Preliminary design of the SAFE platform

Benoît Montagu et. al.

PLOS Workshop 2011-10-23
The SAFE team

André DeHon  Ben Karel  Thomas F. Knight, Jr.
Gregory Malecha  Benoît Montagu  Robin Morisset
Greg Morrisett  Benjamin C. Pierce
Randy Pollack  Sumit Ray  Olin Shivers
Jonathan M. Smith  Gregory Sullivan
and many more...
Common Weaknesses Enumeration: Top 25

<table>
<thead>
<tr>
<th>Rank</th>
<th>Score</th>
<th>ID</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>93.8</td>
<td>CWE-89</td>
<td>Improper Neutralization of Special Elements used in an SQL Command ('SQL injection')</td>
</tr>
<tr>
<td>[2]</td>
<td>83.3</td>
<td>CWE-78</td>
<td>Improper Neutralization of Special Elements used in an OS Command ('OS Command Injection')</td>
</tr>
<tr>
<td>[3]</td>
<td>79.0</td>
<td>CWE-120</td>
<td>Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')</td>
</tr>
<tr>
<td>[4]</td>
<td>77.7</td>
<td>CWE-79</td>
<td>Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')</td>
</tr>
<tr>
<td>[6]</td>
<td>76.8</td>
<td>CWE-862</td>
<td>Missing Authorization</td>
</tr>
<tr>
<td>[7]</td>
<td>75.0</td>
<td>CWE-798</td>
<td>Use of Hard-coded Credentials</td>
</tr>
<tr>
<td>[8]</td>
<td>75.0</td>
<td>CWE-311</td>
<td>Missing Encryption of Sensitive Data</td>
</tr>
<tr>
<td>[9]</td>
<td>74.0</td>
<td>CWE-434</td>
<td>Unrestricted Upload of File with Dangerous Type</td>
</tr>
<tr>
<td>[10]</td>
<td>73.8</td>
<td>CWE-807</td>
<td>Reliance on Untrusted Inputs in a Security Decision</td>
</tr>
<tr>
<td>[11]</td>
<td>73.1</td>
<td>CWE-250</td>
<td>Execution with Unnecessary Privileges</td>
</tr>
<tr>
<td>[12]</td>
<td>70.1</td>
<td>CWE-352</td>
<td>Cross-Site Request Forgery (CSRF)</td>
</tr>
<tr>
<td>[13]</td>
<td>69.3</td>
<td>CWE-22</td>
<td>Improper Limitation of a Pathname to a Restricted Directory ('Path Traversal')</td>
</tr>
<tr>
<td>[14]</td>
<td>68.5</td>
<td>CWE-494</td>
<td>Download of Code Without Integrity Check</td>
</tr>
<tr>
<td>[16]</td>
<td>66.0</td>
<td>CWE-829</td>
<td>Inclusion of Functionality from Untrusted Control Sphere</td>
</tr>
<tr>
<td>[17]</td>
<td>65.5</td>
<td>CWE-732</td>
<td>Incorrect Permission Assignment for Critical Resource</td>
</tr>
<tr>
<td>[18]</td>
<td>64.6</td>
<td>CWE-676</td>
<td>Use of Potentially Dangerous Function</td>
</tr>
<tr>
<td>[19]</td>
<td>64.1</td>
<td>CWE-327</td>
<td>Use of a Broken or Risky Cryptographic Algorithm</td>
</tr>
<tr>
<td>[20]</td>
<td>62.4</td>
<td>CWE-131</td>
<td>Incorrect Calculation of Buffer Size</td>
</tr>
<tr>
<td>[21]</td>
<td>61.5</td>
<td>CWE-307</td>
<td>Improper Restriction of Excessive Authentication Attempts</td>
</tr>
<tr>
<td>[22]</td>
<td>61.1</td>
<td>CWE-001</td>
<td>URL Redirection to Untrusted Site ('Open Redirect')</td>
</tr>
<tr>
<td>[23]</td>
<td>61.0</td>
<td>CWE-134</td>
<td>Uncontrolled Format String</td>
</tr>
<tr>
<td>[24]</td>
<td>60.3</td>
<td>CWE-190</td>
<td>Integer Overflow or Wraparound</td>
</tr>
<tr>
<td>[25]</td>
<td>59.9</td>
<td>CWE-759</td>
<td>Use of a One-Way Hash without a Salt</td>
</tr>
</tbody>
</table>

http://cwe.mitre.org/top25/index.html#Listing
# Common Weaknesses Enumeration: Top 25

<table>
<thead>
<tr>
<th>Rank</th>
<th>Score</th>
<th>ID</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>[1]</td>
<td>93.8</td>
<td>CWE-89</td>
<td>Improper Neutralization of surrogate or untrusted data by using Command (Command Injection)</td>
</tr>
<tr>
<td>[2]</td>
<td>83.3</td>
<td>CWE-78</td>
<td>Improper Neutralization of surrogate or untrusted data by Using Command (Command Injection)</td>
</tr>
<tr>
<td>[3]</td>
<td>79.0</td>
<td>CWE-120</td>
<td>Buffer Copy with Different Endness (Cross-Site Scripting)</td>
</tr>
<tr>
<td>[4]</td>
<td>77.7</td>
<td>CWE-79</td>
<td>Improper Neutralization of surrogate or untrusted data by Using Command (Command Injection)</td>
</tr>
<tr>
<td>[6]</td>
<td>76.8</td>
<td>CWE-862</td>
<td>Missing Authorization for a Critical Function</td>
</tr>
<tr>
<td>[7]</td>
<td>75.0</td>
<td>CWE-798</td>
<td>Use of Hard-Coded Cryptographic Algorithm</td>
</tr>
<tr>
<td>[8]</td>
<td>75.0</td>
<td>CWE-311</td>
<td>Missing Authorization for a Critical Function</td>
</tr>
<tr>
<td>[9]</td>
<td>74.0</td>
<td>CWE-434</td>
<td>Unrestricted Access to Secure Function</td>
</tr>
<tr>
<td>[10]</td>
<td>73.8</td>
<td>CWE-807</td>
<td>Reliance on Out of Band (OOB) Communications for Security Checks</td>
</tr>
<tr>
<td>[11]</td>
<td>73.1</td>
<td>CWE-250</td>
<td>Elevation of Privilege or Denial of Service</td>
</tr>
<tr>
<td>[12]</td>
<td>70.1</td>
<td>CWE-352</td>
<td>Use of Hard-Coded Cryptographic Algorithm</td>
</tr>
<tr>
<td>[13]</td>
<td>69.3</td>
<td>CWE-22</td>
<td>Improper Restriction of Malicious Content</td>
</tr>
<tr>
<td>[14]</td>
<td>68.5</td>
<td>CWE-490</td>
<td>Improper Restriction of Malicious Content</td>
</tr>
<tr>
<td>[15]</td>
<td>67.8</td>
<td>CWE-808</td>
<td>Improper Restriction of Malicious Content</td>
</tr>
<tr>
<td>[16]</td>
<td>66.0</td>
<td>CWE-872</td>
<td>Improper Restriction of Malicious Content</td>
</tr>
<tr>
<td>[17]</td>
<td>65.5</td>
<td>CWE-763</td>
<td>Missing Authorization for a Critical Function</td>
</tr>
<tr>
<td>[18]</td>
<td>64.6</td>
<td>CWE-797</td>
<td>Use of Hard-Coded Cryptographic Algorithm</td>
</tr>
<tr>
<td>[19]</td>
<td>64.1</td>
<td>CWE-329</td>
<td>Use of Hard-Coded Cryptographic Algorithm</td>
</tr>
<tr>
<td>[21]</td>
<td>61.5</td>
<td>CWE-710</td>
<td>Use of Hard-Coded Cryptographic Algorithm</td>
</tr>
<tr>
<td>[22]</td>
<td>61.1</td>
<td>CWE-706</td>
<td>Use of Hard-Coded Cryptographic Algorithm</td>
</tr>
<tr>
<td>[23]</td>
<td>61.0</td>
<td>CWE-182</td>
<td>Use of Hard-Coded Cryptographic Algorithm</td>
</tr>
<tr>
<td>[24]</td>
<td>60.3</td>
<td>CWE-190</td>
<td>Integer Overflow or Wraparound</td>
</tr>
<tr>
<td>[25]</td>
<td>59.9</td>
<td>CWE-759</td>
<td>Use of a One-Way Hash without a Salt</td>
</tr>
</tbody>
</table>

http://cwe.mitre.org/top25/index.html#Listing
The SAFE project

- Memory safety
- Access control
- Information flow
The SAFE project

<table>
<thead>
<tr>
<th>PL</th>
<th>Memory safety</th>
<th>Access control</th>
<th>Information flow</th>
<th>Verification</th>
</tr>
</thead>
<tbody>
<tr>
<td>OS</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>HW</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Clean slate design + Formal methods

Occasion to try new abstractions

Precise specifications
Global guarantees
Machine checked proofs
Clean slate design improves co-design Facilitates Formal methods.

Occasion to try new abstractions
Simpler, cleaner design
Fully correct wrt. specs

Precise specifications
Global guarantees
Machine checked proofs
Clean slate design

improves

Formal methods

facilitates

co-design

Occasion to try new abstractions
Simpler, cleaner design
Fully correct wrt. specs

Precise specifications
Global guarantees
Machine checked proofs

It is feasible

- Examples: CompCert compiler, seL4.verified
**Userware**
(written in Breeze)

**Concreteware**
(written in Tempest; formally verified)

**Hardware**
(written in BlueSpec)

---

**System Services**
(device drivers, persistent storage, networking, ...)

**User Programs**

---

**TMU manager**

**Scheduler**

**IPC**

**Memory manager / GC**

---

**TMU rule cache**

**SAFE Processor**

**Stock TPM**
Type/memory safety

**Hardware**
- **TMU**
- **rule cache**
- **SAFE Processor**

**Concreteware**
- **TMU manager**
- **Scheduler**
- **IPC**
- **Memory manager / GC**

**Userware**
- **System Services**
  - (device drivers, persistent storage, networking, ...)
- **User Programs**

**User Programs**

**User Programs**
Type/memory safety

Userware

Concreteware
- TMU manager
- Scheduler
- IPC
- Memory manager / GC

Hardware
- TMU
- rule cache
- SAFE Processor

Mostly functional language
Automatic memory management
Dynamically typed
Contracts

Mostly functional language
Automatic memory management
Dynamically typed
Contracts

Userware

Hardware

Concreteware
Type/memory safety

**Userware**
- Mostly functional language
- Automatic memory management
- Dynamically typed
- Contracts

**Concreteware**
- TMU manager
- Global GC and allocator (Verified)

**Hardware**
- TMU
- rule cache

**Memory manager / GC**

**System Services**
(device drivers, persistent storage, networking, ...)

**User Programs**

**SAFE Processor**

**Hardware**

**Concreteware**

**Userware**

**Type/memory safety**

Mostly functional language
- Automatic memory management
- Dynamically typed
- Contracts

Global GC and allocator (Verified)

System Services (device drivers, persistent storage, networking, ...)

User Programs

SAFE Processor
Type/memory safety

**Userware**
- Mostly functional language
- Automatic memory management
- Dynamically typed
- Contracts

**Concreteware**
- Type tags
- “Fat” pointers

**Hardware**
- Global GC and allocator (Verified)
- Type tags
- “Fat” pointers

**Memory manager / GC**
- Global GC and allocator (Verified)

- **TMU manager**
- **SAFE Processor**
- **rule cache**

**Hardware**
- **TMU**
- **rule cache**
Information flow tracking

**Userware**

- System Services
  - (device drivers, persistent storage, networking, ...)
- User Programs

**Concreteware**

- TMU manager
- Scheduler
- IPC
- Memory manager / GC

**Hardware**

- TMU
  - rule cache
- SAFE Processor

**Platform**

- Hardware
  - Concreteware
  - Userware
Information flow tracking

Userware

Dynamic analysis
- Jif-style labels (DLM)
- **classify** 314159 to secretForBOB

Concreteware

TMU manager
Scheduler
IPC

Memory manager / GC

Hardware

TMU
rule cache
SAFE Processor
Information flow tracking

Userware

Concreteware

Hardware

- Dynamic analysis
- Jif-style labels (DLM)
- **classify** 314159 to secretForBOB
- Challenging!

- TMU manager
- Scheduler
- IPC
- Memory manager

- TMU rule cache
- SAFE Processor

User Programs

System Services (device drivers, persistent storage, networking, ...)

Userware (written in Breeze)

Concreteware (written in Tempest; formally verified)

SAFE Processor

TMU

rule cache

Memory manager / GC

Stock TPM

TMU manager

Userware

Hardware

Concreteware

System Services (device drivers, persistent storage, networking, ...)

Userware (written in Breeze)

Concreteware (written in Tempest; formally verified)
Information flow tracking

Userware

Concreteware

Hardware

- Dynamic analysis
- Jif-style labels (DLM)
- classify 314159 to secretForBOB

- IF labels = HW tags

SAFE 
Processor
TMU
rule cache
Hardware
(written in BlueSpec)
Memory manager / GC
Stock 
TPM
TMU 
manager ...
System Services
(device drivers, persistent 
storage, networking, ...)
Userware 
(written in Breeze)
User 
Programs

Challenging!
 Least privilege discipline

**Userware**
- System Services
  (device drivers, persistent storage, networking, ...)
- User Programs

**Concreteware**
- TMU manager
- Scheduler
- IPC
- Memory manager / GC

**Hardware**
- TMU rule cache
- SAFE Processor
Least privilege discipline

- Authorities/compartments:
  - (Bob, bobAccess) = newPrin;
  - raiseAuth bobAccess;
  - setAuth emptyAuth;

**Userware**

**Concreteware**

- TMU manager
- Scheduler
- IPC

**Hardware**

- TMU
- rule cache
- SAFE Processor

**Memory manager / GC**
Least privilege discipline

 Authorities/compartments:
  - (Bob, bobAccess) = newPrin;
  - raiseAuth bobAccess;
  - setAuth emptyAuth;

 No kernel mode
 ZKOS

 Hardware

 Userware

 Memory manager / GC

 Scheduler

 TMU manager

 IPC
Least privilege discipline

- Authorities/compartments: (Bob, bobAccess) = newPrin;
- raiseAuth bobAccess;
- setAuth emptyAuth;

- No kernel mode
- ZKOS

- Authority checking
- Gates = auth-closures

- Userware
  - Hardware (written in BlueSpec)
  - Memory manager / GC
  - Stock TPM
  - TMU manager
  - IPC
  - Scheduler
  - System Services (device drivers, persistent storage, networking, ...)
  - Userware (written in Breeze)
  - User Programs

- SAFE Processor
  - TMU
  - rule cache
  - File cache
  - Safe
  - Processor

- Concreteware (written in Tempest; formally verified)

- Authorities:
  - (Bob, bobAccess) = newPrin;
  - raiseAuth bobAccess;
  - setAuth emptyAuth;

- No kernel mode
- ZKOS
SAFE Hardware: TMU

Diagram showing the flow of data through the TMU, including the PC, I-Store, Register File, ALU, and Memory.
SAFE Hardware: TMU

Diagram showing the flow of data and control signals through the TMU hardware components: PC, I-Store, Register File, Memory, ALU, and TMU. The diagram includes tags, result tag, new PC tag, security violation, and Combine Tags.
SAFE Hardware: TMU

**Userware**
- System Services (device drivers, persistent storage, networking, ...)
- User Programs

**Concreteware**
- TMU manager
- Scheduler
- IPC
- Memory manager / GC

**Hardware**
- TMU
- rule cache
- SAFE Processor

Userware

Concreteware

Hardware
SAFE Hardware: TMU

- **Userware**
  - System Services (device drivers, persistent storage, networking, ...)
  - User Programs

- **Concreteware**
  - TMU manager
  - Scheduler
  - IPC
  - Memory manager / GC

- **Hardware**
  - TMU
  - SAFE Processor
  - rule cache
SAFE Hardware: TMU

- **TMU**
  - **rule cache**
- **Software rules**
- **System Services**
  - (device drivers, persistent storage, networking, ...)
- **User Programs**
- **Memory manager / GC**
- **Scheduler**
- **IPC**
- **Userware**
- **Concreteware**
- **Authority**
- **Tags**
- **Hardware**
- **SAFE Processor**

**isAuthorized?**
**New tags**

(Elements written in BlueSpec, formally verified using Tempest; User programs written in Breeze.)
SAFE Hardware: TMU

**Cache system**

**Software rules**

**TMU manager**

**System Services** (device drivers, persistent storage, networking, ...)

**User Programs**

**Userware**

**Concreteware** (written in Tempest; formally verified)

**Hardware**

**SAFE Processor**

**User Programs**

**Memory manager / GC**

**Scheduler**

**IPC**

**Authority Tags**

**New tags**

**isAuthorized?**

**Query**

**Update**

**Query**

**TMU rule cache**

**Query**

**Update**
Verification

Userware

System Services
(device drivers, persistent storage, networking, ...)

User Programs

Concreteware

TMU manager

Scheduler

IPC

Memory manager / GC

Hardware

TMU

rule cache

SAFE Processor
Verification

Breeze
non-interference

Concreteware

System Services
(device drivers, persistent storage, networking, ...)

User Programs

TMU manager
Scheduler
IPC

Memory manager / GC

Hardware

TMU
rule cache

SAFE
Processor

User Programs

System Services
(device drivers, persistent storage, networking, ...)

User Programs

System Services
(device drivers, persistent storage, networking, ...)

User Programs
Verification

**Breeze**
- non-interference

**Concreteware**
- TMU manager
- Scheduler
- IPC
- Memory manager / GC

**ISA**
- non-interference

**System Services**
- (device drivers, persistent storage, networking, ...)

**User Programs**

**SAFE Processor**
- TMU
- rule cache
- Stock TPM
- TMU manager
- System Services
- User Programs
- Memory manager / GC
- Scheduler
- IPC
Verification

- **Breeze**
  - non-interference

- **ISA**
  - non-interference

- **Compiler**

- **System Services**
  - (device drivers, persistent storage, networking, ...)

- **User Programs**

- **TMU manager**

- **Scheduler**

- **IPC**

- **Memory manager / GC**

- **TMU**
  - rule cache

- **SAFE Processor**

- **Concreteware**
  - (written in Tempest; formally verified)

- **User Programs**
  - (written in Breeze)
Verification

- **Breeze**
  - non-interference

- **System Services**
  - (device drivers, persistent storage, networking, ...)

- **User Programs**

- **Correctness**

- **Compiler**

- **ISA**
  - non-interference

- **TMU**
  - rule cache

- **SAFE Processor**

**System Components**:
- Hardware (written in BlueSpec)
- Memory manager / GC
- Stock TPM
- TMU manager
- IPC
- Scheduler
- System Services (device drivers, persistent storage, networking, ...)
- User Programs

**User Components**:
- Userware (written in Breeze)
- User Programs
- Correctness

**Formal Verification**
- Breeze non-interference
- ISA non-interference
Verification

Breeze
non-interference

ISA
non-interference

System Services
(device drivers, persistent storage, networking, ...)

User Programs

Correctness

Compiler

Proof

Tests

TMU
rule cache

SAFE Processor

Compiler

Concreteware
(written in Tempest; formally verified)

User Programs

Userware
(written in Breeze)
Current state of SAFE

Userware

System Services
(device drivers, persistent storage, networking, ...)

User Programs

Concreteware

TMU manager

Scheduler

IPC

Memory manager / GC

Hardware

TMU

rule cache

SAFE Processor
Current state of SAFE

**Userware**
- Interpreter
- Standard library + examples
- Non-interference proof (verified)

**Concreteware**
- TMU manager
- Scheduler
- IPC

**Hardware**
- TMU rule cache
- SAFE Processor

Memory manager / GC
Current state of SAFE

**Userware**
- Interpreter
- Standard library + examples
- Non-interference proof (verified)

**Concreteware**
- Detailed design
- Correctness proof started (on simpler version)

**Hardware**
- SAFE Processor
- TMU
- rule cache

**Memory manager / GC**
Current state of SAFE

**Userware**
- Interpreter
- Standard library + examples
- Non-interference proof (verified)

**Concreteware**
- Detailed design
- Correctness proof started (on simpler version)
- Prototype allocator

**Hardware**
- TMU
- Rule cache
- SAFE Processor

- Stock
- Memory manager / GC
- System Services (device drivers, persistent storage, networking, ...)
- Hardware (written in BlueSpec)
- Userware (written in Breeze)
- User Programs

- Interpreter
- Standard library + examples
- Non-interference proof (verified)

- Detailed design
- Correctness proof started (on simpler version)
- Prototype allocator

- TMU
- Rule cache
- SAFE Processor
Current state of SAFE

**Userware**
- Interpreter
- Standard library + examples
- Non-interference proof (verified)

**Concreteware**
- Detailed design
- Correctness proof started (on simpler version)
- Prototype allocator

**Hardware**
- It runs!
- Ongoing tests
Challenges:

Evaluation:
- Pertinence of our choices?

Least privilege:
- Formal metric about robustness?

Information flow:
- Issues with declassification, concurrency...

Large scale verification:
- No precedent!