Harun Oz\textsuperscript{1}, Ahmet Aris\textsuperscript{1}, Abbas Acar\textsuperscript{1}, Güliz Seray Tuncay\textsuperscript{2}, Leonardo Babun\textsuperscript{1}, and Selcuk Uluagac\textsuperscript{1}

\textsuperscript{1}Florida International University, \textsuperscript{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)

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


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

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.
How does the FSA API work?

1. User visits the website.
2. Website pops up read permission box.
3. Website pops up write permission box.
4. Web application modifies user files.
Impact Analysis

- **Impact** - local directories:
  - FDA: Full Directory Access
  - SDA: Sub-directory Access

- **Impact** - others:
  - Cloud-integrated directories
  - External storage devices
  - Network shared folders

<table>
<thead>
<tr>
<th>Directory</th>
<th>Windows FDA</th>
<th>Windows SDA</th>
<th>Linux FDA</th>
<th>Linux SDA</th>
<th>macOS FDA</th>
<th>macOS SDA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Documents</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Desktop</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Pictures</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Videos</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Music</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Downloads</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Data Partition</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Cloud Provider</th>
<th>Versioning Scheme</th>
<th>Affected by RøB?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Google Drive</td>
<td>30 days or 100 versions</td>
<td>X</td>
</tr>
<tr>
<td>Microsoft OneDrive</td>
<td>25 versions</td>
<td>X</td>
</tr>
<tr>
<td>Dropbox</td>
<td>30 days (personal), 180 days (business)</td>
<td>X</td>
</tr>
<tr>
<td>Apple iCloud</td>
<td>No versioning</td>
<td>✔</td>
</tr>
<tr>
<td>Box Individual</td>
<td>No versioning</td>
<td>✔</td>
</tr>
</tbody>
</table>

✔ files are irrecoverable after a ransomware attack. X files can be restored due to the cloud provider's versioning.
Effectiveness of Current Defense Solutions

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

- 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
Distinct FSA API Behavior on the user-files.

1. Reads the original file.
2. Creates a swap file and writes the changes.
3. Swaps the original file with the swap file.

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.**

```
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 - 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 - 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 - 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 - 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.
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.

Old UI

Read access permission box

Old UI

Write access permission box

New UI

Warning: www.example.com will attempt to steal your sensitive information. Get more information on the possible risks.

Old UI

Old UI
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.
Selected References (1)


Selected References (2)


[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.


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