Design Space Exploration for Security

Eunsuk Kang
UC Berkeley

IEEE SecDev
Nov 3, 2016
Explore and evaluate the potential impact of design alternatives **before** committing to a decision.
Product Engineering

Automotive design, aeronautics, structural engineering
Impact of design parameters on performance, reliability, safety...
Chemical Space Exploration

Drug discovery & material design
Computer System Design

Embedded systems, software/hardware co-design
Compiler optimizations
Software product lines
Designing Travel Itinerary

Goal: Get the One Ring to Mordor

Map of Middle-earth
Designing Travel Itinerary

Goal: Get the One Ring to

Option 1: Blizzard

Option 2: Bad dude

Option 3: Another bad dude

Option 4: Too far

Mordor
Deciding the Fate of the Nation

Evaluating & comparing the alternatives
What about Security?
Developers already perform DSE!

Why is passing the session id as url parameter insecure?

I recently followed a discussion, where one person was stating that passing the session id as url parameter is insecure and that cookies should be used instead. The other person said the opposite and argued that Paypal, for example, is passing the session id as url parameter because of security reasons.

Is passing the session id as url parameter really insecure? Why are cookies more secure? What possibilities does an attacker have for both options (cookies and url parameter)?
Designing single-sign-on with JSONP/CORS?

What is the difference between localStorage, sessionStorage, session and cookies?

Demystifying Web Authentication (Stateless Session Cookies)

Why Does OAuth v2 Have Both Access and Refresh Tokens?

Session Authentication vs Token Authentication

Why is passing the session id as url parameter insecure?

Is either GET or POST more secure than the other?

When comparing an HTTP GET to an HTTP POST, what are the differences from a security perspective? Is one of the choices inherently more secure than the other? If so, why?

I realize that POST doesn't expose information on the URL, but is there any real value in that or is it just security through obscurity? Is there ever a reason that I should prefer POST when security is a concern?

Edit:
Over HTTPS, POST data is encoded, but could URLs be sniffed by a 3rd party? Additionally, I am dealing with JSP; when using JSP or a similar framework, would it be fair to say the best practice is to avoid placing sensitive data in the POST or GET altogether and using server side code to handle.
Hypothesis

Many developers don’t know what alternatives are available (but they pick one anyway, often arbitrarily)
Hypothesis

Many developers don’t know what alternatives are available (but they pick one anyway, often arbitrarily)

Many developers lack domain expertise required to reason about the impact of their decisions on security
Hypothesis

Many developers don’t know what alternatives are available (but they pick one anyway, often arbitrarily)

Many developers lack domain expertise required to reason about the impact of their decisions on security

Lots of knowledge available (guidelines, CVEs, StackOverflow), but relatively few *design* methods/tools to leverage them
Many developers don’t know what alternatives are available (but they pick one anyway, often arbitrarily)

Many developers lack domain expertise required to reason about the impact of their decisions on security

Lots of knowledge available (guidelines, CVEs, StackOverflow), but relatively few design methods/tools to leverage them

Can we build a framework to establish & support DSE as its own activity in a secure development process?
Elements of a DSE Framework

- System Description
- Evaluation Metric
- Constraints

Exploration Engine

Design Candidate 1
Design Candidate 2
...
Design Candidate N
Elements of a DSE Framework

System Description
Partial/incomplete, with one or more design parameters

e.g.
Description: An architectural model of a shopping cart application
Parameter: Choice of session encoding (cookie vs URL param)
Elements of a DSE Framework

Design Candidate
A complete description with specific parameter values

e.g.
Candidate 1: The cart design with sessions as cookies
Candidate 2: The cart design with sessions as URL param
Elements of a DSE Framework

Evaluation Metric
A function that maps each candidate to some value e.g. 

**Boolean** function: *Yes* if given policy satisfied, *no* if not

“Only the owner of a shopping cart should be able to access its content”
Elements of a DSE Framework

**Evaluation Metric**
A function that maps each candidate to some value
e.g.
More complex function: **A list of potential vulnerabilities**
“Candidate 1 may be vulnerable to CSRF attack”
“Candidate 2 may be vulnerable to session fixation”
Elements of a DSE Framework

- System Description
- Evaluation Metric
- Constraints

Exploration Engine

- Design Candidate 1
- Design Candidate 2
- Design Candidate N

**Evaluation Metric**
A function that maps each candidate to some value
e.g. **Non-security** metrics: Performance, usability, etc.
Combination of both security & non-security metrics
Elements of a DSE Framework

- System Description
- Evaluation Metric
- Constraints

Exploration Engine

- Design Candidate 1
- Design Candidate 2
- Design Candidate N

**Constraints**
Rules out certain design candidates as invalid

*E.g.*
“Only consider app designs that satisfy the given policy”
“Only consider candidates that use HTTPS for all requests”
Elements of a DSE Framework

Exploration Engine
- Enumerates candidates & computes metrics for each
- Provides candidate ranking/trade-off analysis
- May leverage simulation, verification, data-driven analysis...
An Instantiation of DSE Framework
Protocol Implementations

Vulnerable implementations from secure protocol spec!
Platform-specific details absent from the spec
Certain security issues only visible once implemented
e.g., OAuth, OpenID [Sun2012, Wang2012]
Proactively discover decisions that may introduce vulnerabilities into a protocol implementation
Decision as a mapping from abstract to concrete operations
Domain KB for evaluating candidates against security policy
(To appear in FSE 2016)
Mapping as Design Decisions

<table>
<thead>
<tr>
<th>AddItem</th>
<th>HttpReq</th>
</tr>
</thead>
<tbody>
<tr>
<td>itemID</td>
<td>choc</td>
</tr>
<tr>
<td>token</td>
<td>j6oAOxC</td>
</tr>
<tr>
<td>method</td>
<td>POST</td>
</tr>
<tr>
<td>url</td>
<td><a href="http://www.mystore.com/addItem">http://www.mystore.com/addItem</a></td>
</tr>
<tr>
<td>headers</td>
<td>??</td>
</tr>
<tr>
<td>body</td>
<td>??</td>
</tr>
<tr>
<td>response</td>
<td>??</td>
</tr>
</tbody>
</table>

Which URL do I designate for this operation?  
Which HTTP method do I use?  
How do I transmit the item ID & token?

**Key idea:**
Specify mapping partially using symbolic constraints  
Use an algorithmic engine to explore candidate mappings
Experiments

Domain model of web platform components
HTTP server, browser behavior, SOP, web attacker...
Generic, reusable across multiple web protocols

Verification of protocol implementations
OAuth, IFTTT, HandMe.In
Modeling and verification in Alloy
Detected new & known attacks
Other Potential Applications

**Policy Design and Analysis**
Social network privacy settings (Margrave)
Access control policies (XACML)
Health record systems (Cassandra, PATRN)

**Secure API Design**
Key management, PIN processing, EMV cards
Automated analysis & testing (Focardi, Steel, Bond)

**Network Design and Administration**
Firewall configurations (ConfigChecker, Vantage...)
SDN (Flowlog, Frenetic, Veriflow...)
Related Work

Verification & Static Analysis
Often ex post facto, on a complete system artifact
Yes/no answer, less emphasis on exploration
Theories & techniques still applicable to DSE!
Related Work

**Verification & Static Analysis**
Often ex post facto, on a complete system artifact
Yes/no answer, less emphasis on exploration
Theories & techniques still applicable to DSE!

**Threat Modeling & Risk Analysis**
Like DSE, best carried out in early development stage
Emphasis on the threat space
Relatively little tool support
(Microsoft SDL, TRIKE, ThreatModeler)
Related Work

Verification & Static Analysis
Often ex post facto, on a complete system artifact
Yes/no answer, less emphasis on exploration
Theories & techniques still applicable to DSE!

Threat Modeling & Risk Analysis
Like DSE, best carried out in early development stage
Emphasis on the threat space
Relatively little tool support
(Microsoft SDL, TRIKE, ThreatModeler)

Leveraging Security Knowledge
Lots collected over the years: CVE, CWE, CAPEC,...
Not easily mechanizable
Some examples: MulVal, attack graphs, COPS
Several recent efforts on formalizing the web
[Akhawe10, Bansal12, Fett14]
Conclusions

Support DSE as its own development activity!
We all do it anyway (in our head), so might as well do it right
Many useful theories & techniques already available
But also many challenges (see paper)
Conclusions

Support DSE as its own development activity!
We all do it anyway (in our head), so might as well do it right
Many useful theories & techniques already available
But also many challenges (see paper)

Will DSE naturally result in secure systems?
No! Still need other security practices
Frameworks, guidelines, testing, program analysis, etc.
But thinking about security early helps!
Conclusions

**Support DSE as its own development activity!**
We all do it anyway (in our head), so might as well do it right
Many useful theories & techniques already available
But also many challenges (see paper)

**Will DSE naturally result in secure systems?**
No! Still need other security practices
Frameworks, guidelines, testing, program analysis, etc.
But thinking about security early helps!

“Plans are worthless, but planning is everything.”

- Dwight D. Eisenhower