Today’s Menu

What Every Computer Scientist Should Know about Security

GuardRails
Jonathan Burket (BACS 2)
Patrick Mutchler (BSCS 4)
Michael Weaver (BSCS 4)
Muzzamil Zaveri (BACS 4)

Efficient Secure Computation
Yan Huang (CS PhD)
Yikan Chen (CPE PhD)
Jerry Ye (BSCS 3)
Samee Zahur (CS PhD)

I’m looking for new students for the summer for both projects (and other ideas)!

What Every Human Should Know About Security

“Many children are taught never to talk to strangers, an extreme precaution with minimal security benefit.”

“Emma Lion loves to make new friends, but Mama tells her to be careful and never talk to strangers. Emma sees new people to meet everywhere she goes. How will she know who is a stranger?”

Security

Data Evans
www.cs.virginia.edu/evans
cs2190
3 March 2011

Image: Roger Halbeir
Security

- Technical questions
  - Figuring out who is not a “stranger” (authentication)
  - Controlling access to resources (protection and authorization)
- Value judgments
  - Managing risk vs. benefit (policy)
- Deterrents
  - If you get caught, bad things happen to you

Protecting assets from misuse

Computer Security

Study of computing systems in the presence of adversaries

Quiz

Authentication, Protection, Authorization, Policy, or Deterrent?

Authentication, Protection, Authorization, Policy, or Deterrent?
Web Security is Annoying and Tedious

Access Control

```ruby
if include_subprojects && !active_children.empty?
  ids = [id] + active_children.collect {|c| c.id}
  conditions = ['#{Project.table_name}.id IN (#{ids.join(',')})']
  AND #{Project.visible_by}
end
```

(Example from Redmine project management tool)

Input Validation

```
"User: <a href='profile_page'>" + user_name + "</a>"
My user_name is "<script language='javascript'>doEvil();</script>"
```

(Cross-site scripting)

Example Policies

<table>
<thead>
<tr>
<th>Annotation</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>@delete, :admin, :to login</td>
<td>Only administrators can delete this object</td>
</tr>
<tr>
<td>@edit, pswrd, self.id == user.id, :to login</td>
<td>Only the user may change that user's password</td>
</tr>
<tr>
<td>@create, User, log_create; true</td>
<td>Whenever a User object is created, write to log</td>
</tr>
</tbody>
</table>

Policies are attached to classes or individual fields. Can perform arbitrary checking and actions based on read, edit, append, create, destroy events.

Enforcing Policies

GuardRails Proxy

GuardRails Alias

Protected Object

Protected Field

Taint Tracking

```
“foo” + “bar” => “foobär”
```

URL Parameters

Form Data

Other User Input

Controller

View

Tainted HTML

Safe HTML

Context-Sensitive Sanitization

Taint Status

Database

Model

Data

Taint Status
Possible Projects

Automating Annotations
Client/Server Side Integration
Evaluation

Secure Computation in the Real(ish) World

Yan Huang (CS PhD)
Yikan Chen (CPE PhD)
Jerry Ye (BCS 3)
Samee Zahur (CS PhD)

“Genetic Dating”

Bob
Alice

Genome Compatibility Protocol

WARNING! Don’t Reproduce

1990: Human Genome Project starts, estimate $3B to sequence one genome ($0.50/base)

2000: Human Genome Project declared success, cost $300M

Cost Per Base of DNA Sequencing and Synthesis


<table>
<thead>
<tr>
<th>Year</th>
<th>reference</th>
<th>Technology</th>
<th>Sample</th>
<th>Average Reported Coverage depth (fold)</th>
<th>Reported sequencing consumables cost</th>
<th>Estimated cost per 40-fold coverage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1990</td>
<td>Sanger (ABI)</td>
<td>XV</td>
<td>7</td>
<td>$10,000,000</td>
<td>$57,000,000</td>
<td></td>
</tr>
<tr>
<td>1995</td>
<td>Roche454</td>
<td>D7</td>
<td>7</td>
<td>$1,000,000</td>
<td>$5,700,000</td>
<td></td>
</tr>
<tr>
<td>1995</td>
<td>Illumina</td>
<td>NA12857</td>
<td>30</td>
<td>$250,000</td>
<td>$330,000</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>Helicos</td>
<td>SRQ</td>
<td>28</td>
<td>$48,000</td>
<td>$69,000</td>
<td></td>
</tr>
<tr>
<td>2000</td>
<td>Illumina</td>
<td>NA07022</td>
<td>87</td>
<td>$6,005</td>
<td>$3,700</td>
<td></td>
</tr>
<tr>
<td>2009</td>
<td>this work</td>
<td>this work</td>
<td>NA19240</td>
<td>63</td>
<td>$3,491</td>
<td>$2,200</td>
</tr>
<tr>
<td>2009</td>
<td>this work</td>
<td>this work</td>
<td>NA20431</td>
<td>45</td>
<td>$1,726</td>
<td>$1,500</td>
</tr>
</tbody>
</table>

Secure Two-Party Computation

Bob's Genome: ACTG...
Markers (~1000): [A, T, ...]

Alice's Genome: ACTG...
Markers (~1000): [A, G, ...]

Can Alice and Bob compute a function of their private data, without exposing anything about their data (other than the result)?

Secure Function Evaluation

Alice (circuit generator)  Bob (circuit evaluator)

Agree on $f(a, b) \rightarrow x$

Outputs $f(a, b)$ without revealing $a$ to Bob or $b$ to Alice.

Andrew Yao, 1982/1986

Computing with Lookup Tables

<table>
<thead>
<tr>
<th>Input 0</th>
<th>Input 1</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_0$</td>
<td>$b_0$</td>
<td>0</td>
</tr>
<tr>
<td>$a_0$</td>
<td>$b_1$</td>
<td>0</td>
</tr>
<tr>
<td>$a_1$</td>
<td>$b_0$</td>
<td>0</td>
</tr>
<tr>
<td>$a_1$</td>
<td>$b_1$</td>
<td>1</td>
</tr>
</tbody>
</table>

AND

$S_{a_0} = S_{a_1}$

Computing with Garbled Tables

<table>
<thead>
<tr>
<th>Input 0</th>
<th>Input 1</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>$a_0$</td>
<td>$b_0$</td>
<td>$Enc_{a_0, b_0}(o_0)$</td>
</tr>
<tr>
<td>$a_0$</td>
<td>$b_1$</td>
<td>$Enc_{a_0, b_1}(o_0)$</td>
</tr>
<tr>
<td>$a_1$</td>
<td>$b_0$</td>
<td>$Enc_{a_1, b_0}(o_0)$</td>
</tr>
<tr>
<td>$a_1$</td>
<td>$b_1$</td>
<td>$Enc_{a_1, b_1}(o_1)$</td>
</tr>
</tbody>
</table>

Garbled Circuit Protocol

Alice (circuit generator)  Bob (circuit evaluator)

Creates random keys: $a_0, a_1, b_0, b_1, o_0, o_1$

Sends $a_i$ to Bob based on her input value

How does the Bob learn his own input wires?
**Primitive: Oblivious Transfer**

Alice

Knows $b_0, b_1$

Learns nothing

Bob

Picks $i \in \{0, 1\}$

Learns $b_i$ (only)

Oblivious: Alice doesn’t learn which secret Bob obtains
Transfer: Bob learns one of Alice’s secrets

Rabin, 1981; Even, Goldreich, and Lempel, 1985; many subsequent papers

**Chaining Garbled Circuits**

![Diagram of garbled circuits]

**Building Secure Computing Systems**

We can do any computation privately this way!

**Cost metric very different from normal circuits**

Generating/evaluating each gate requires several encryption operations

Can only execute each gate once

XOR is free (and NOT) is nearly free

**Framework for Efficiently Executing Circuits**

Pipeline generation and evaluation

**Possible Projects**

Design and implement a secure computation
Fingerprint matching
Genome analysis
Image recognition
Auctions
Improve garbled circuit evaluation
Multi-core, GPU
Stronger Adversary Model

“If you have something that you don’t want anyone to know, maybe you shouldn’t be doing it in the first place.”

Eric Schmidt (then CEO of Google)