Application Security

Abstract

We'll be exploring some of the principles of Secure Application Development over the next 30 minutes.

Andrew Millar

Information Security Specialist

Application Security

Abstract

We'll be exploring some of the principles of Secure Application Development over the next 30 minutes.

~~Andrew Millar~~

~~Information Security Specialist~~

Nic Jackson

Java developer ;-)

What is Security?

What is Security?

> security is an *emergent property* of your overall product or system health. It's not an isolated feature that gets shipped and is then relegated to minor upkeep

So what is Secure Application Development

To borrow Justin's analogy:

> The superior doctor prevents sickness. The mediocre doctor attends to impending sickness. The inferior doctor treats actual sickness.

Why should we care?

The purpose of Secure Application Development

To contribute to our business's mission through not only developing the features and business logic we require, but also in protecting our assets by implementing appropriate safe guards for them

The Foundation of Information Security

Introducing the CIA

Introducing the CIA

- Confidentiality

Introducing the CIA

- Confidentiality - only allowing access to data to which the user is permitted

Introducing the CIA

- Confidentiality - only allowing access to data to which the user is permitted
- Integrity

Introducing the CIA

- Confidentiality - only allowing access to data to which the user is permitted
- Integrity - ensuring data is not tampered with or altered by unauthorized users

Introducing the CIA

- Confidentiality - only allowing access to data to which the user is permitted
- Integrity - ensuring data is not tampered with or altered by unauthorized users
- Availability

Introducing the CIA

- Confidentiality - only allowing access to data to which the user is permitted
- Integrity - ensuring data is not tampered with or altered by unauthorized users
- Availability - ensuring systems and data are available to authorized users when they need it

Protecting the Foundation

Protecting the Foundation

- Minimize attack surface area

Protecting the Foundation

- Minimize attack surface area
- Least Privilege

Least Privilege gone wrong

CVE-2016-1247

> Nginx web server packaging on Debian-based distributions such as Debian or Ubuntu was found to create log directories with insecure permissions which can be exploited by malicious local attackers to escalate their privileges from nginx/web user (www-data)

CVE-2016-1247 Demo

Protecting the Foundation

- Minimize attack surface area
- Least Privilege
- Defense in Depth

Protecting the Foundation

- Minimize attack surface area
- Least Privilege
- Defense in Depth
- Secure Defaults

Protecting the Foundation

- Minimize attack surface area
- Least Privilege
- Defense in Depth
- Secure Defaults
- Fail Securely

Fail Securely Fail

Stolen from OWASP

```
isAdmin = true;
try {
   codeWhichMayFail();
   isAdmin = isUserInRole( “Administrator” );
}
catch (Exception ex) {
   log.write(ex.toString());
}
```

Protecting the Foundation

- Minimize attack surface area
- Least Privilege
- Defense in Depth
- Secure Defaults
- Fail Securely
- Separation of duties

Protecting the Foundation

- Minimize attack surface area
- Least Privilege
- Defense in Depth
- Secure Defaults
- Fail Securely
- Separation of duties
- Keep it Simple

Protecting the Foundation

- Minimize attack surface area
- Least Privilege
- Defense in Depth
- Secure Defaults
- Fail Securely
- Separation of duties
- Keep it Simple
- Maintain an audit trail

A good Audit Trail

Answers:

- Who
- What
- When
- Why
- How

Audit Fail

```plain Aug 18 11:00:57 [AuthController] Authentication failed. ```

Audit Win

```plain Aug 18 11:00:57 [AuthController] Authentication failure for andrww@example.com by 127.0.0.1 - user unknown - /user/login /user/myaccount Aug 18 11:01:18 [AuthController] Authentication failure for andrew@example.com by 127.0.0.1 - invalid password - /user/login?err=1 /user/login Aug 18 11:02:01 [AuthController] Authentication failure for andrew@example.com by 127.0.0.1 - incorrect 2FA code - /user/login?err=2 /user/login ```

To recap

- Minimize attack surface area
- Least Privilege
- Defense in Depth
- Secure Defaults
- Fail Securely
- Separation of duties
- Keep it Simple
- Maintain an audit trail

Audit Win

```
Aug 18 11:00:57 [AuthController] Authentication failure
for andrww@example.com by 127.0.0.1 - user unknown -
/user/login /user/myaccount
Aug 18 11:01:18 [AuthController] Authentication failure
for andrew@example.com by 127.0.0.1 - invalid password -
/user/login?err=1 /user/login
Aug 18 11:02:01 [AuthController] Authentication failure
for andrew@example.com by 127.0.0.1 - incorrect 2FA code
- /user/login?err=2 /user/login
```

So where do I start?

So where do I start?

Have a look at:

- The Open Web Application Security Project - https://www.owasp.org
- SANS - https://www.sans.org
  Particulary their Whitepaper on a Framework for Secure Application Design and Development. It's from 2002, but the ideas are still relevent - https://www.sans.org/reading-room/whitepapers/application/framework-secure-application-design-development-842
- Security Analysis tools in your language

Application Security Testing:

Much like linters, there are Analysis tools for most languages

For Example:

Ruby/Rails - Brakeman, Codesake Dawn, Space

Java - Find Security Bugs plugin for FindBugs, PMD, Infer

PHP - RIPS

Javascript/Node - scanjs plugin for eslint, RetireJS

C - Infer, FlawFinder

Go - SafeSQL, Dingo Hunter, Go StaticCheck, ieffAssign

That's all for now folks!

I hope this has been an interesting introduction to the concepts of Secure Application development!