OWASP Enterprise Security API (ESAPI) for C Plus Plus

Dan Amodio
ESAPI for C Project Leader
Dan.Amodio@owasp.org
Dan.Amodio@aspectsecurity.com

April 5th, 2012
Who am I?

- OWASP
  - ESAPI – C Project leader
  - ESAPI – C++ Contributor
- Work
  - Application Security Engineer – Aspect Security
- Experience
  - Code Reviews
  - Architecture Reviews
  - Penetration Testing
  - Software Development
- Have Wife, Daughter, Hobbies, etc.
You?

- Developers
- Managers
- Security Professionals
This Presentation

- ESAPI Project Overview
- ESAPI for C Plus Plus (yes... really.)
- Integrating Security Controls (DEMO)
- ESAPI Future (3.0)
WHAT IS ESAPI?
Free and Open Source (OWASP)

OWASP
The Open Web Application Security Project

Category: OWASP Enterprise Security API

ESAPI (The OWASP Enterprise Security API) is a free, open source, web application security control library that makes it easier for programmers to write secure applications. The ESAPI libraries are designed to make it easier for programmers to retrofit security into existing applications. The ESAPI libraries also serve as a solid foundation for new development.

For language-specific differences, all OWASP ESAPI versions have the same basic design:

- There is a set of security control interfaces. They define for example types of parameters that are passed to types of security controls.
- There is a reference implementation for each security control. The logic is not organization-specific and the logic is not application-specific. An example: string-based input validation.
- There are optionally your own implementations for each security control. There may be application logic contained in these classes which may be developed by or for your organization. An example: enterprise authentication.

This project source code is licensed under the BSD license, which is very permissive and about as close to public domain as is possible. The project documentation is licensed under the Creative Commons Attribution license. You can use or modify ESAPI however you want, even include it in commercial products.

The following organizations are a few of the many organizations that are starting to adopt ESAPI to secure their web applications: American Express, Apache Foundation, Boco Allen Hamilton, Aspect Security, Foundation (MoDoS), The Hartford, Intellive Campus, Lockheed Martin, MITRE, U.S. Navy, SPAWAR, The World Bank, SANS Institute.

Please let us know how your organization is using OWASP ESAPI. Include your name, organization's name, and a brief description of how you are using it. The project lead can be reached here. The project maintainer can be reached here.

Let's talk here

Got developer cycles? Related resources

ESAPI Communities ESAPI Coding

OWASP Cheat Sheet Series
Free and Open Source (OWASP)
Enhanced Small Arms Protective Insert
Armor for your apps
Custom Enterprise Web Application

OWASP Enterprise Security API


Your Existing Enterprise Services or Libraries
ESAPI Pattern Across Languages

- Security Control Interfaces
- Reference Implementations
- Customizable
Why Centralized Controls are Important?

![Diagram of MyEnterpriseApplication with layers: Controller, Service Layer, Data Access Layer, View Layer]

1. Controller
2. Service Layer
3. Data Access Layer
4. View Layer
Too many cooks in the kitchen!
No Central Controls
Develop Lower Risk Applications

Vulnerabilities and Security Controls

- Missing: 35%
- Broken: 30%
- Ignored: 20%
- Misused: 15%
Potential ESAPI Cost Savings

- AppSec Training
- AppSec Requirements
- AppSec Design
- AppSec Implementation
- AppSec Verification
- AppSec Remediation
- AppSec Standards
- AppSec Management

Cost Savings Comparison:
- ESAPI
- Typical
ESAPI Language Availability

- Java EE
- Dot NET
- ASP
- PHP
- ColdFusion
- Python
- JavaScript
- Objective C
- Force.com
- Ruby
- C
- C++
- Perl
## Feature Set vs. Programming Language

<table>
<thead>
<tr>
<th>Feature Set</th>
<th>Java</th>
<th>.NET</th>
<th>php</th>
<th>C#</th>
<th>ASP.NET</th>
<th>Python</th>
<th>JavaScript</th>
<th>C++</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Authentication</strong></td>
<td>2.0</td>
<td>1.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Identity</strong></td>
<td>2.0</td>
<td>1.4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Access Control</strong></td>
<td>2.0</td>
<td>1.4</td>
<td>1.4</td>
<td></td>
<td>1.4</td>
<td>1.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Input Validation</strong></td>
<td>2.0</td>
<td>1.4</td>
<td>1.4</td>
<td>1.4</td>
<td>1.4</td>
<td></td>
<td>2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Output Escaping</strong></td>
<td>2.0</td>
<td>1.4</td>
<td>1.4</td>
<td></td>
<td>1.4</td>
<td>1.4</td>
<td>2.0</td>
<td></td>
</tr>
<tr>
<td><strong>Canonicalization</strong></td>
<td>2.0</td>
<td>1.4</td>
<td>1.4</td>
<td>1.4</td>
<td>1.4</td>
<td>1.4</td>
<td>2.0</td>
<td>???</td>
</tr>
<tr>
<td><strong>Encryption</strong></td>
<td>2.0</td>
<td>1.4</td>
<td>1.4</td>
<td></td>
<td>1.4</td>
<td>1.4</td>
<td></td>
<td>2.0</td>
</tr>
<tr>
<td><strong>Random Numbers</strong></td>
<td>2.0</td>
<td>1.4</td>
<td>1.4</td>
<td></td>
<td>1.4</td>
<td>1.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Exception Handling</strong></td>
<td>2.0</td>
<td>1.4</td>
<td>1.4</td>
<td>1.4</td>
<td>1.4</td>
<td>1.4</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td><strong>Logging</strong></td>
<td>2.0</td>
<td>1.4</td>
<td>1.4</td>
<td>1.4</td>
<td>1.4</td>
<td>1.4</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td><strong>Intrusion Detection</strong></td>
<td>2.0</td>
<td>1.4</td>
<td>1.4</td>
<td></td>
<td>1.4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Security Configuration</strong></td>
<td>2.0</td>
<td>1.4</td>
<td>1.4</td>
<td>1.4</td>
<td>1.4</td>
<td>1.4</td>
<td>2.0</td>
<td>TBD</td>
</tr>
<tr>
<td><strong>WAF</strong></td>
<td>2.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
WHY ESAPI FOR C++?
Reasoning

- Sponsored by Government
- Currently ESAPI for C
- C++ is still popular and used in critical applications
Almost 40k C++ Projects on Sourceforge

http://sourceforge.net/directory/
Over 6k C++ Jobs on Dice

<table>
<thead>
<tr>
<th>Job Title</th>
<th>Company</th>
<th>Location</th>
<th>Date Posted</th>
</tr>
</thead>
<tbody>
<tr>
<td>Senior Software Engineer. C or C++, TCP/IP</td>
<td>Brains Workgroup, Inc.</td>
<td>Teaneck, NJ</td>
<td>Feb-16-2012</td>
</tr>
<tr>
<td>Senior C++ Software Developer</td>
<td>Brains Workgroup, Inc.</td>
<td>Teaneck, NJ</td>
<td>Feb-16-2012</td>
</tr>
<tr>
<td>Software Engineer - C++</td>
<td>Idea Connectors Inc</td>
<td>San Mateo, CA</td>
<td>Mar-02-2012</td>
</tr>
</tbody>
</table>
Retro-fit Existing Applications

- Critical Utilities / Systems
  - Telecom
  - Defense
  - Banking / Trading

- Enterprise Apps
  - Point of Sale
  - Employee Interfaces
  - Airline applications

- Terminal Systems

- ???
New Applications

- MMO Games
- Critical Utilities / Systems
- Embedded Applications
- Server Applications
- ???
ESAPI C++ Controls

- Authentication
- User
- Access Control
- Validation
- Encoding
- Execution
- Encryption
- Logging
Example ESAPI Integration

DEMO
Example Workflow

Generate Report

Call Batch File
Design Choices, Controls, Dependencies

ARCHITECTURE
Design Approach

- Based off Java design
- Removed Web Specifics
- Reached out to the community
ESAPI C++ Controls

Custom Enterprise Application

OWASP Enterprise Security API for C++

Your Existing Enterprise Services or Libraries
ESAPI C++ Controls

- Authentication
- User
- Access Control
- Validation
- Encoding
- Execution
- Encryption
- Logging
General Requirements

- Cross-Platform
- Light weight
- Easy to setup and use
- Thread / Memory safe
- Not a memory management solution
Cross-Platform Testing

- Windows / Unix
- Compilers
  - Visual Studio 2008 / 2010
  - GCC
  - Intel ICC
- Unit testing
Light weight

- Few Dependencies
  - Boost
  - Crypto++
Easy to setup and use

- Documentation
- Few dependencies
- Require as little as possible from the developer
Thread / Memory Safe

- Locking
- Minimal use of pointers
- Code review
- Assertions (nullptr/0/null?)
- SafeInt class written by David LeBlanc
  - http://safeint.codeplex.com/
Memory Management

- Not a memory management solution
- Consistent with Java Implementation
- Requirement - Not broken
  - Jeff Walton
  - Kevin Wall (Fixed ESAPI Java crypto)
- Wei Dai's Crypto++
  - http://www.cryptopp.com/
Current Project State

- Not production ready
  - Some unfinished components and issues
    - Unicode
    - Reference Implementations

- Need contributors and testers
How to get involved (C++)


- **Google Code**

- **OWASP**

- **Mailing List**
  - [https://lists.owasp.org/mailman/listinfo/owasp-esapi-c++](https://lists.owasp.org/mailman/listinfo/owasp-esapi-c++)
  - [owasp-esapi-c++@lists.owasp.org](mailto:owasp-esapi-c++@lists.owasp.org)
How to get involved (C)

- **Google Code**

- **OWASP**

- **Mailing List**
  - [https://lists.owasp.org/mailman/listinfo/owasp-esapi-c](https://lists.owasp.org/mailman/listinfo/owasp-esapi-c)
  - owasp-esapi-c@lists.owasp.org
ESAPI Project Future

- ESAPI Community
- Pluggable Architecture
  - Just get what you need
- Lots of Documentation!
  - Cheat Sheets / Guides
  - Videos
QUESTIONS?

Dan.Amodio@AspectSecurity.com

QUESTIONS?