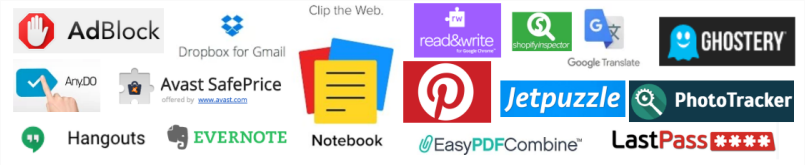
EmPoWeb: Empowering Web Applications with Browser Extensions [S&P'19]

Dolière Francis Somé — doliere.some@cispa.de

Rendez-vous de la Recherche et de l'Enseignement de la Sécurité des Systèmes d'Information (RESSI) — 18.12.2020

Browser Extensions vs. Web Applications



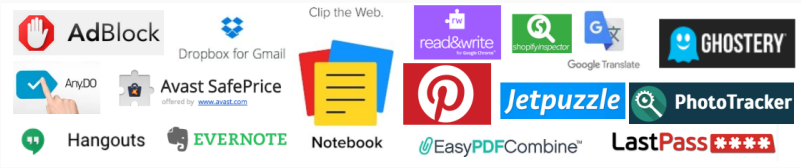
Browser Extensions vs. Web Applications

Web applications are restricted

- Same Origin Policy (SOP): can only access same-site data, cookies, etc.
- Cannot directly access extensions contexts

Extensions are privileged

- Not subject to SOP: can access user sensitive data on all sites
- Can directly manipulate web applications



Threat Model

2 threat models usually considered for extensions security

- Malicious extensions – [Jagpal et al. USENIX'15]
- Vulnerable extensions
  - Web Attacker – [Bandhakavi et al. USENIX'10, Carlini et al. USENIX'12, Calzavara et al. ETAPS'15]

Threat Model

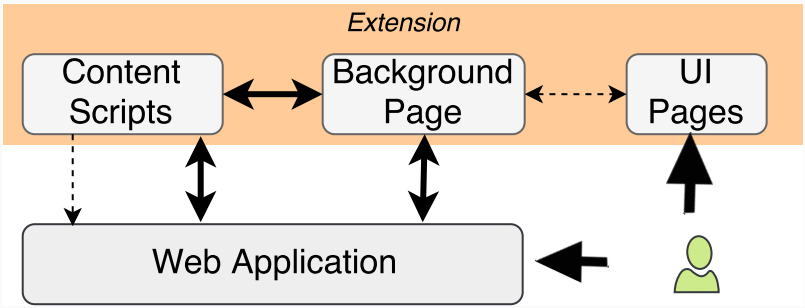
2 threat models usually considered for extensions security

- Malicious extensions – [Jagpal et al. USENIX'15]
- Vulnerable extensions
  - Web Attacker – [Bandhakavi et al. USENIX'10, Carlini et al. USENIX'12, Calzavara et al. ETAPS'15]

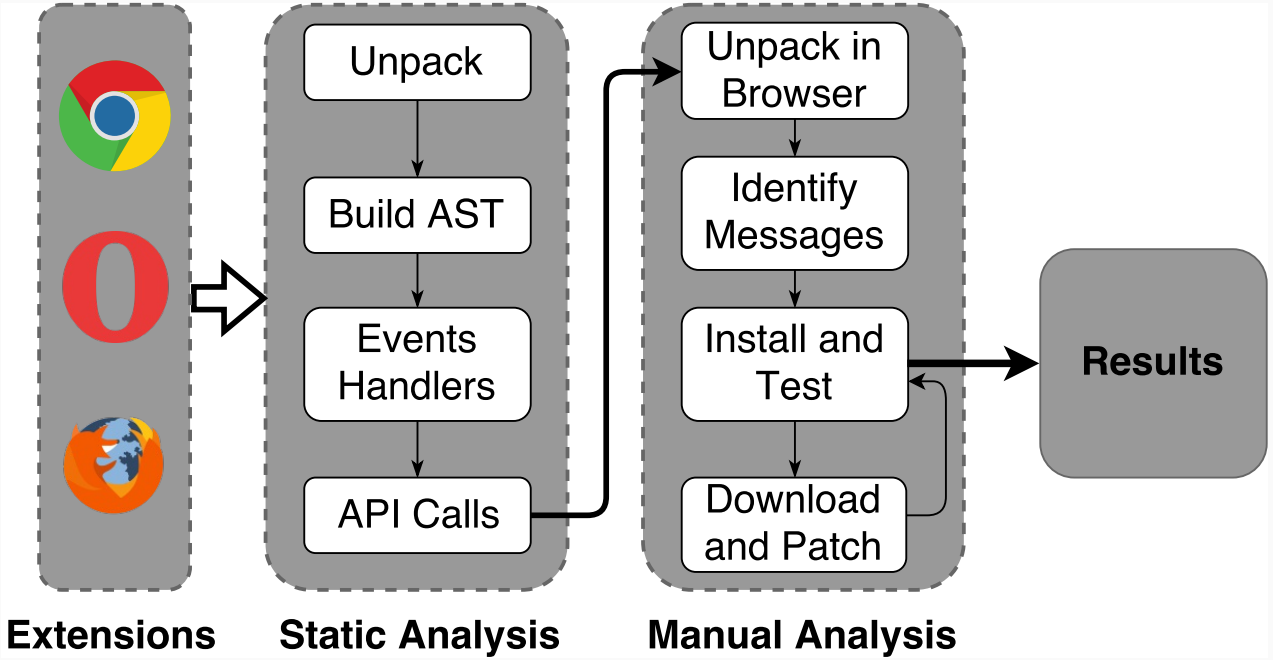
Our focus is the *web attacker*:

Exploit vulnerabilities in extensions through message passing APIs

- `postMessage`
- `onMessageExternal`



Methodology: Static and Manual Analysis



Methodology: Static and Manual Analysis

1. Statically analyze extensions code
   - Listening for messages
   - Triggering sensitive API calls

   Flag suspicious extensions

Methodology: Static and Manual Analysis

1. Statically analyze extensions code
   - Listening for messages
   - Triggering sensitive API calls

   Flag suspicious extensions
2. Manually review suspicious extensions
   - Eliminate false positives
   - Discover exploitable sensitive APIs
   - Construct messages signatures
   - Mount exploits

Methodology: Static and Manual Analysis

1. Statically analyze extensions code
   - Listening for messages
   - Triggering sensitive API calls

   Flag suspicious extensions
2. Manually review suspicious extensions
   - Eliminate false positives
   - Discover exploitable sensitive APIs
   - Construct messages signatures
   - Mount exploits

The analysis tool — `http://www-sop.inria.fr/members/Doliere.Some/empoweb/extsanalyzer/` — `https://gitlab.com/doliere/extsanalyzer`

Results: ∼200 Vulnerable Extensions Exploited

	Chrome	Firefox	Opera	Total
Extensions analyzed	66,401	9,391	2,523	78,315
Execute Code	15	2	2	19
Bypass Same Origin Policy	48	9	6	63
Read Cookies	8	-	-	8
Read Browsing History	40	-	-	40
Read Bookmarks	37	1	-	38
Get Extensions Installed	33	-	-	33
Store/Retrieve Data	85	2	3	90
Trigger Downloads	29	5	2	36
Total of unique extensions	**171**	**16**	**10**	**197**

DEMO: Bypass SOP and Read Cookies

Erail.in: Chrome extension with ∼405k users that

- exposes all user cookies to any web application
- allows to bypass the Same Origin Policy (SOP)

More demos and videos at – `http://www-sop.inria.fr/members/Doliere.Some/empoweb/extensions/`

Case study: static analysis [1/2]

Static analysis output on the *eRail.in* Chrome extension


```
{
    "com_via_cs": {
        "to_back": {
            "back": {
                "ajax": {
                    "$.get": "",
                    "$.post": "",
                    "$.ajax": "",
                    "XMLHttpRequest": ""
                },
                "cookies": {
                    "chrome.cookies.getAll": "",
                    "chrome.cookies.remove": "",
                    "cookies": ""
                }
            }
        }
    }
}
```

Case study: manual analysis [2/2]

eRail.in is exploitable to bypass SOP and get user cookies

- Read user cookies

```
{
  ACTION:  "GETCOOKIE"
}
```

- Access user sensitive data (i.e. mails on Gmail)

```
{
  ACTION: "GET_BLOB",
  URL: "https://mail.google.com"
}
```

Reporting to Browser Vendors

- Firefox and Opera removed the vulnerable extensions
- Chrome - planning to work on vulnerable extensions

  Do not use these exploits against users of the vulnerable (Chrome) extensions

Reporting to Browser Vendors

- Firefox and Opera removed the vulnerable extensions
- Chrome - planning to work on vulnerable extensions

  Do not use these exploits against users of the vulnerable (Chrome) extensions

- Important media coverage (100+ links)
- Discussions in the community

Mitigation and Future Work

- Quick fix: more rigorous review process
  - Static analysis (tools like ours) can help
  - Cross-browser tools for analyzing extensions

Mitigation and Future Work

- Quick fix: more rigorous review process
  - Static analysis (tools like ours) can help
  - Cross-browser tools for analyzing extensions
- Changes in extensions architecture
  - Fine-grained permission system to track origin of messages in extensions
  - Detect suspicious exchanges between extensions and web apps

Mitigation and Future Work

- Quick fix: more rigorous review process
  - Static analysis (tools like ours) can help
  - Cross-browser tools for analyzing extensions
- Changes in extensions architecture
  - Fine-grained permission system to track origin of messages in extensions
  - Detect suspicious exchanges between extensions and web apps
- Future Work
  - Exploring more security and privacy threats
  - Proposals to make extensions more trustworthy

Conclusion

- Vulnerable extensions can be exploited by web applications to
  - access sensitive user data, cookies, etc.
  - execute malicious code in extensions context

  etc.

Conclusion

- Vulnerable extensions can be exploited by web applications to
  - access sensitive user data, cookies, etc.
  - execute malicious code in extensions context

  etc.
- Need tools and methods to find such vulnerabilities in extensions
  - static analysis tools like ours can help
  - changes in extensions system to consider those threats

Conclusion

- Vulnerable extensions can be exploited by web applications to
  - access sensitive user data, cookies, etc.
  - execute malicious code in extensions context

  etc.
- Need tools and methods to find such vulnerabilities in extensions
  - static analysis tools like ours can help
  - changes in extensions system to consider those threats
- More work on browser extensions security and privacy
  - consider more threats
  - make extensions more trustworthy

Final Notes

Works on browser extensions at CISPA

- Static analysis tool under submission
- CORS headers manipulations
- Dynamic analysis of extensions for vulnerabilities
- Clickjacking with web accessible resources
- Secure Contexts in browser extensions

Thanks !
Questions !