Protecting Browsers From Extension Vulnerabilities

Adam Barth, Adrienne Porter Felt, Prateek Saxena, Aaron Boodman

Browser Extensions

- 1/3 of Firefox users run at least 1 extension
- Extensions are *not* the same as plugins

gTranslate

Benign-but-Buggy

- **Extensions are not written by security experts**
- Extensions interact extensively with web sites
- Firefox extensions run with the browser's full privileges
- An attacker can usurp a vulnerable extension's broad privileges

Example Attack

- Liverani and Freeman, “Abusing Firefox Extensions”
  - Cool Previews 2.7 accepted URIs without any filtering
  - data: URI’s contents are rendered with privileges
  - Malicious URI leads to remote code execution

```
<script>

var getWorkingDir= Components.classes["@mozilla.org/file/directory_service;1"].
getService(Components.interfaces.nsIProperties).get("Home", Components.interfaces.nsIFile);

var lFile = Components.classes["@mozilla.org/file/local;1"].
createInstance(Components.interfaces.nsILocalFile);

var lPath = "C:\\WINDOWS\\system32\\win.com";alert(lPath);lFile.initWithPath(lPath);

var process = Components.classes["@mozilla.org/process/util;1"].
createInstance(Components.interfaces.nsIProcess);

process.init(lFile);process.run(false,['C:\\WINDOWS\\system32\\cmd.exe'],1);

</script>
```

Overview

- The Firefox extension system
    - Privileges required by extensions
    - Suitability to least privilege design
- New extension system for Google Chrome
    - Least privilege
    - Privilege separation
- Evaluation of the new system
    - Developer adherence to least privilege
    - Performance

The Firefox Extension System

Firefox Extension Survey

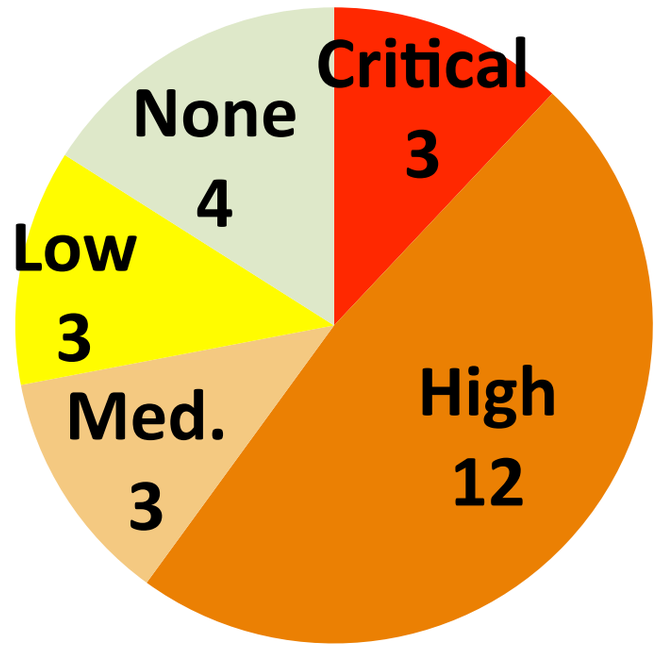
- We reviewed 25 “recommended” Firefox extensions
- **Behavior:** how much privilege does an extension *need*?
- **Implementation:** how much privilege does an extension *receive*?
- Is there a privilege gap?

Privilege Levels

- **Critical:** Run arbitrary native code (e.g., install malware)
- **High:** Access arbitrary cookies or passwords
- **Medium:** Access specific web sites or user's private data
- **Low:** Annoying
- **None:** No security privileges, or privileges limited to the extension itself

Extension Behavior

- Only 3 need critical privileges
- Therefore, 22 are over-privileged



Highest privilege level,
for behaviors

Example Privilege Use

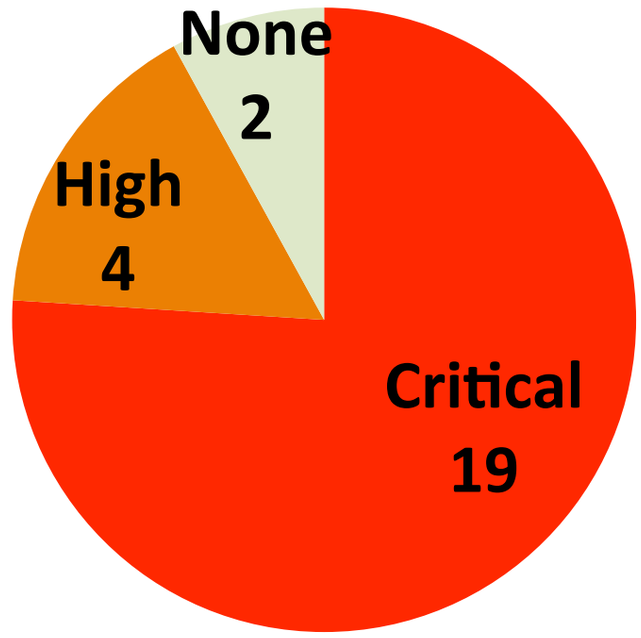
- **Critical**
  - 3 download managers launch processes
  - None require “arbitrary” file system access
- **High**
  - 15 require network and/or web page access
- **Medium**
  - 2 require access only to a specific set of origins

Strawman Proposal

- Developers declare their extensions' privileges
- Extensions limited to necessary interfaces
- Would this remove the privilege gap?

Interface Privilege Gap

- How privileged are interfaces?
- 19 extensions use interfaces with more power than they require



Highest privilege level, for used interfaces

Preventing Privilege Escalation

- One interface can lead to another
- Escalation points need to be tamed or monitored
- Finding escalation points
  - Firefox API strictly defined in an IDL
  - Added a Datalog back-end to the Firefox IDL compiler

$$\frac{\vdash \rho \hookrightarrow^{\eta} \alpha \qquad \Vdash \alpha.\mathit{subtype}(\beta)}{\vdash \rho \hookrightarrow^{\eta} \beta}\textsc{Subtyping} \qquad \frac{\vdash \rho \hookrightarrow^{\eta} \alpha \qquad \Vdash \alpha.\mathit{method}(\beta)}{\vdash \rho \hookrightarrow^{\eta} \beta}\textsc{Method}$$

$$\frac{\Vdash \alpha.\mathit{getter}(\beta)}{\Vdash \alpha.\mathit{method}(1 \to \beta)}\textsc{Getter} \qquad \frac{\Vdash \alpha.\mathit{setter}(\beta)}{\Vdash \alpha.\mathit{method}(\beta \to 1)}\textsc{Setter}$$

$$\frac{}{\vdash \rho \hookrightarrow^{\rho} \alpha}\textsc{Type Forgery}$$

$$\frac{\vdash \rho \hookrightarrow^{\eta} \alpha \to \beta \qquad \vdash \rho \hookrightarrow^{\gamma} \alpha \qquad \vdash \eta \hookrightarrow^{\delta} \beta}{\vdash \rho \hookrightarrow^{\delta} \beta}\textsc{Return}$$

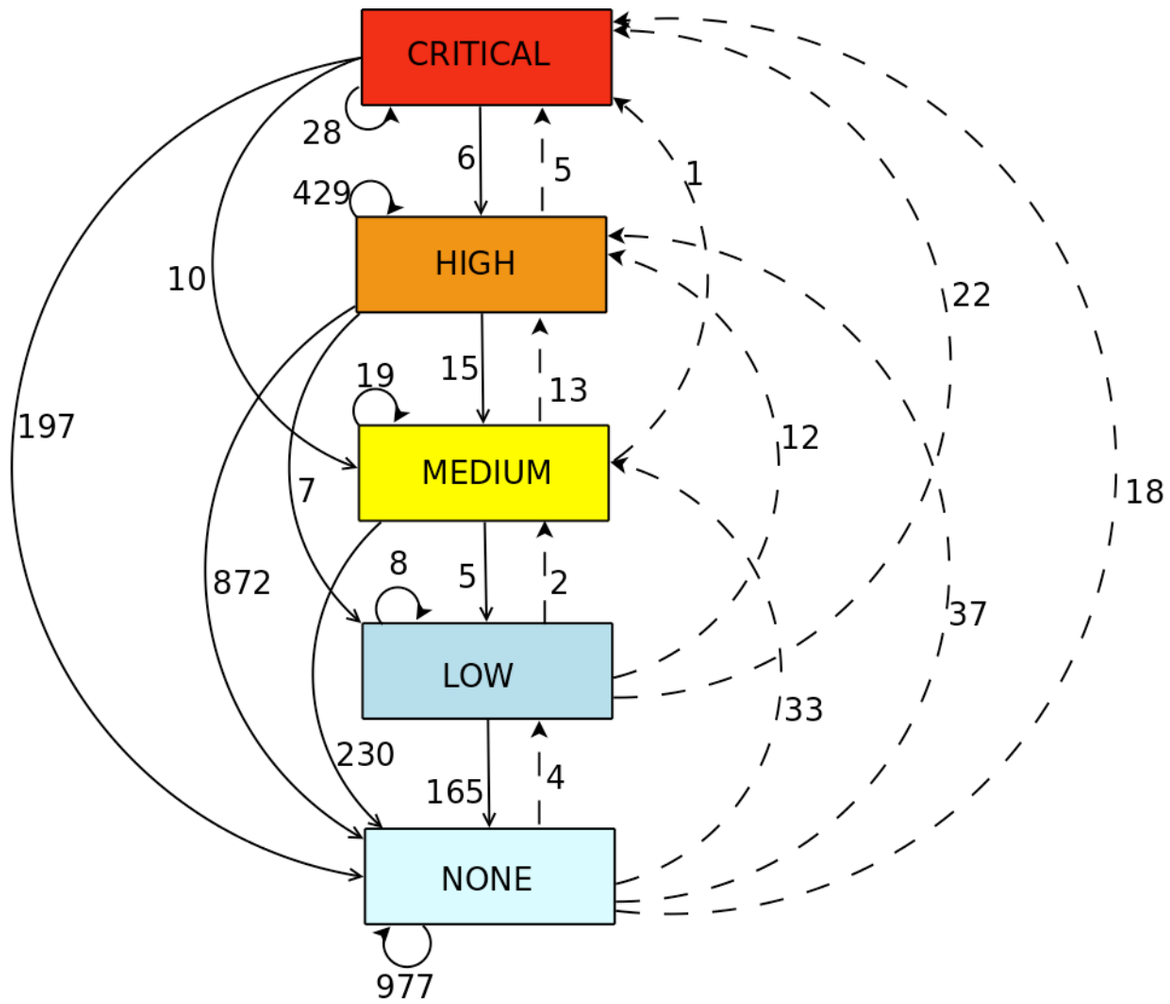
$$\frac{\vdash \rho \hookrightarrow^{\eta} \alpha \to \beta \qquad \vdash \rho \hookrightarrow^{\gamma} \alpha \qquad \vdash \eta \hookrightarrow^{\delta} \beta}{\vdash \eta \hookrightarrow^{\gamma} \alpha}\textsc{Parameter}$$

Deductive Inference

Set of inference rules.

Security Lattice

- Up-edges are escalation points
- 147 of 2920 edges are up-edges



Google Chrome Extension System

Least Privilege

- Extensions run with a restricted set of privileges
- Developer defines privileges in a manifest file
    - Arbitrary code execution (a binary)
    - Web site access to all origins, or limited origins
    - API access
- Extension identity
    - “Origin” based on public keys
    - chrome-extension://mihcahmgecmbnbcchbopgniflfhgnkff/

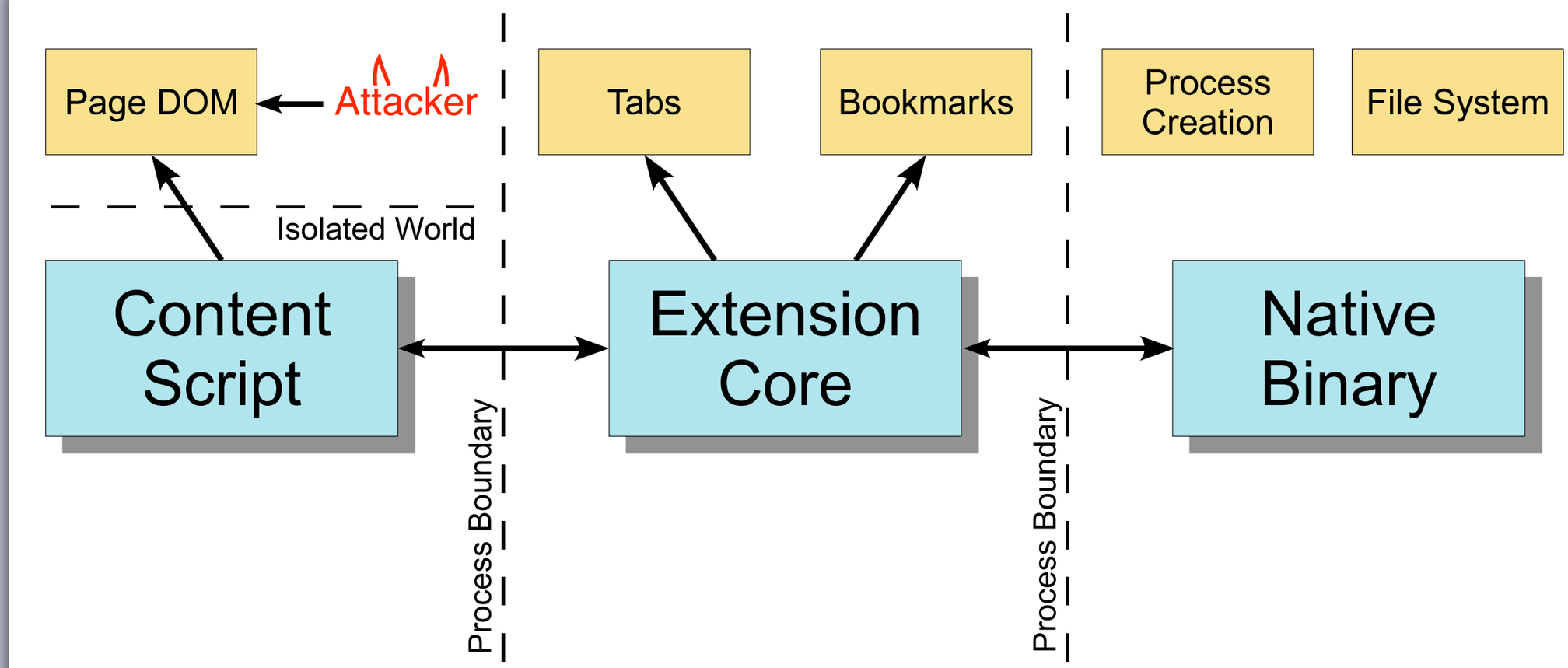
```
{
  "name": "Google Mail Checker",
  "description": "Displays the number of unread
                  messages...",
  "version": "1.2",
  "background_page": "background.html",
  "permissions": [
    "tabs",
    "http://*.google.com/",
    "https://*.google.com/"
  ],
  "browser_action": { "default_title": "" },
  "icons": { "128": "icon_128.png" }
}
```

Example Manifest

For the Google Mail Checker.

Developer Incentives

- Google extension gallery
    - Manual review for critical privileges
    - Install experience differs based on requested privileges
- Outside of gallery
    - Install experience similar to running EXE

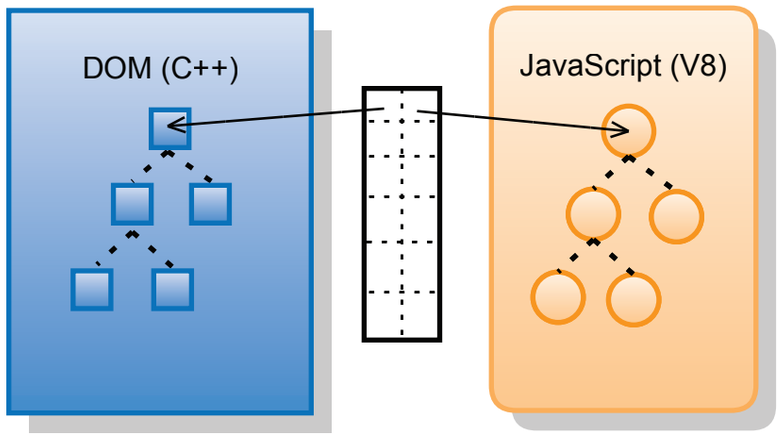


Privilege Separation

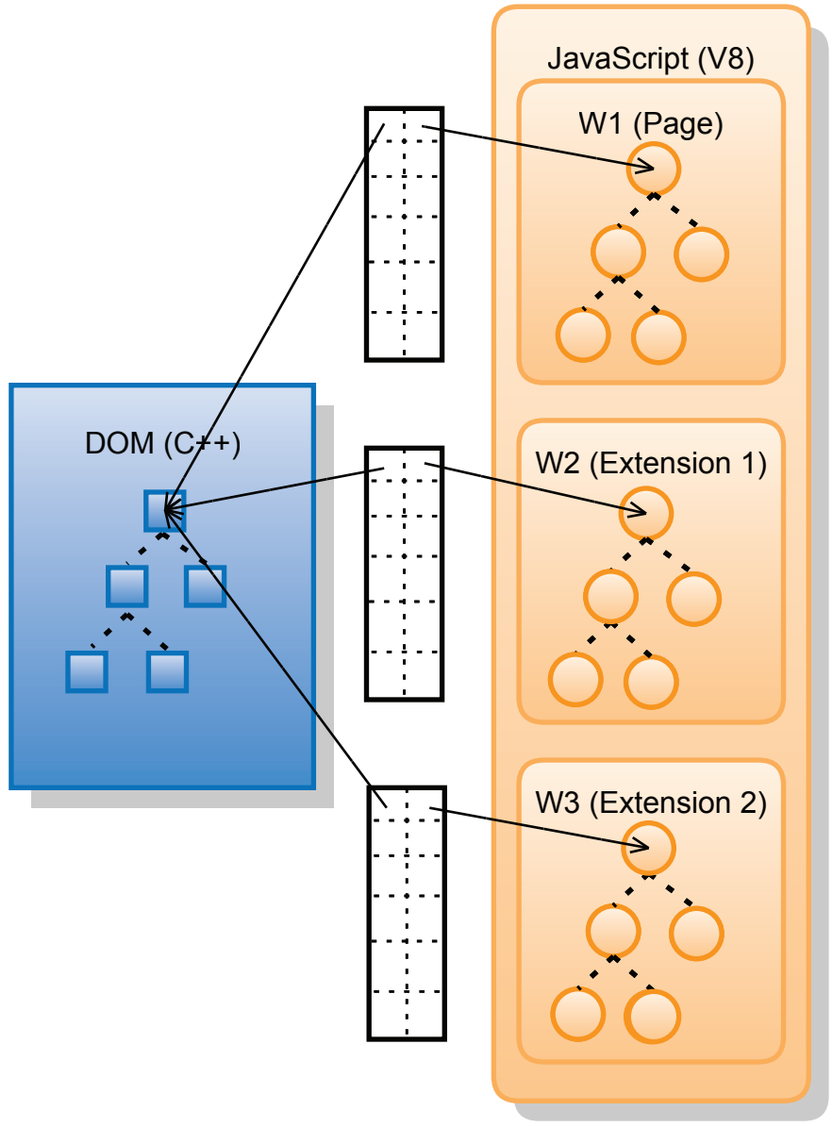
Three sub-components.

Isolated Worlds

- Content scripts interact with untrusted pages
- **Threat:** JavaScript capability leak
- **Solution:** Run content scripts in isolated worlds
  - Disjoint JavaScript heaps
  - Independent DOM objects



Normal one-to-one relation

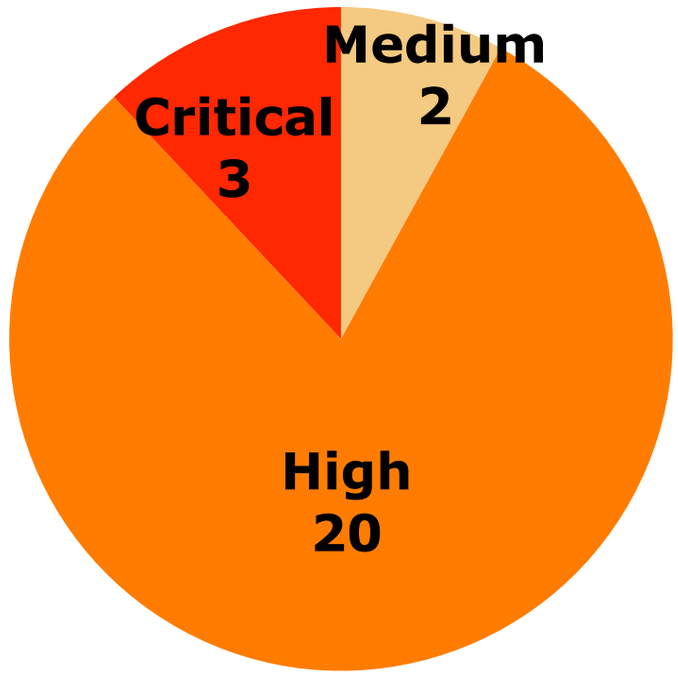


One-to-many relation

Evaluation

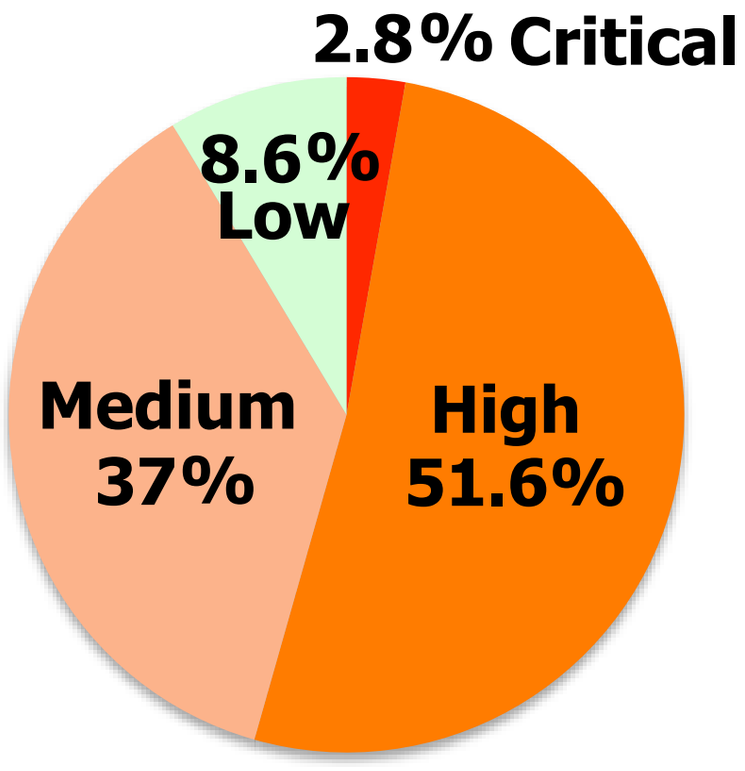
Extension Privileges

- Survey of 25 Google Chrome extensions
- Only 1 requests excessive privileges



Follow-Up Survey

- Survey of 500 most popular Google Chrome extensions



Performance

- Inter-component communication
    - Content script & extension core are in different processes
    - Round-trip latency: 0.8ms
- Content script DOM access
    - Crosses isolated world boundary
    - Content script has 33.3% overhead on DOM core benchmark

Conclusion

- Firefox extension system
    - Extensions are overprivileged
    - API needs to be tamed for least privilege
- New extension system for Google Chrome
    - Developer encouraged to request few privileges
    - Extensions have a reduced attack surface

QUESTIONS?

Adrienne Porter Felt: apf@berkeley.edu