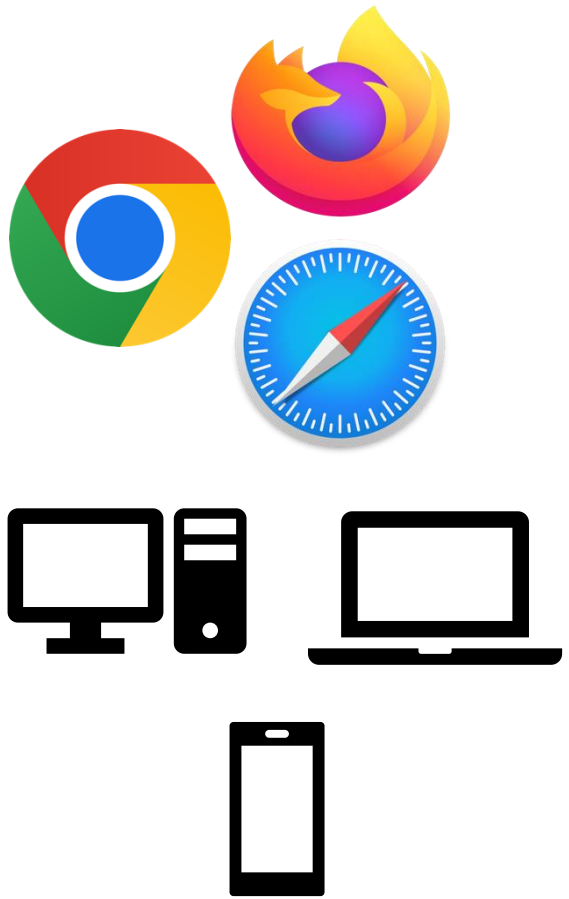
Shiny Shells, Rusty Cores
A Crowdsourced Security Evaluation of Integrated Web Browsers

Gertjan Franken, Pieter Claeys, Tom Van Goethem, Lieven Desmet

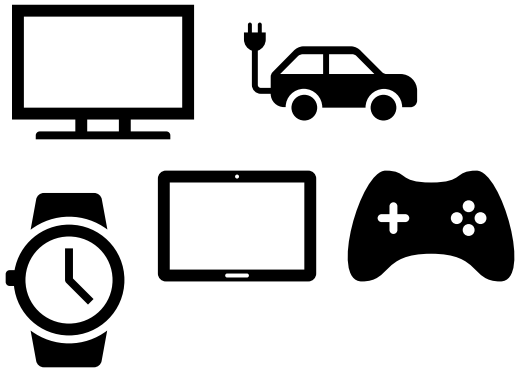
DistriNet
KU LEUVEN

Jacques Linard - Still life with shells and coral

Standalone web browsers



Integrated / embedded browsers



- Pre-installed
- Sometimes no alternative

What we already knew…

Standalone web browsers are complex and vulnerable

As shown by:

- Thousands of bug reports
- Plethora of security research

That's okay because:

- Dedicated security teams
- Bug reporting platforms
- Automated updates
- Version transparency

Underexplored

Some embedded web browsers are vulnerable

As shown by:

- Hacking of specific Smart TVs through embedded browser [1,2]
- E-reader apps and hardware [3]

That's okay because:

[1] https://www.blackhat.com/us-13/archives.html#Grattafiori

[2] Yann Bachy, Frédéric Basse, Vincent Nicomette, Eric Alata, Mohamed Kaâniche, Jean-Christophe Courrège, and Pierre Lukjanenko. Smart-tv security analysis: Practical experiments. In 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, pages 497–504, 2015.

[3] G. Franken, T. Van Goethem and W. Joosen, "Reading Between the Lines: An Extensive Evaluation of the Security and Privacy Implications of EPUB Reading Systems," 2021 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2021, pp. 1730-1747, doi: 10.1109/SP40001.2021.00015.

How secure are integrated browsers?

Crowdsourcing

- Participants enroll their own products
- Automated security evaluation

Vast diversity of products

- Closed-source firmware
- No comprehensive overview

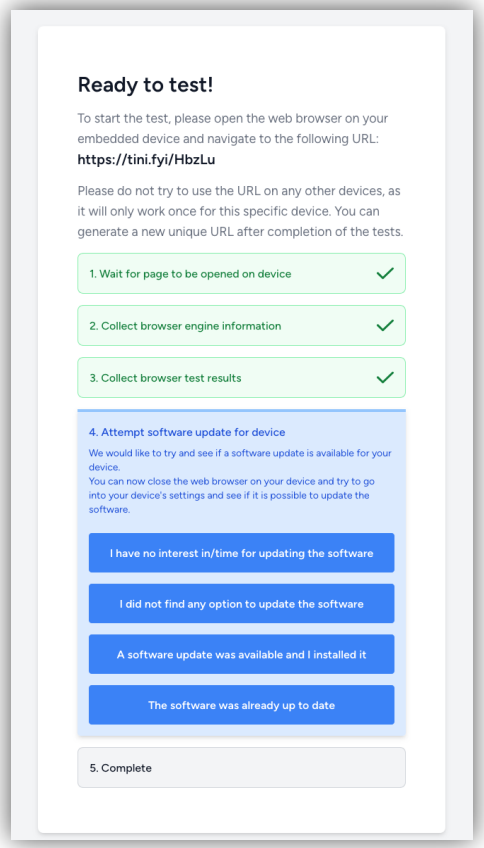
Real-world insights

- Are embedded browsers in the wild up-to-date?

CheckEngine

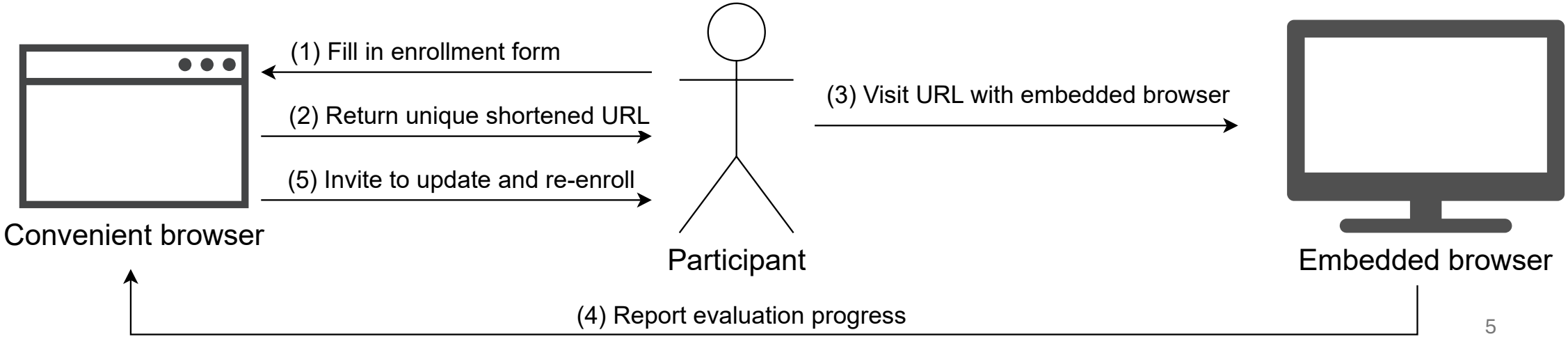
Automated crowdsourcing framework

https://github.com/DistriNet/CheckEngine



Primary design goal:

Limit user friction as much as possible!



Dynamic evaluation

- [ ] **Security policies are enabled and implemented correctly**

  - Extension of BrowserAudit [1]
  - Various policies
    - SOP
    - CSP
    - CORS
    - Cookie attributes
    - HSTS

- [ ] **Browser is up-to-date**

  - Browser feature fingerprinting script to validate user agent string

[1] Charlie Hothersall-Thomas, Sergio Maffeis, and Chris Novakovic. 2015. BrowserAudit: automated testing of browser security features. In Proceedings of the 2015 International Symposium on Software Testing and Analysis (ISSTA 2015). Association for Computing Machinery, New York, NY, USA, 37–47. https://doi.org/10.1145/2771783.2771789

Enrollments

- Recruitment
  - Distributing flyers at local industry and science events in Belgium
  - Online through LinkedIn and departmental-wide mail within university

- Total: 76 enrollments
  - 53 unique products
  - 68 unique software version

Results

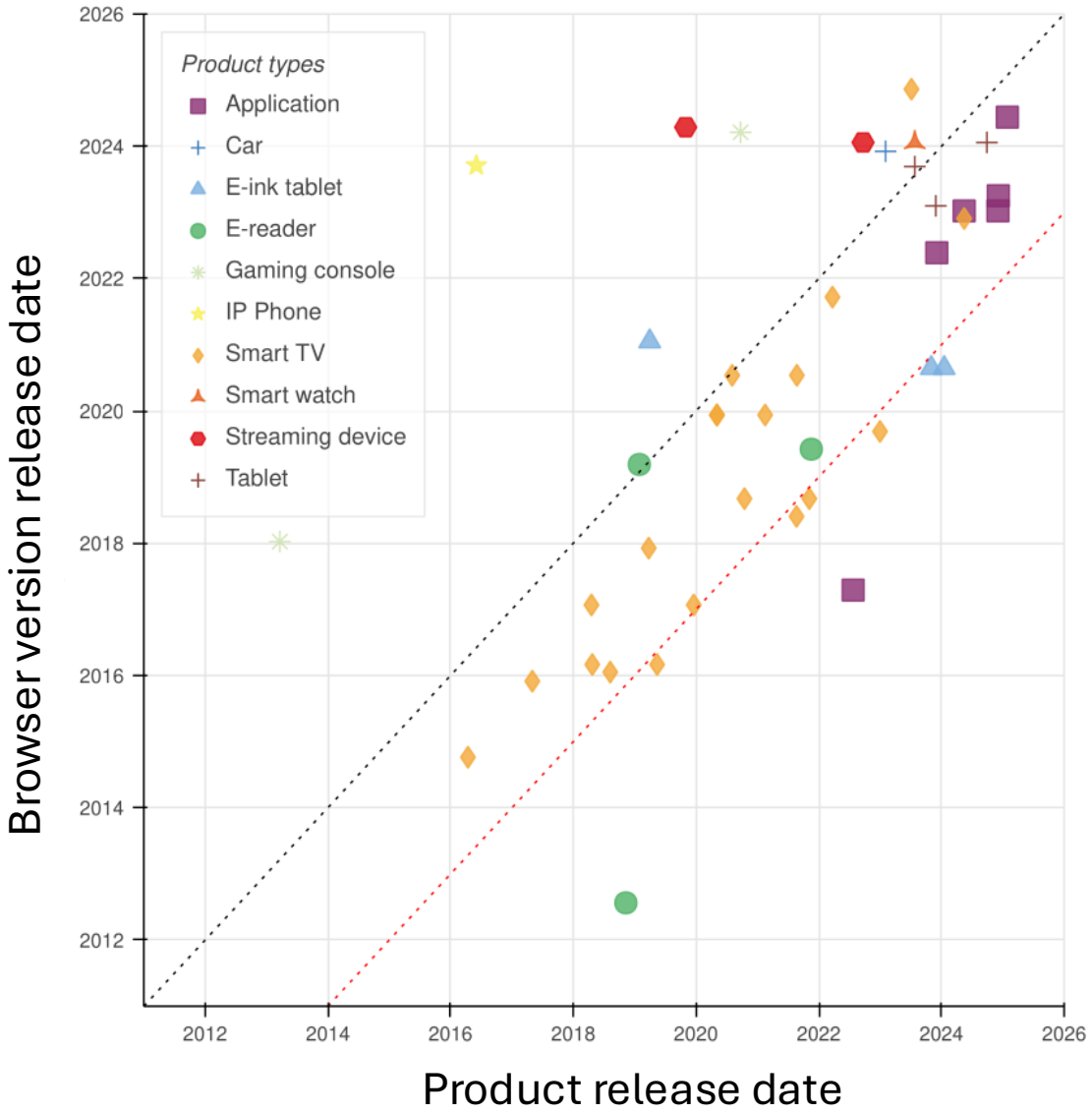
The good

In general, security policies are enabled and implemented correctly

The bad…

Many browsers were outdated and did not receive timely updates

Obsolescence at product market release



- Strong correlation between year product release and browser release
  - Median browser age upon release
    1.27 years

- Optimistic representation:
  disregard for potential updates after purchase

Some vendors shipping obsolete browsers advertise free (security) updates for several years

Update policy

N = 13
All products with multiple enrollments and different software versions

(Limited) data indicates that embedded browsers are only updated with major software updates

Frequent OS updates that exclude the browser might give the user a false sense of security

Type	Browser update	Software update
Application 1	⊠-●	⊠- M
Application 2	⊠-○-●	?
Car 1	⊠-○	⊠- m
E-ink tablet 1	⊠-○-○-○	⊠- p - p - p
E-ink tablet 2	⊠-●	⊠- M
E-reader 1	⊠-○-○-○-○	⊠- m - m - m - m
E-reader 2	⊠-●	⊠- M
Smart TV 1	⊠-○	⊠- m
Smart TV 2	⊠-●	⊠- M
Smart TV 3	⊠-○	⊠- m
Smart TV 4	⊠-○	?
Smart TV 5	⊠-○	?
Streaming device 1	⊠-○	?

⊠ Initial software version

Browser update

○ Software update

● Software update included browser update

Sofware update

p Patch

m Minor update

M Major update

? Unknown software update type

Case studies

E-ink tablet
- Market release: January 2024
- Chromium 85 (August 2020)

Reproduction of two $3000 and one $1000 vulnerabilities

Gaming platform applications
- Most recent software version
- Chromium 109 – 126 (January 2023 - June 2024)

UBISOFT

AMD Software Adrenalin Edition

Reproduction of various vulnerabilities and/or insecure configurations

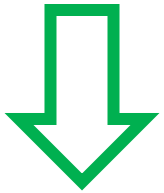
Old smart TV
- Market release: January 2021
- Chromium 79 (December 2019)

Reproduction of one $3000 vulnerability

Cause for obsolescence

Technical

Upgrading browser might break dependent components



Separate user-facing browser from browser engine used for UI

Oversight

Vendor is not aware

Neglect

Vendor does not care

Awareness and transparency

- E.g., Product security labels [1,2]
- Regulatory oversight

[1] Emami-Naeini, Pardis, et al. "Ask the experts: What should be on an IoT privacy and security label?." *2020 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2020.
[2] P. Morgner, C. Mai, N. Koschate-Fischer, F. Freiling and Z. Benenson, "Security Update Labels: Establishing Economic Incentives for Security Patching of IoT Consumer Products," 2020 IEEE Symposium on Security and Privacy (SP), San Francisco, CA, USA, 2020, pp. 429-446, doi: 10.1109/SP40000.2020.00021.

Takeaways

While most browsers feature essential security policies many remain ***unpatched for years****, leaving users exposed to considerable risks.*

This issue is amplified due to ***lack of transparency*** *and* ***misleading information about security updates****.*

Although we hope that vendors see this as a call to action to prioritize embedded browser security, we suspect ***external incentives are necessary****.*

DistriNet

KU LEUVEN

paper

linkedin.com/in/gertjan-franken
bsky.app/profile/gertjan.fr