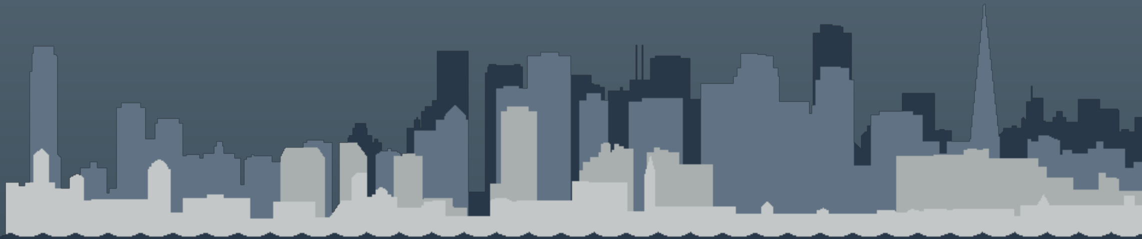
TIZEN™ DEVELOPER CONFERENCE
MAY 7–9, 2012



Breaking Same-Origin for Fun and Profit

Dean Pierce
Intel - OTC Security

Who are you and why are you on stage?

- Intel - Open Source Technology Center
- Infrastructure, User Privacy, Emerging Threats

- Background in Security
- Attending security conferences for 10 years
- Speaking for 7 years
- Defcon, BSides, Toorcon, XCon

- I am not a Tizen App or WRT developer
- I am not an XSS specialist

What's this talk all about then?

- Same Origin Policy (SOP)
    - code from a given domain should only be allowed to touch data from the same domain
    - cornerstone of HTML/JS security model

- Cross Site Scripting (XSS)
    - well known attack surface in exploit dev community
    - vague recollection in web dev community
    - almost completely unknown in app dev community

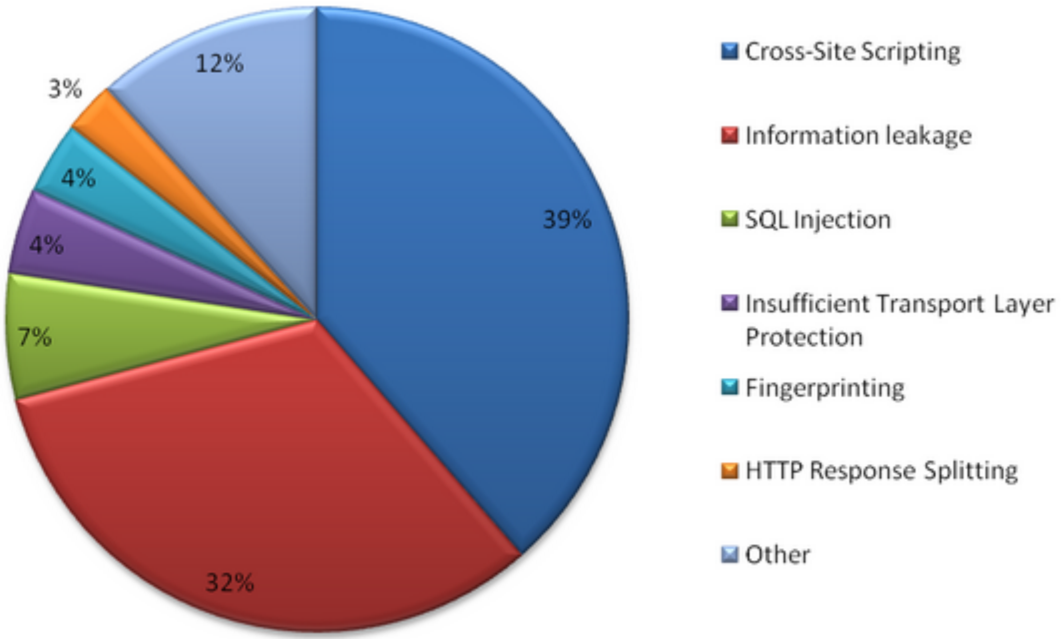
What is XSS?

- Websites often display information based on data provided by a someone who cannot be trusted.

- XSS most commonly occurs when strings are taken from users, and inserted directly into the DOM without properly HTML encoding.

- Unfiltered strings can be used to insert things like <script> tags, allowing javascript to be inserted into the current domain, from a remote location.

What is XSS?



http://projects.webappsec.org/Web-Application-Security-Statistics

Classic Example of XSS

- http://site.com/page.php?id=main
- Welcome to the Main page!

- http://site.com/page.php?id=candybar
- Sorry, the page "candybar" doesn't exist

- http://site.com/page.php?id=<script>alert('Yay!');</script>
- Yay!

What is a Same-Origin "Break"?

- "Same Origin" is a good mantra, but optimistic.
  - <img src="http://bank.com/sendmoney?dest=crime">
  - <img src="skype://8675309;rm -rf /">
  - <iframe src="http://evilbrowserexploit.com/attack>"
  - <iframe src="http:/bank.com/page?id=<script>evil(); ...

- Multiple domains **can** communicate.
- Oftentimes behavior will vary from client to client.
- Javascript gets more powerful every day.

How did this happen?

- HTML was designed for making pretty looking documents.
- Javascript was designed to be a dumbed down Java.
- New features were implemented frequently as various corporations battled over control of the internet.

- The rate that the internet exploded forced browsers to grow up in a hurry, but developers always pushed for more control over the browsers.
- Security solutions, rules, and best practices were only ever created in response to widespread attacks.

The Past

- Website Defacement
- Session Hijacking
- DNS Pinning
- Request Forgery
- XSS Worms (Samy is my hero)
- Automated Browser Exploitation
- Clickjacking
- Javascript Keyloggers

- Client-side issues often ignored

The Future

- A lot of good work has been done by Kyle Osborne.
  - skype, chromeos

- Ring 0 is for suckers, OS exploitation is pointless.

- All the interesting stuff is in the browser.
- More and more powerful web runtimes.
- Escalation done via domain hopping.
- Unexpected javascript is the only rootkit you need.

What does any of this have to do with Tizen?

- Tizen app architecture is based on HTML5/JS
- Secure tooling environments are almost non-existent in HTML5/JS
- We can set the precedent for how secure WRT apps should be developed.

- We can stop bad habits before they start.
- We can take a strong stance on security before it gets out of hand.

What sorts of apps are at risk?

- SMS/Email/RSS/News readers.
- Anything that displays content from remote sources.
- "Remote sources" can even be local API calls.

A Simple SMS Reader

- It started with an amazing blog entry.
    - http://giscaro.wordpress.com/

- I followed along, wrote up the app, and tested it with the event injector.

- My first test SMS : "hey there"
- My second test SMS : "<script>alert('yay!');</script>"

A Simple SMS Reader

- Message taken from message[i].body.plainBody
- String placed straight into the DOM with message_thread. append()

- var clean_string = $('<div/>').text(scary_string_here).html();

Who is to blame?

- XSS is **HARD**
  - subscribe to reddit.com/r/xss

- Developers shouldn't need to be security experts to write secure code.
- It is generally considered best practice to deal with these issues at the Framework level.
- Users should not be tempted to touch the DOM, they should be using javascript widget objects, like JQuery Mobile.
- Some better tooling in the SDK is the earliest place to catch bugs.

So what? It's just Javascript!

- Following the rules
  - Filesystem access
  - Data access
  - App to App
  - System resources

- Attacker gets complete control over the vulnerable domain
- Tizen WRT apps are first class citizens!

So what? It's just Javascript!

- Breaking the rules
  - Webkit is scary! (do not change from webkit!)
  - 59 potentially exploitable bugs disclosed in March alone.
  - There is no update strategy for webkit on Tizen.
  - NaCl storytime (Mark Dowd is a Rockstar)

    2009, 600 people, 22 bugs, 12 from Mark, most in the first few hours
  - WebGL / WebCL

So, what can we do?

- Improve the API
  - maybe return pre-filtered strings from device API?
  - establish recommended widget library

- Improve the SDK
  - SDK should yell if the user starts writing

- Always move forward, but learn from the past.

APPLAUSE

Questions?

APPLAUSE

TIZEN DEVELOPER CONFERENCE MAY 7–9, 2012

Thank You!

contact : dean.e.pierce@intel.com

TIZEN™ DEVELOPER CONFERENCE MAY 7–9, 2012