Crosswalk Security on Tizen 3.0

Xu Zhang(xu.u.zhang@intel.com)
Agenda

• Tizen 3.0 Security Overview
• Crosswalk Overview
• Crosswalk Security Features
• Conclusion
• Contribute to Crosswalk
Tizen 3.0 Security Overview

• **The Objectives:**
  • Protect user data
  • Protect system resources (including the network)
  • Provide application isolation

• **Tizen 3 provide key security features:**
  • Discretionary Access Control (DAC)
  • Mandatory Access Control (MAC) with the kernel LSM SMACK
  • Cynara the permission management and check service
  • Security manager
Crosswalk Share Process Mode for Per User

Browser Process
(shared)

Crosswalk Application One
- Render Process
- Extension Process
  WebAppOne label

Crosswalk Application Two
- Render Process
- Extension Process
  WebAppTwo label

Crosswalk Application N
- Render Process
- Extension Process
  WebAppN label
Crosswalk Security Features

- **Relationship with Tizen 3.0 security**
  - Build over: SMACK, Cynara and security manager

- **Goals of Crosswalk Security:**
  - Keep sensitive resource and data safely
  - Web applications should be run with least privileges
  - Web applications are sandboxed so they can only see their own resources
  - Mitigate a broad class of content injection vulnerabilities, such as cross-site scripting (XSS) attacks

- **Features**
  - API access control
  - Web widget application signing
  - CSP (Content Security Policy)
  - WARP (Widget Access Request Policy)
API Access Control

Launcher

Web Package
  Config.xml

Installer

Web App
  RP
  EP
  BP

Core C/C++ API

Cynara

Policies
API Access Control - Crosswalk Installer

Crosswalk

Web App Package
- config.xml
- html, css, etc.

Crosswalk Installer
1. Signature verification
2. Extract & Filter Permissions
3. Extract & Copy Files

Tizen

Files on disk
/home/$user/.xwalk-service/applications/$appid

Security Manager

SMACK
A. Web App Process Label
B. File Labels

Cynara
Permission Policy
API Access Control - Crosswalk Installer (Cont.)

- **General flow of installing web application**
  1. Crosswalk installer unzip Crosswalk web application package
  2. Crosswalk installer verify signature of the package
  3. Crosswalk extract permissions list from config.xml
  4. According to privilege level, Crosswalk filter invalid permissions out of permissions list
  5. Crosswalk installer calls security-manager to insert policies and set SMACK label to resource files
API Access Control – Launcher

![Diagram of API access control for Launcher]

- **menu-screen**
  - [menu-screen-permissions]

- **launchpad**
  - [launchpad-permissions]

- **another app**
  - [another app-permissions]

- **App1**
  - xwalk-launcher (Extension Process)
  - [App1-permissions]
  - Render Process
  - [App1-permissions]

- **App2**
  - xwalk-launcher (Extension Process)
  - [App2-permissions]
  - Render Process
  - [App2-permissions]

- **AppN**
  - xwalk-launcher (Extension Process)
  - [AppN-permissions]
  - Render Process
  - [AppN-permissions]

- **xwalk browser process**
  - [xwalk-daemon-permissions]

**Legend:**
- toplevel session child process: [permissions applied by session]
- launchpad child process: [permissions applied by launchpad]
- xwalk child process: [permissions applied by browser process]
API Access Control – Launcher (Cont.)

• **General flow of launching web application**
  1. User trying to launch an application by clicking the application icon in home screen or another application throwing an intent to launch a web app.
  2. The launchpad_daemon fork a child process xwalk-launcher and xwalk-launcher create a new xwalk extension process instance
  3. xwalk-launcher send an dbus message to running application manager (Browser process). Application manager launch web application and start a new render view (render process)
API Access Control – Runtime Check

Crosswalk Application

Crosswalk Browser Process Label

Browser Process (shared)

Privileged Operations
1. Check Cynara privileges for WebApp
2. If OK, allow access

Render Process
Extension Process

WebApp label

Cynara

Tizen System Services
• **General flow of checking API access in browser process**

1. When sensitive W3C JS API is invoked, render process send IPC to browser process
2. Browser process request Cynara to check API permission
3. Cynara return ALLOW/DENY to browser process
4. If the operation is allowed, browser process access Tizen system service
Application Signing

- **Tizen application must be signed with 2 signatures:**
  - Author signature
  - Distributor signature
- **Based on W3C recommendations for XML digital signature**
- **Crosswalk verifies that the application has been properly signed with the certificate**
- **Decide privilege level of web application**
  - Platform
  - Partner
  - Public
  - Untrusted
Content Security Policy (CSP)

- CSP works as a whitelisting mechanism for resources loaded or executed by web applications
- Policy applies to a wide variety of resources
- The policy is defined via the application’s manifest as follows

```json
{
    ..., 
    <tizen:content-security-policy>script-src 'self'</tizen:content-security-policy>
    ... 
}
```

- CSP support in Crosswalks is based on Chromium and Blink implementation
**Widget Access Request Policy (WARP)**

- All network accesses by widgets are denied by default
- A widget must declare in its manifest which network resources it will access (such as XMLHttpRequest, iframe, img, script, etc.)
- `<access>` element in config.xml. Developers can specify protocols, domains, and sub-domains.

```xml
<widget xmlns="http://www.w3.org/ns/widgets">
  ...
  ...
  <access origin="https://example.net" subdomains="true" />
  <access origin="http://example.com" subdomains="false" />
  ...
</widget>
```
Conclusion

• To developers:
  • You need to declare the required permissions in the manifest
  • Declare the minimum set of permissions you really need
  • Pay attention to proper error handling in your application
Acknowledgements

• Thanks the contribution from Terri Oda, Casey Schaufler, Xinchao He, Yongkang You and Peter Wang
Contribute to Crosswalk

- **Project website:**
  - [https://crosswalk-project.org](https://crosswalk-project.org)
- **Mail list:**
  - [crosswalk-help](https://crosswalk-help)
  - [crosswalk-dev](https://crosswalk-dev)
- **IRC**
  - #crosswalk at irc://freenode.net
- **Bug Track:**
  - [https://crosswalk-project.org/jira/](https://crosswalk-project.org/jira/)
Question?