Fine penetration tests for fine websites

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Pentest-Report Tor Browser UI Changes 04.2023

Cure53, Dr.-Ing. M. Heiderich, M. Pedhapati, L. Herrera

Index

Fine penetration tests for fine websites

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Introduction

"We believe everyone should be able to explore the internet with privacy. We are the Tor Project, a 501(c)(3) US nonprofit. We advance human rights and defend your privacy online through free software and open networks."

From https://www.torproject.org/

This report describes the results of a security assessment of the Tor Browser complex, with a very specific focus on the recent changes introduced to the Tor Browser's UI towards usability improvements. The project, which included a penetration test and a dedicated source code audit, was carried out by Cure53 in April 2023.

Registered as *TTP-02*, the examination was requested by The Tor Project (TTP) in March 2023 and then scheduled to start in April 2023, with enough time for preparations for the responsible teams. It marks the second security-centered cooperation between Cure53 and TTP.

In terms of the exact timeline and specific resources allocated to *TTP-02*, Cure53 completed the research in CW16 and CW17 of 2023. In order to achieve the expected coverage for this task, a total of ten days were invested. In addition, it should be noted that a team of three senior testers was formed and assigned to prepare, execute, and deliver this project. Given the tasks at hand, the examination was contained into a single work package (WP):

- **WP1**: Crystal-box penetration tests and code audits covering Tor Browser UI changes

It can be derived from the above that the so-called crystal-box methodology was utilized. Cure53 was provided with detailed testing information, as well as all other means of access required to complete the tests. Additionally, all sources corresponding to the test-targets were pinpointed, to make sure the project can be executed in line with the agreed-upon framework.

Overall, the project progressed effectively. To facilitate a smooth transition into the testing phase, all preparations were completed in CW15. Throughout the engagement, communications were conducted via a private Signal group. Stakeholders - including testers and internal staff from TTP - could participate in discussions in this space.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

The quality of the interactions throughout the test was excellent, with no outstanding queries. These steady exchanges contributed positively to the overall outcomes of this project. The scope was well prepared and clear, which played a major role in avoiding significant roadblocks during the test. Cure53 offered frequent status updates about the test and the emerging findings. Live-reporting was neither specifically requested, nor seen as a necessity in light of the observed type and number of flaws.

On the topic of discoveries, it shall be noted that the Cure53 team succeeded in achieving very good coverage of the WP1 scope. Of the seven security-related discoveries, three were classified as security vulnerabilities and four were categorized as general weaknesses with lower exploitation potential.

The total number of findings can be seen as moderate, which can be interpreted as a good sign regarding the overall security posture of the Tor Browser UI. Furthermore, another factor contributing to the positive verdict is the absence of major flaws. In fact, all of the discovered issues were ranked with *Low* scores at most, with the bulk of tickets simply containing information on advanced protections. Although the listed findings would not cause major damage, Cure53 recommends swiftly resolving them in order to further strengthen the Tor Browser.

The following sections first describe the scope and key test parameters. Next, all findings are discussed in grouped vulnerability and miscellaneous categories. Flaws assigned to each group are then discussed chronologically. In addition to technical descriptions, PoC and mitigation advice will be provided where applicable.

The report closes with drawing broader conclusions relevant to the April 2023 project. Based on the test team's observations and collected evidence, Cure53 elaborates on the general impressions and reiterates the verdict. The final section also includes tailored hardening recommendations for the Tor Browser complex, especially in relation to the introduced UI changes examined during this *TTP-02* assessment.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Scope

- **Penetration tests & code audits covering UI changes in Tor Browser**
  - **WP1**: Crystal-box penetration tests & code audits covering UI changes in Tor Browser
    - **General audit information:**
      - https://pad.riseup.net/p/TorBrowser-Usability-Changes-Audit
    - **Main repositories:**
      - https://gitlab.torproject.org/tpo/applications/tor-browser
    - **GitHub milestones:**
      - https://gitlab.torproject.org/groups/tpo/-/milestones/29#tab-issues
      - https://gitlab.torproject.org/groups/tpo/-/milestones/31#tab-issues
    - **Relevant commits and files:**
      - https://pad.riseup.net/p/s30-browser-files
  - **Test-supporting material was shared with Cure53**
  - **All relevant sources were shared with Cure53**

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Identified Vulnerabilities

The following section lists all vulnerabilities and implementation issues identified during the testing period. Notably, findings are cited in chronological order rather than by degree of impact, with the severity rank offered in brackets following the title heading for each vulnerability. Furthermore, all tickets are given a unique identifier (e.g., *TTP-02-001*) to facilitate any future follow-up correspondence.

TTP-02-003 WP1: Data URI allows JS execution despite *safest* security level *(Low)*

The Tor Browser incorporates a feature that makes it possible for users to adjust the security level of the browser. This feature is designed to mitigate web-based attacks targeting the users' security and anonymity, doing so by disabling certain web features.

It has been discovered that the proper security level protections are not applied when there is a top-level navigation to a data URI for the first time while being in the *safest* state. However, when the same data URI is reloaded, the correct protections are applied. This issue could potentially be used by attackers to trick users into executing JavaScript despite the *safest* security level set. Given it is not possible to automatically open or redirect the top-level page to a data URI, user-interaction is required to exploit this vulnerability. As a result, the severity of the issue has been downgraded to *Low*.

PoC:

```
<html>
<body><br><br><center>
<h2>
        <a href="data:text/html,<script>alert(`Javascript
        execution :)`);</script>">Click on "Copy Link" and then
        "Paste and Go" in the address bar!</a>
</h2>
</center></body>
</html>
```

Steps to reproduce:

1. Open the Tor Browser and connect to it.
2. Set the security level to *safest*.
3. Save the PoC above as an HTML file and open it on the Tor Browser.
4. Right-click on the link and select *Copy Link*.
5. Right-click on the address bar and click on *Paste and Go*.
6. Observe an alert being displayed, which demonstrates that JavaScript was executed, even though the browser has the *safest* security level set.

To mitigate this issue, Cure53 advises setting the protections related to the security levels on data URIs as well.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

TTP-02-004 OOS: No user-activation required to download files *(Low)*

When investigating the new security warnings related to the downloaded files, Cure53 observed that there were no safeguards in place to restrict the number of files that could be downloaded by a single webpage.

This issue is a result of the *browser.download.enable_spam_prevention* flag being set to *false*. As a result, malicious pages can download an unlimited number of arbitrary files to the user's *Downloads* folder without any user-interaction. Additionally, this could also lead to DoS attacks on the user's browser. The approach would entail continuous downloads of files.

PoC:

```
<html>
    <body>
        <script>
            onload = () => {
                let counter = 0;
                let timer = setInterval(() => {
                    if (counter == 50)
                        clearInterval(timer);
                    download.click();
                    counter++;
                }, 1);
            }
        </script>
        <a id="download" href="data:text/html,1337" download="poc.html"></a>
    </body>
</html>
```

Steps to reproduce:

1. Open the Tor Browser and connect to it.
2. Save the PoC above as an HTML file and open it on the Tor Browser.
3. Observe that fifty downloads will start automatically and be dropped in the default *Downloads* folder.

To mitigate this issue, Cure53 advises changing the value of the *browser.download.enable_spam_prevention* flag to *true*. The change will prevent malicious pages from automatically downloading multiple files. Instead, it will require the user to actively grant permission to the webpage, if they wish to initiate multiple downloads.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

TTP-02-006 WP1: Information leaks via custom homepage *(Low)*

It was discovered that setting a custom homepage can lead to information leaks under specific circumstances, specifically when malicious approaches are combined with using the *Reset your Identity* feature. Specifically, when a user has their custom homepage opened in a browser tab and then decides to use the *Reset their identity* feature, the homepage will automatically open again after the browser restarts with the *new identity*. If the custom homepage is malicious, it could track the moment the user left the page and infer that the new user who shortly accessed their page is the same as the previous user.

Furthermore, a malicious webpage could use the *onbeforeunload* function to determine with confidence whether the user initiated an *identity reset*. If the user tried to close the browser or navigate away, the *onbeforeunload* dialog would be displayed and block further actions, giving enough time for the script to ping the server. In contrast, if the user chose to *reset their identity*, the browser would be automatically closed, and no ping would be sent. The PoC below demonstrates how the above sequence could be achieved. Additional steps to track when the user left and rejoined the page would have to be added to properly infer the user's *new identity*.

PoC:

```
<script>
  let exit;
  onbeforeunload = () => { exit = true; return ""; }
  let timer = setInterval(()=>{
    if (exit) {
      let img = new Image();
      img.src = "/exited";
      clearInterval(timer);
      timer = false;
    }
  }, 1);
</script>
```

Steps to reproduce:

1. Open the Tor Browser and connect to it.
2. Save the PoC above as an HTML file and open it on the Tor Browser.
3. Observe a request made to */exited* if the user tried to close the browser or navigate away from the tab. See that the data will be handled differently if the user tries to *reset their identity*.

To mitigate this issue, Cure53 advises removing the ability to set custom homepages from the options available to users. Alternatively, the custom homepage should not be opened automatically upon usage of the *Reset your Identity* feature.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Miscellaneous Issues

This section covers any and all noteworthy findings that did not incur an exploit but may assist an attacker in successfully achieving malicious objectives in the future. Most of these results are vulnerable code snippets that did not provide an easy method by which to be called. Conclusively, whilst a vulnerability is present, an exploit may not always be possible.

TTP-02-001 WP1: XSS in TorConnect's captive portal *(Info)*

TorConnect's captive portal performs a redirect to a URL that is retrieved from the *redirect* parameter located in the *query* string. No validations are performed to guarantee that the scheme of the URL is valid before having it used in the redirection. Note that the next step is performed after the user successfully connects to TOR.

Fortunately, arbitrary JavaScript execution is prevented due to the strict CSP policy that is applied to the *about:torconnect* page. Hence, the severity has appropriately been set at *Info* only.

Affected file:
browser/components/torconnect/content/aboutTorConnect.js

Affected code:

```
async init() {
  // see if a user has a final destination after bootstrapping
  let params = new URLSearchParams(new URL(document.location.href).search);
  if (params.has("redirect")) {
    const encodedRedirect = params.get("redirect");
    this.redirect = decodeURIComponent(encodedRedirect);
  } else {
    // if the user gets here manually or via the button in the urlbar
    // then we will redirect to about:tor
    this.redirect = "about:tor";
  }[...]
}
```

Steps to reproduce:

1. Open the Tor Browser and access *about:torconnect?redirect=javascript:alert(document.domain);*
2. Click on *Connect* and check the DevTools to verify that JavaScript execution was prevented by CSP.

To mitigate this issue, Cure53 advises validating the scheme of the URL from the *redirect* parameter, and verifying it against an allow-list of safe schemes.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

TTP-02-002 WP1: Redirect prevents switching to new Tor Circuit *(Info)*

It was discovered that navigation initiated through the new Tor Circuit feature can be hijacked. This can be accomplished by redirecting the current website to a cached page immediately after the Tor Circuit switch starts. As a result, the attacker-initiated navigation occurs before the Tor Circuit's browser-initiated navigation and, subsequently, the next step is canceled.

An attacker could exploit this vulnerability to prevent users from switching circuits while browsing a malicious webpage. Although this prevents the user from changing their Tor Circuit, it was concluded that this does not pose any immediate security risk, and as such, the severity mark was appropriately set at *Info*.

PoC:

```
<?php header("cache-control: max-age=604800");
      header("Age: 100"); ?>
<html>
    <script>
        let status = false;
        onbeforeunload = () => {
            status = true;
        }
        let timer = setInterval(() => {
            if (status) {
                status = false;
                clearInterval(timer);
                location.href = location.href;
        } }, 1);
    </script>
</html>
```

Steps to reproduce:

1. Open the Tor Browser and connect to it.
2. Save the PoC above as a PHP file and serve it through a PHP server.
3. Access the file a few times through the Tor Browser to make sure it gets cached by the browser.
4. Click on the *Tor Circuit* button and then on the *New Tor circuit for this site* option.
5. The page will quickly be reloaded but the Circuit will remain the same.

To mitigate this issue, Cure53 advises forcing the navigation initiated by the new *Tor Circuit* feature to be completed. Cancellation of a user-initiated navigation is ill-advised in this scenario. However, during the testing phase, the team was unable to pinpoint the specific code responsible for this issue. As a result, the mitigation advice provided is currently incomplete.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

TTP-02-005 WP1: Redirect to *about:blank* hides the *new Tor Circuit* button *(Info)*

It is possible to hide the *Tor Circuit* button from the address bar for a given tab by listening to the *onbeforeunload* event and redirecting the page to *about:blank* when the event is triggered.

If a user attempts to reset their identity by clicking on the *New Tor circuit for this site* option, the navigation can be hijacked by the attacker's script. A blank page will be displayed as a consequence. If the user attempts to navigate back to the previous page using the *Back* button, the *Tor Circuit* button will not be displayed in the address bar.

Similarly to TTP-02-002, this issue was found not to pose any immediate security risk and is included as *Info* only.

PoC:

```
<script>
    let status;
    onbeforeunload = () => {
        status = true;
    }
    let timer = setInterval(() => {
        if (status) {
            status = false;
            clearInterval(timer);
            location = "about:blank";
        }
    }, 1);
</script>
```

Steps to reproduce:

1. Open the Tor Browser and connect to it.
2. Save the PoC above as an HTML file and open it in the browser.
3. Click on the *Tor Circuit* button and then on the *New Tor circuit for this site* option.
4. The page will be redirected to *about:blank*.
5. Click on the *Back* option and observe that the *Tor Circuit* button is hidden for this page.

To mitigate this issue, Cure53 advises applying the same mitigation as specified in the TTP-02-002 ticket. Given these issues seem to be related and they might share the same root cause, it is recommended to consider and address them together.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

TTP-02-007 WP1: Missing *about:* pages in *shouldShowTorConnect* check *(Info)*

It was discovered that the *about:welcome, about:privatebrowsing*, and *about:home* pages are not redirecting to *about:tor* when they are accessed by a user who has not connected to Tor yet.

While this behavior does not present any immediate security risk, it can potentially cause confusion or alarm users who may access these pages before being connected to the Tor network. To ensure consistency across all *about:* pages, it is recommended to deploy relevant changes.

Affected file:
browser/base/content/utilityOverlay.js

Affected code:

```
if (TorConnect.shouldShowTorConnect) {
    if (
        url === "about:tor" ||
        (url === "about:newtab" &&
        Services.prefs.getBoolPref("browser.newtabpage.enabled", false))
    ) {
        url = TorConnect.getRedirectURL(url);
    }
}
```

In order to reproduce this issue, simply open the Tor Browser, access *about:home* and note that the page does not perform an automated redirection to *about:tor*.

To mitigate the problem, Cure53 advises including additional checks to validate whether the URL matches *about:welcome*, *about:privatebrowsing* or *about:home*. If a match is found, the page should be redirected to *about:tor*.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Conclusions

In light of the very few minor discoveries, Cure53 concludes that the Tor Browser UI boasts a strong security premise. Despite dedicated searches for compromisable items and routes, three members of the Cure53 team were unable to document any major flaws. The issues stemming from this April 2023 examination propose some minor changes, which could be considered by the Tor Browser team to further enhance the overall standing of the complex in general, and the UI in particular.

To facilitate this *TTP-02* inspection, the client provided a detailed list of commits and file review lists. This happened before the beginning of the security assessment, and later helped to identify critical areas of interest and sharpened the scope definition. Proving extremely valuable, the material made it possible for the testing team to swiftly familiarize themselves with all relevant features. Thus, the testers' efforts were accordingly planned and focused.

In order to properly understand the newly implemented features, the testing team conducted a comprehensive review of the relevant documentation and an extensive examination of the code changes pertaining to the UI. The analysis confirmed that the code is of a high quality, with key functionalities appropriately commented on for clarity. The codebase was written to a first-rate standard and evidently conformed to secure coding practices. The majority of the code reviewed consisted of HTML, JavaScript, and CSS files. Concerning the recent UI changes, a code review and deep-dive analysis to determine any potential for client-side vulnerabilities was performed. The testers looked for *postMessage* issues, prototype-pollution, DOM XSS sinks and similar input-manipulation issues. This led to the discovery of an XSS problem in TorConnect's captive portal. Fortunately, a strict CSP prevented exploitation of this vulnerability (see TTP-02-001).

During the assessment of the *security-level* component and its integration with the NoScript extension, several attempts were made to bypass the restrictions employed by the feature. In the end, Cure53 demonstrated an arbitrary JavaScript execution through the use of a Data URI. This issue, noted in TTP-02-003, occurred even when setting the *security level* to the *safest* level.

As for the new Tor Circuit feature, two issues were uncovered where a malicious website can prevent users from switching circuits, or even hide the Tor Circuit button from the address bar. These flaws are being tracked in tickets TTP-02-002 and TTP-02-005, respectively.

Dr.-Ing. Mario Heiderich, Cure53
Bielefelder Str. 14
D 10709 Berlin
cure53.de · mario@cure53.de

Furthermore, it was observed that several changes were made to align the Tor Browser with the current behavior of Mozilla Firefox with regard to handling of downloads. Specifically, a new security warning was added for the downloaded files to enhance user awareness and prevent inadvertent execution of malicious content. However, a specific configuration flag was found to be set to *false*, which meant that an attacker could download an unlimited number of files without any user-interaction. This issue has been documented in ticket TTP-02-004.

Another issue was discovered in the usage of a custom homepage in conjunction with the process of identity resetting. Specific circumstances could foster information leaks, with an example of a malicious page having the capacity to track users across restarts (see TTP-02-006). An inconsistency was also found in the way redirects to *about:tor* happen when accessing certain *home:* pages. As TTP-02-007 explains, this could confuse and alarm users.

Additional tests were conducted to identify HTTP leaks within data URIs, but all network connections were blocked as expected, thus indicating that the system functions as designed. Other less common features such as PDF and HTML file-saving were tested, but no issues were identified during these assessments.

Further tests aimed at injection attacks were conducted in the "*.onion available*" feature. Various payloads were utilized in the *Onion-Location* header, but it was determined that proper scheme validations were being correctly applied. In other words, Cure53 can confirm the security and integrity of this feature.

In conclusion, following the completion of this security audit, Cure53 garnered an unanimously strong impression regarding the security premise deployed by the Tor Browser. This positive verdict specifically extends to the recent UI changes, which were not found to disrupt correctness of the protection mechanisms on offer. This viewpoint is further corroborated by the small number of *Low* and *Info* reported during this April 2023 project. The TTP team has been successful in averting attacks and sufficiently safeguarding the Browser complex.

Cure53 would like to thank Gaba from The Tor Project team for their excellent project coordination, support and assistance, both before and during this assignment.