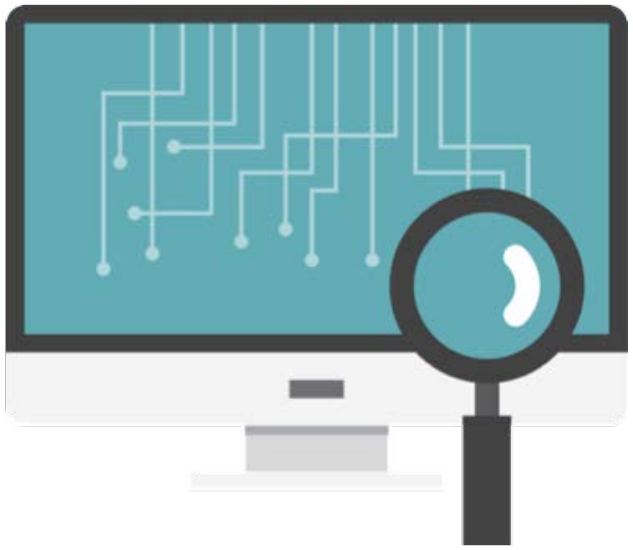
Error, flaw, failure, or fault in computer code that causes incorrect or unexpected results or unintended behaviors

But where do they live?



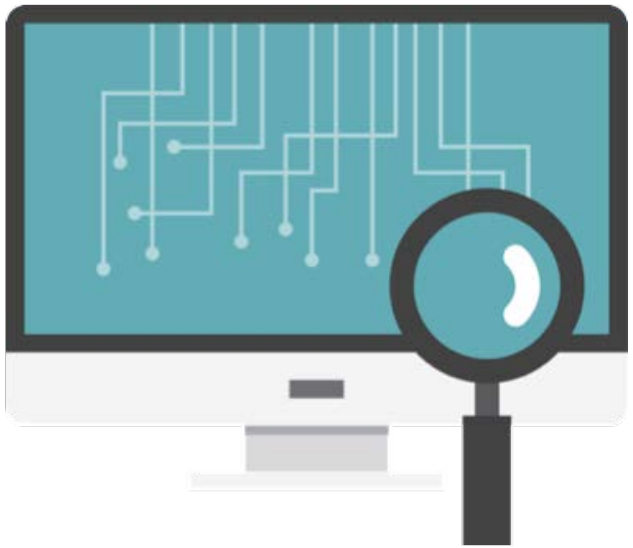
Start with attack surface

- Point at which untrusted data enters and is processed
  - E.g. Code that processes the network packet for network server



Dark Corners

- Greater chance of bugs, compared to well tested
- Cold path
  - E.g. A packet that is never sent by a real client, but code exists to handle it



Complex Components

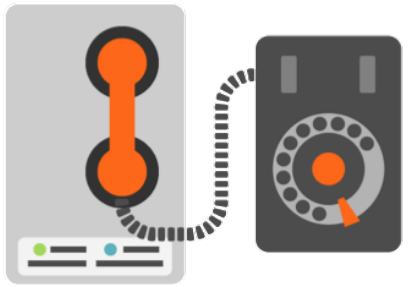
- Devs rely on, but often didn't write/fully understand
  - Low-level/Back-end components
  - Allocators
  - Garbage collection
  - JIT

Bug Discovery

Finding flaws in software

Automation or Human?

Find the Bugs in Your Code



Find parsing bugs in complex parts of code that are hard to read manually

Search for known patterns, misused APIs, etc.

Audit JavaScript event handler in a browser for UAF, because you've seen mistakes there in the past

Summary

How do bugs manifest?

Where do they hide?

How to find them?