RøB: Ransomware over Modern Web Browsers

Harun Oz[1], Ahmet Aris[1] , Abbas Acar[1] , Güliz Seray Tuncay[2], Leonardo Babun[1], and Selcuk Uluagac[1]

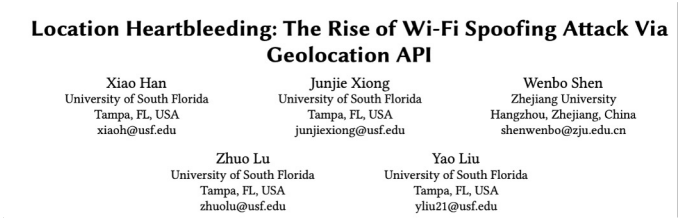
[1]Florida International University, [2]Google

32nd USENIX Security Symposium
9-11 August 2023, Anaheim, CA

Modern Web Applications

- The entire web ecosystem is evolving.
- The web applications are *more powerful & faster:*
  - File System Access (FSA) & Geolocation API
  - Runs low-level code via Webassembly (Wasm).
- Growing trend towards web applications:
  - Less expensive to maintain & update.
- They can be a new threat vector:
  - Can be exploited or
  - Can be abused.

New Threat Vector(s)



Location Heartbleeding: The Rise of Wi-Fi Spoofing Attack Via Geolocation API

Xiao Han
University of South Florida
Tampa, FL, USA
xiaoh@usf.edu

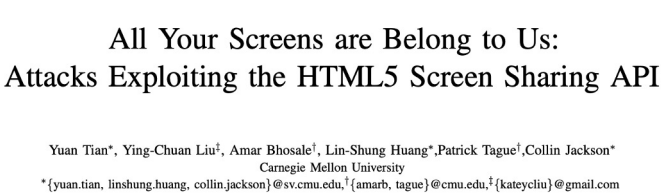
Junjie Xiong
University of South Florida
Tampa, FL, USA
junjiexiong@usf.edu

Wenbo Shen
Zhejiang University
Hangzhou, Zhejiang, China
shenwenbo@zju.edu.cn

Zhuo Lu
University of South Florida
Tampa, FL, USA
zhuolu@usf.edu

Yao Liu
University of South Florida
Tampa, FL, USA
yliu21@usf.edu

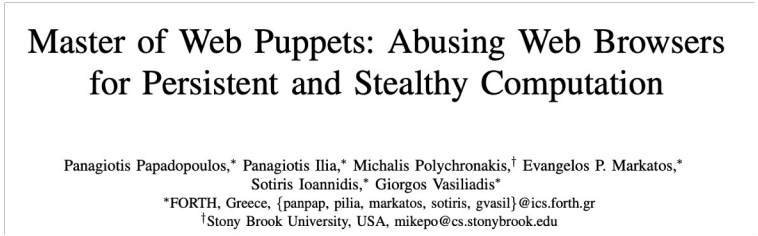
1-ACM SIGSAC Conference on Computer and Communications Security (CCS), 2022.



All Your Screens are Belong to Us:
Attacks Exploiting the HTML5 Screen Sharing API

Yuan Tian*, Ying-Chuan Liu‡, Amar Bhosale†, Lin-Shung Huang*,Patrick Tague†,Collin Jackson*
Carnegie Mellon University
*{yuan.tian, linshung.huang, collin.jackson}@sv.cmu.edu,†{amarb, tague}@cmu.edu,‡{kateycliu}@gmail.com

2-IEEE Symposium on Security and Privacy (S&P), 2014.



Master of Web Puppets: Abusing Web Browsers for Persistent and Stealthy Computation

Panagiotis Papadopoulos,* Panagiotis Ilia,* Michalis Polychronakis,† Evangelos P. Markatos,*
Sotiris Ioannidis,* Giorgos Vasiliadis*
*FORTH, Greece, {panpap, pilia, markatos, sotiris, gvasil}@ics.forth.gr
†Stony Brook University, USA, mikepo@cs.stonybrook.edu

3-Network and Distributed System Security Symposium (NDSS), 2021.

Most Websites Don't Need to Vibrate: A Cost–Benefit Approach to Improving Browser Security

Peter Snyder
University Of Illinois at Chicago
psnyde2@uic.edu

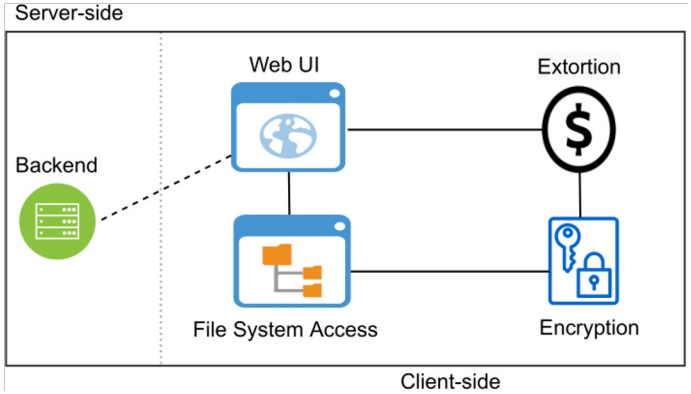
Cynthia Taylor
University Of Illinois at Chicago
cynthiat@uic.edu

Chris Kanich
University Of Illinois at Chicago
ckanich@uic.edu

4-ACM SIGSAC Conference on Computer and Communications Security (CCS), 2017.

Ransomware over Browsers

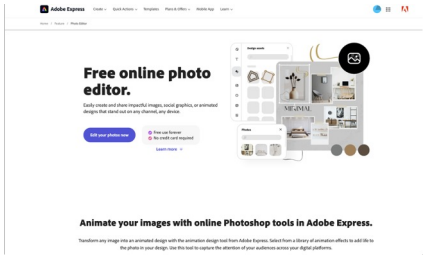
- In this work, we developed proof-of-concept browser-based ransomware - RøB
- Performs its malicious actions through the browser via FSA API and Wasm.
- The FSA API is working exactly as designed,
  - but abused by a malicious web app.



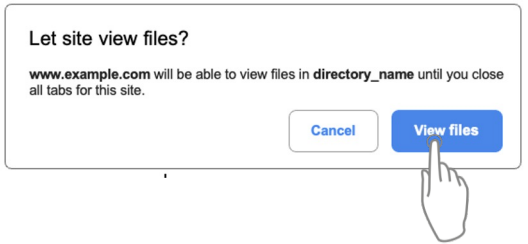
System Model

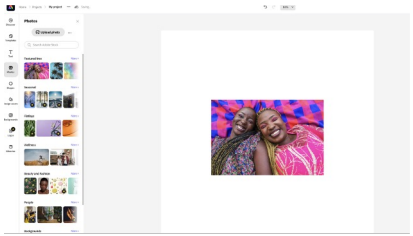
How does the FSA API work?

User visits the website.



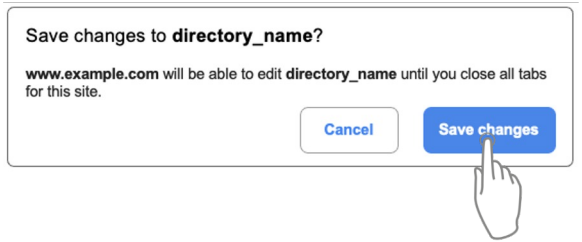
Website pops up read permission box.





Web application modifies user files.

Website pops up write permission box.



Impact Analysis

- ***Impact*** - local directories:
  - FDA: Full Directory Access
  - SDA: Sub-directory Access
- ***Impact -*** others:
  - Cloud-integrated directories
  - External storage devices
  - Network shared folders

Directory	Windows		Linux		macOS	
	FDA	*SDA*	*FDA*	*SDA*	*FDA*	*SDA*
Documents	✗	✓	✗	✓	✗	✓
Desktop	✗	✓	✗	✓	✗	✓
Pictures	✓	✓	✓	✓	✓	✓
Videos	✓	✓	✓	✓	✓	✓
Music	✓	✓	✓	✓	✓	✓
Downloads	✗	✓	✗	✓	✗	✓
Data Partition	✗	✓	✓	✓	✓	✓

✓ *browser-based ransomware can encrypt files in the directory.*
X *access is denied, preventing encryption.*

Cloud Provider	Versioning Scheme	Affected by RøB?
Google Drive	30 days or 100 versions	✗
Microsoft OneDrive	25 versions	✗
Dropbox	30 days (personal), 180 days (business)	✗
Apple iCloud	No versioning	✓
Box Individual	No versioning	✓

✓ *files are irrecoverable after a ransomware attack.*
X *files can be restored due to the cloud provider's versioning.*

Actual Threat

Security Measures

Effectiveness of Current Defense Solutions

- Static Analysis-based Solutions
  - *Evadable via code obfuscation.*

```
async function q1w2e3r4t5() {
    const v1b2n3m4 = await window['z1x2c3v4b5']();

    for await (const s5d6f7g8 of v1b2n3m4['u6y7t8r9']())
      if (s5d6f7g8['w8e9r7t6'] === 'file') {
        const k9l0i8u7 = await s5d6f7g8['o5p4a3s2']();
        const j6h5g4f3 = await k9l0i8u7['d1f2g3h4']();

        await z9x8c7v6(s5d6f7g8, k9l0i8u7);
      }
```

- Dynamic Analysis-based Solutions
  - *Fileless nature → No payload for analysis environment*
  - *All actions done by benign process of the browsers → False Negatives*
  - *Disadvantages of (Wasm) → No encryption system call*

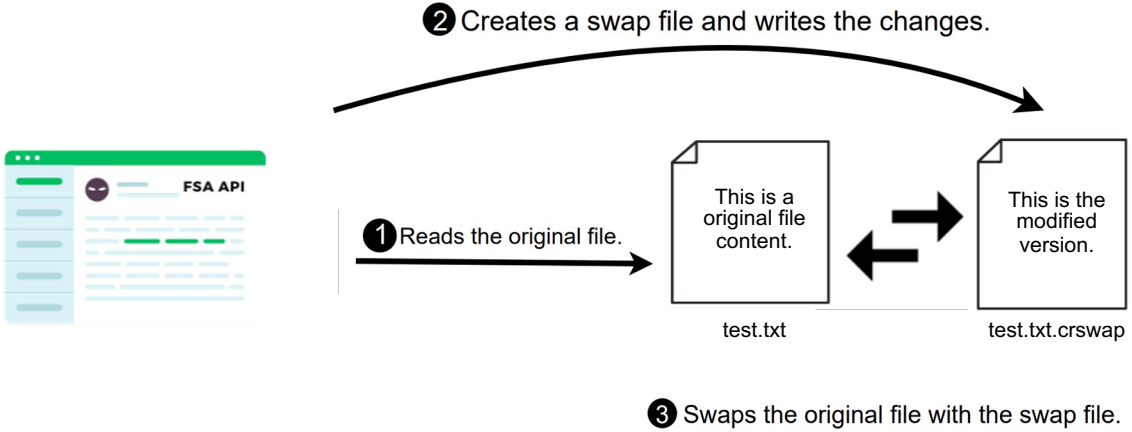
- Key-extraction-based Solutions
  - Possible but not practical (!)
  - Resource overhead

Defense Solutions

FIU

Solution – 1: Malicious File Identification via Hooking (1)

Distinct FSA API Behavior on the user-files.



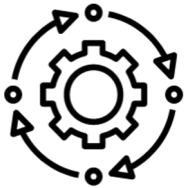
Hooking after the `write()` function call of the FSA API allow us analyze the both *modified files and original files.*

Solution – 1: Malicious file Identification via Hooking (2)

- Encryption operation **increases the entropy of a file** while **keeping its size relatively unchanged.**
- Benign modification **increases or decreases the size of the file** while **keeping the entropy unchanged.**

Dataset: 5000 files
(1000 for each file type)

Create 500K malicious
500K modified versions of
files



Calculate entropy and size differences on the files and train a machine learning classifier

~%99 Accuracy rate on average for every type of files in our dataset.

- It is not a silver bullet.
  - The adaptive attacker can arrange the size and entropy of the encrypted file.

Solution - 2: Local Activity Monitoring

- Specifically we monitor the following local activities:
  - The FSA API function calls
  - Browser process system calls
  - File system activities of the web app
- Distinct features between the benign and malicious FSA usage.
- False negative alerts
  - Benign apps might perform mass modification.
- It can be integrated to the current defense solutions.

RøB's encryption | **Bening Modification (VSCode)**

getFile()	- file1.txt	getFile()	- file1.txt
Write()	- file1.txt	Write()	- file1.txt
Write.Close()	- file1.txt	Write.Close()	- file1.txt
getFile()	- file2.txt	getFile()	- file1.txt
Write()	- file2.txt	Write()	- file1.txt
Write.Close()	- file2.txt	Write.Close()	- file1.txt
getFile()	- file3.txt	getFile()	- file2.txt
Write	- file3.txt	Write()	- file2.txt
Write.close()	- file3.txt	Write.Close()	- file2.txt

A side-by-side comparison of FSA API function calls of RøB (left) and VSCode (right)

RøB's encryption | **Bening Modification (VSCode)**

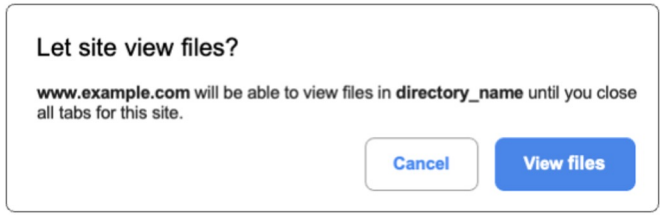
OPEN	- file1.txt	OPEN	- file1.txt
ACCESS	- file1.txt	ACCESS	- file1.txt
CLOSE	- file1.txt	CLOSE	- file1.txt
CREATE	- file1.txt.crswp	CREATE	- file1.txt.crswp
MODIFY	- file1.txt.crswp	MODIFY	- file1.txt.crswp
MOVED_FROM	- file1.txt.crswp	MOVED_FROM	- file1.txt.crswp
MOVED_TO	- file1.txt	MOVED_TO	- file1.txt
OPEN	- file2.txt	OPEN	- file1.txt
ACCESS	- file2.txt	ACCESS	- file1.txt
CLOSE	- file2.txt	CLOSE	- file1.txt
CREATE	- file2.txt.crswp	CREATE	- file1.txt.crswp
MODIFY	- file2.txt.crswp	MODIFY	- file1.txt.crswp
MOVED_FROM	- file2.txt.crswp	MOVED_FROM	- file1.txt.crswp
MOVED_TO	- file2.txt	MOVED_TO	- file1.txt

A side-by-side comparison of file system activities of RøB (left) and VSCode (right)

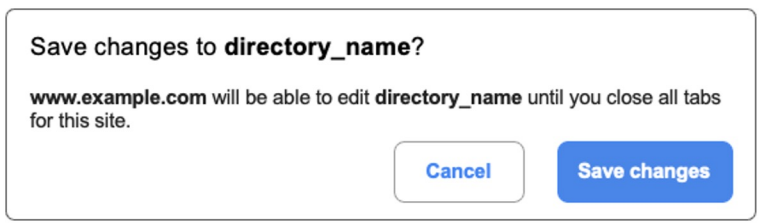
Solution - 3: New User Interface Design (1)

- We found the following issues in the current permission boxes:

  - ❌ They look very similar.
  - ❌ Security risks are not mentioned.
  - ❌ The changes made by the web applications are not stated.
  - ❌ The possible access to the subdirectories are not indicated.



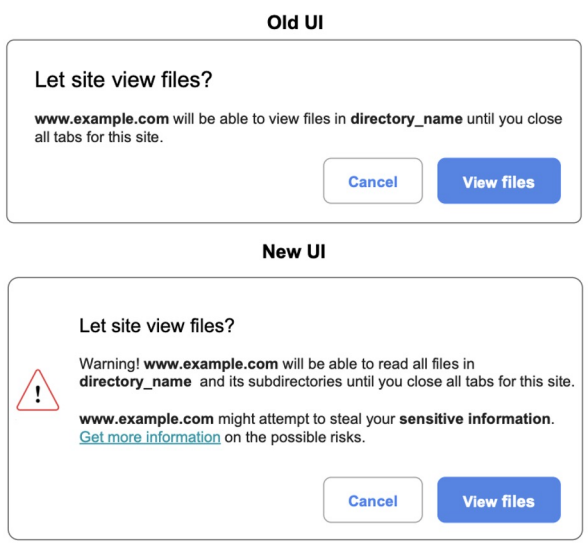
Read access permission box



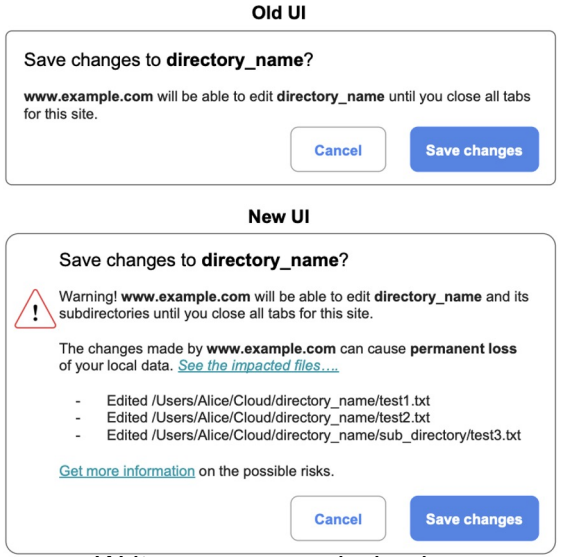
Write access permission box

Solution - 3: New User Interface Design (2)

- We aim to better-inform the users about the risks and implications of allowing web applications to interact with local files.
- It is still under the user's control.
- User does not need to install any software.



Read access permission box



Write access permission box

Concluding remarks

- For the first time in the literature, we extensively studied **a new attack vector** for ransomware over the browsers.
- We conducted **comprehensive impact analysis** on three different OSs, 29 distinct directories, five cloud providers, and five antivirus solutions.
- We evaluated the **(in)effectiveness of existing ransomware detection solutions** against this new type of ransomware.
- We proposed **three potential defense solutions** to mitigate this new attack vector.

FIU

Selected References (1)

[1]-Harun Oz, Ahmet Aris, Albert Levi, and A. Selcuk Uluagac. 2022. A Survey on Ransomware: Evolution, Taxonomy, and Defense Solutions. ACM Comput. Surv. 54, 11s, Article 238 (January 2022), 37 pages. https://doi.org/10.1145/3514229.

[2]-Shrenik Bhansali, Ahmet Aris, Abbas Acar, Harun Oz, and A. Selcuk Uluagac. A First Look at Code Obfuscation for WebAssembly. 15th ACM Conference on Security and Privacy in Wireless and Mobile Networks. https://doi.org/10.1145/3507657.3528560.

[3]-Abbas Acar, Long Lu, Engin Kirda, and A. Selcuk Uluagac, An Analysis of Malware Trends in Enterprise Networks, The 22nd Information Security Conference (ISC), 2019.

[4]-E. Tekiner, A. Acar, A. S. Uluagac, E. Kirda and A. A. Selcuk, "SoK: Cryptojacking Malware," 2021 IEEE European Symposium on Security and Privacy (EuroS&P), Vienna, Austria, 2021, pp. 120-139, doi: 10.1109/EuroSP51992.2021.00019.

[5]-Faraz Naseem, Ahmet Aris, Leonardo Babun, Ege Tekiner, and A. Selcuk Uluagac, “MINOS: A Lightweight Real-Time Cryptojacking Detection System,” Network and Distributed System Security Symposium (NDSS), 2021

[6]- Ege Tekiner, Abbas Acar, and A. Selcuk Uluagac, “A Lightweight IoT Cryptojacking Detection Mechanism in Heterogeneous Smart Home Networks,” Network and Distributed System Security Symposium (NDSS), 2022.

[7]-A. van der Heijden and L. Allodi, “Cognitive triaging of phishing attacks,” in 28th USENIX Security Symposium, 2019.

[8]-W3C, “File system access,” https://wicg.github.io/file-system-access/, 2023.

[9]-M. Weeks, “Internal affairs: Hacking file system access from the web, The Black Hat USA, 2021.

[10]-Xiao Han, Junjie Xiong, Wenbo Shen, Zhuo Lu, and Yao Liu. 2022. Location Heartbleeding: The Rise of Wi-Fi Spoofing Attack Via Geolocation API. In Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security (CCS '22). Association for Computing Machinery, New York, NY, USA, 1383–1397. https://doi.org/10.1145/3548606.3560623.

FIU

Selected References (2)

[11]-Peter Snyder, Cynthia Taylor, and Chris Kanich. 2017. Most Websites Don't Need to Vibrate: A Cost-Benefit Approach to Improving Browser Security. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security (CCS '17). Association for Computing Machinery, New York, NY, USA, 179–194. https://doi.org/10.1145/3133956.3133966

[12]-Tian, Yuan & Liu, Ying-Chuan & Bhosale, Amar & Huang, Lin-Shung & Tague, Patrick & Jackson, Collin. (2014). All your screens are belong to us: Attacks exploiting the HTML5 screen sharing API. Proceedings - IEEE Symposium on Security and Privacy. 10.1109/SP.2014.10.

[13]-Papadopoulos, Panagiotis, Panagiotis Ilia, Michalis Polychronakis, Evangelos P. Markatos, Sotiris Ioannidis and Giorgos Vasiliadis. “Master of Web Puppets: Abusing Web Browsers for Persistent and Stealthy Computation.” Network and Distributed System Security Symposium (NDSS), 2021.

[14]-A. P. Felt, R. W. Reeder, A. Ainslie, H. Harris, M. Walker, C. Thompson, M. E. Acer, E. Morant, and S. Consolvo, “Rethinking connection security indicators,” in Twelfth Symposium on Usable Privacy and Security, 2016.

[15]- L.-S. Huang, A. Moshchuk, H. J. Wang, S. Schecter, and C. Jackson, “Clickjacking: Attacks and defenses,” in 21st USENIX Security Symposium, 2012.

[16]-A. Kharaz, S. Arshad, C. Mulliner, W. Robertson, and E. Kirda, “UNVEIL: A large-scale, automated approach to detecting ransomware,” in 25th USENIX Security Symposium, 2016.

[17]-A. P. Felt, S. Egelman, D. A. Matthew Finifter, and D. Wagner, “How to ask for permission,” in 7th USENIX Workshop on Hot Topics in Security, 2012.

Q&A

Thank You!

Harun Oz

Email: hoz001@fiu.edu

Personal Website: harunoz.net

Linkedin: linkedin.com/in/ozharun/

Code & Data: https://go.fiu.edu/RoB_Code

Lab Website: csl.fiu.edu

Code & Data

CSL Lab