Project Ideas

• Semester long projects of medium scope

• TAs presenting project ideas today

• Students can submit their own ideas
  – Send to cs161projectidea@gmail.com
  – To be approved by staff
  – Short presentation of approved ideas this Wed.
Project Groups

• Each group is 6 people, *no exceptions*
  – Can be with lab partner, but doesn’t need to be

• Form your own groups

• Use the discussion forum!
Project Group Submission

• Groups choose top 2 project preferences
  – We’ll try hard to give top preference
  – Multiple groups on same project

• Provide times the group can meet
  – Needs to be many, many times!

• Web submission
Project Signup Schedule

• 1/23 Monday – TA project presentation

• 1/24 Tuesday – Students submit project ideas

• 1/25 Wednesday – Approved ideas presented by students

• 2/1 Wednesday – Group signups due
Web Security

Joel
Content Security Policy for Web Applications

- Content Security Policies (CSP) can be applied to sites to stop XSS
- ...but requires modifying the application
- Modify a large application (e.g. MediaWiki) to use an effective CSP
- Show that the application still works with the policy applied
Privilege Granularity in Chrome Extensions

• Extensions add functionality to web browsers
• Chrome limits privileges to only those requested
  – Coarse grained
• How well does the granularity match actual functionality?
• Evaluate this over several hundred extensions
• Find common patterns in extensions
  – Propose alternative privileges?
More Web Security

Dev
Measuring Incoherencies on the Web Platform

- **Goal**: Write an addon and a crawler to measure the prevalence of same-origin-policy inconsistencies. For example, cross-origin overlap, document.domain usage.

- **Motivation**: Can’t improve what you don’t know. The current situation is a mess.

- **Evaluation**: Number of checks implemented and scale of data collected.

- **Prereqs**: HTML, JavaScript, the Web
Privilege Separation of HTML5 applications

• Goal: Implement privilege separated versions of popular HTML5 applications

• Motivation: TCB Reduction, auditability, SECURITY!

• Evaluation: TCB reduction achieved, functionality reduced, security analysis

• Prereqs: HTML, JavaScript
Implementation of DSI in Firefox

• Goal: Implement a nonce based approach to XSS mitigation
• Motivation: XSS is difficult to protect against purely on the server side. Enlist help from the browser.
• Evaluation: HTMLPurifier test cases passed
• Prereqs: C/C++ knowledge, HTML, JavaScript
Measuring JavaScript Dynamicity

• Goal: Write an addon and a crawler to measure the prevalence of crazy js on the web

• Motivation: JS consists of a number of crazy features that make analysis difficult. A measurement will tell us what we can ignore and what we can’t.

• Evaluation: Number of checks implemented and scale of data collected.

• Prereqs: HTML, JavaScript
Android Security

Steve
Similarity Among Android Applications by GUI Feature Extraction

• **Goals:** Develop a system to compute similarity between GUIs in Android apps
  – Examine both static elements (XML) and dynamic elements (DEX)

• **Motivation:** Piracy, malware detection
  – Similar looking applications with underlying differences in code is a good metric for detecting trojaned applications
  – Copied or stolen interface detection

• **Description:** Feature extraction and comparisons Android GUIs
  – Students will be expected to evaluate their tool against no less than 1000 applications and demonstrate and evaluate their approach

• **Prereq:** Android, Java, C++, machine learning a plus!
Measuring Intent Security Problems in Android

• **Goals:** Develop a tool to detect problems with Android intents and measure their prevalence among a large set of applications. Suggest proposals to fix most common bugs.

• **Motivation:** Intents can leak information or be used to abuse privilege
  – Pressing need to quantify the prevalence of these errors
  – Can shed insight into developing a better Intent system to make Android more secure.

• **Description:** Understand common flaws with the Intent system in android, classify and quantify their prevalence on a large dataset.

• **Prereq:** Android (very experienced!), Java
Android and Testing via Crowd Sourcing

Kevin
Fine-grained permission control engine on Android

• The current coarse-grained permission system:
  – Application-level
  – Install-time decision
  – All-or-nothing decision
• Goal: Fine-grained rule-based permission system
  – (App, Package/Callstack, Permission)
• Outcome:
  – Policy engine
  – Sample rules
Testing via Crowd Sourcing

• HCI-based programs should be tested by a human
  – Event-driven, user-interaction directed
• A first step towards that: describing interactions
  
  - Type “username”
  - Type “pa****rd”
  - Click “Login”
  - Click “CS161”
  - Click “like”
  - ...

• Outcome:
  – Interaction recorder and replayer
An Evaluation of Automated Bug-finding Approaches

Cho
Automated Software Analysis

• Tidal Wave in constraint solving and symbolic execution techniques

• Analysis of software security will be increasingly automated and based on logic

• Different SE approaches
  – “Dynamic” symbolic execution
  – Static checking
  – Model checking

How do they compare?

Source: A. Platzer
What do I need to do?

• Evaluate and compare the best-of-breed tools of the 3 approaches
  – On a common set of real-world applications
  – Focus on security bugs
  – Soundness & Completeness

• [Practical] Determine the kind of programs each approach is well-suited for

• [Research] Gain insights into how they work / apply symbolic execution differently
ACID Test

- Evaluate your own suitability for this project (and your team-mates)
- Google: “KLEE symbolic execution”

**Difficulty:** Was it a breeze?

**Interest:** Does it make you want to learn more?
Privacy

Emil
Enhance Privacy of Open Source Apps

- **Goal:** Combine popular open source applications with UC Berkeley’s platform for private data.
- **Example Apps:** Online document editors, photo galleries, video conferencing, chat rooms, webmail.
- **Why:** Offer rich applications to users with strong privacy guarantees.
Privacy Extension for Browsers

• **Goal:** Prevent a website from sending user data to another website.

• **Example:** Your online tax software should not share your financial data with crooks.

• **How:** Develop a browser extension that intercepts HTTP requests.
Goal: Analyze Google+ data on a global scale.

** We have daily snapshots of the Google+ social graph and profile data. **

Explore and model how social patterns evolve.

Determine importance and weights of traits in social networks.

Why do people accept friend requests?
• **Goal:** Create a single website for submitting applications to multiple graduate schools.

• **Why:** Offer enhanced privacy for students, and letter recommendation writers.
Virtual Machine Forking

• **Goal:** Efficiently isolate web sessions from each other on a server to improve security.
• **Why:** Prevent privacy breaches across users.
• **How:** Fork virtual machine metadata and memory mapping for each user session.
Memory Access Privacy

• **Goal:** Determine what an application is doing by analyzing its memory access pattern.

• **Why:** Demonstrate new form of attack on privacy for outsourced computation.

• **How:** Record and analyze memory traces of applications.
Alternative Authentication

Daniele
Active Authentication based on mouse and keyboard usage

• Goal: write Javascript collection code and Python analysis code to distinguish mouse/keyboard usage patterns
• Motivation: Active authentication aims at strengthening the classic password authentication by observing user behavior
• Evaluation: Robustness and portability of Javascript code. Quality of the analysis (number and uniqueness of extracted features)
• Prereqs: HTML, JavaScript, Python