Fine penetration tests for fine websites

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

Pentest-Report ExpressVPN Browser Extension 09.-10.2022

Cure53, Dr.-Ing. M. Heiderich, M. Kinugawa, M. Pedhapati

Index

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

Introduction

“Go online safely and securely in 2022 with strong VPN encryption. Spoof your location and control the ExpressVPN app from Chrome.”

From https://www.expressvpn.com/vpn-software/chrome-vpn

This report - titled EXP-12 - details the scope, results, and conclusory summaries of a source-code-assisted penetration test and audit against the ExpressVPN browser extension. The work was requested by ExpressVPN in August 2022 and initiated by Cure53 in September and October 2022, namely in CW39 and CW40. A total of six days were invested to reach the coverage expected for this project. All work conducted for this review comprised one sole work package (WP), as follows:

- **WP1**: Source-code-assisted penetration tests against ExpressVPN browser extension

In context, this test marks the twelfth collaborative engagement between ExpressVPN and Cure53. The VPN browser extension has previously been reviewed, but has undergone a number of changes since that assessment. In addition, the Keys browser extension covered in a supplementary report has not yet been subject to examination in any of the previous audits.

Cure53 was provided with sources, builds, test-user accounts, as well as any alternative means of access required to ensure a smooth audit completion. For this purpose, the methodology chosen was white box and a team of three senior testers was assigned to the project’s preparation, execution, and finalization. All preparatory actions were completed in September 2022, namely in CW38, to ensure that testing could proceed without hindrance or delay.

Communications were facilitated via a dedicated, shared Slack channel deployed to combine the workspaces of ExpressVPN and Cure53, thereby creating an optimal collaborative working environment. All participatory personnel from both parties were invited to partake throughout the test preparations and discussions.

In light of this, communications proceeded smoothly on the whole. The scope was well-prepared and transparent, no noteworthy roadblocks were encountered throughout testing, and cross-team queries remained minimal as a result. The ExpressVPN team delivered excellent test preparation and assisted the Cure53 team in every respect to procure maximum coverage and depth levels for this exercise.

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

Cure53 gave frequent status updates concerning the test and any related findings, whilst simultaneously offering prompt queries and receiving efficient, effective answers from the maintainers. Live reporting was offered and subsequently conducted via the aforementioned Slack channel.

Regarding the findings, the Cure53 team achieved comprehensive coverage over the single scope item, detecting a total of two. Both of these findings were categorized as security vulnerabilities, whilst no general weaknesses with lower exploitation potential were identified.

Evidently, the overall yield of findings is considerably minimal, which should be considered a positive indication of the extension's security strength. Once the ExpressVPN team has addressed and mitigated the issues documented in this report, Cure53 would be pleased to confirm that the extension in scope is sufficiently safeguarded and ready for production.

The report will now shed more light on the scope and testing setup as well as provide a comprehensive breakdown of the available materials. Subsequently, the report will list all findings identified in chronological order, starting with the detected vulnerabilities and followed by the general weaknesses unearthed. Each finding will be accompanied by a technical description and Proof of Concepts (PoCs) where applicable, plus any relevant mitigatory or preventative advice to action.

In summation, the report will finalize with a conclusion in which the Cure53 team will elaborate on the impressions gained toward the general security posture of the ExpressVPN browser extension, giving high-level hardening advice where applicable.

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

Scope

- **Source code audits and security assessments against ExpressVPN browser** extension
  - **WP1**: Source-code-assisted penetration tests against ExpressVPN browser extension
    - **Primary audit focus:**
      - ExpressVPN VPN browser extension
      - Tested version: 5.1.2
    - **In-scope items:**
      - Clickjacking vectors
      - Confidentiality of client IP address
      - Frontend UI
  - **All relevant binaries in scope were shared with Cure53**
  - **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

Severity Glossary

The following section details the varying severity levels assigned to the issues discovered in this report.

Critical: The highest possible severity level. Categorizes issues that allow attackers to achieve extensive access to sensitive areas, such as critical systems, applications, data or other pertinent components in scope.

High: Categorizes issues that allow attackers to achieve limited access to sensitive areas in scope. This also includes issues with limited exploitability that can facilitate a significant impact upon the target in scope.

Medium: Categorizes issues that do not incur major impact on the areas in scope. Additionally, issues requiring a more limited exploitation are graded as *Medium*.

Low: Categorizes issues that have a highly limited impact on the areas in scope. Mostly does not depend on the level of exploitation but rather on the minor severity of obtainable information or lower grade of damage targeting the areas in scope.

Info: Categorizes issues considered merely informational in nature. They are mostly considered as hardening recommendations or improvements that can generally enhance the security posture of the areas in scope.

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 throughout the testing period. Please note that findings are listed in chronological order rather than by their degree of severity and impact. The aforementioned severity rank is given in brackets following the title heading for each vulnerability. Furthermore, each vulnerability is given a unique identifier (e.g., *EXP-12-001*) to facilitate any future follow-up correspondence.

EXP-12-001 WP1: VPN disconnection via clickjacking in *networkLock.html* (High)

Fix Note: *The issue was addressed by the ExpressVPN team and the fix was verified by Cure53 who were able to review the related diff & PR. The issue no longer exists.*

The observation was made that the *networkLock.html* page listed in the *web_accessible_resources* manifest property and embeddable via any web page uses the URL specified in the *url* parameter as a redirect or link destination. Here, testing confirmed that this process lacks URL validation and an arbitrary URL can be set as the destination.

Fortunately, the JavaScript execution via *javascript: URLs* is blocked by the extension's Content Security Policy. However, this behavior still facilitates attack potential. Specifically, the specifiable URLs include the *popup.html* page, which offers a button to toggle the VPN on and off. By specifying this URL to the *url* parameter and embedding the page into an iframe, the VPN connection can be disconnected via clickjacking attacks, which would expose the user's real IP address via the network request sent following disconnection.

The following PoC demonstrates the method by which the VPN can be disconnected via clickjacking attacks, with a semi-transparent iframe overlaid on the button tag. In the eventuality *Double-click here* is clicked twice, the VPN will be disconnected. The first click will cause the navigation to the *popup.html* page via the anchor tag, and the second click will disconnect the VPN via the on/off button click.

Notably, the extension's UI can be completely transparent by setting the *opacity* CSS property to 0. The following PoC sets the property to 0.3 for clarity; for context, this issue was tested using Google Chrome.

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

PoC:

```
<iframe
src="chrome-extension://fgddmllnllkalaagkghckoinaemmogpe/html/networkLock.html?
url=chrome-extension://fgddmllnllkalaagkghckoinaemmogpe/html/popup.html"
style="width:250px;height:250px;position:absolute;top:0;left:0;opacity:0.3"></
iframe>
<button   style="position:absolute;top:205px;left:120px;z-index:-1">Double-click
here</div>
```

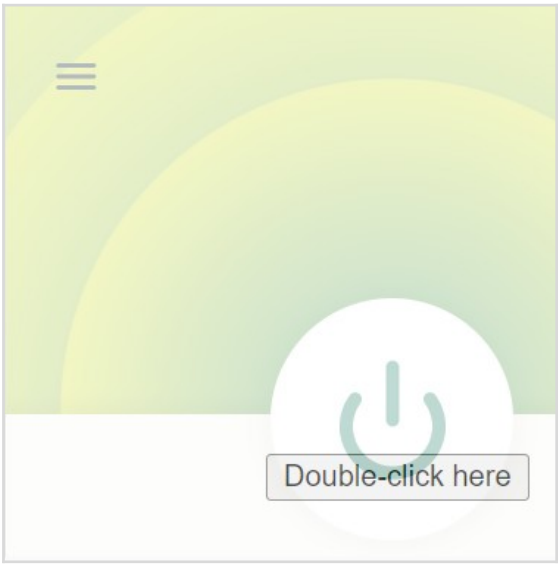

Fig.: Clickjacking attack example.

Affected file:

expressvpn-source/xv_chrome/source/components/networkLock.vue

Affected code:

```
<a :href="urlParams.get('url')">{{ urlParams.get('url') }}</a>
[...]
methods: {
    openWebsite() {
      window.setTimeout(() => {
        window.location.replace(this.urlParams.get('url'));
      }, 500); // give it enough time for the lock to disengage if needed
    },[...]
},
```

To mitigate this issue, Cure53 advises setting only URLs with the *http(s):* protocol as the redirect or link destination.

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

This issue was assigned a **7.1** rating, as stipulated in the breakdown of each scoring component offered below:

https://www.first.org/cvss/calculator/3.1#CVSS:3.1/AV:N/AC:L/PR:N/UI:R/S:U/C:H/I:L/A:N

CWE: https://cwe.mitre.org/data/definitions/20.html

EXP-12-008 WP1: WebRTC blocking feature improperly configured on Firefox *(Info)*

Fix Note: *The issue was addressed by the ExpressVPN team and the fix was verified by Cure53 who were able to review the related diff & PR. The issue no longer exists.*

Testing confirmed that the WebRTC Block feature, which disables WebRTC and purportedly prevents IP leaks, does not function as expected in Firefox and actually facilitates IP address leakage (although, this is the VPN server IP address), unlike its Chrome counterpart. The following snippets highlight the affected source code whereby only *webRTCIPHandlingPolicy* is set to *disable_non_proxied_udp*, but in Firefox *privacy.network.peerConnectionEnabled.set* should also be set to *False* to prevent IP leakage.

Affected file:
sources/xv_chrome/source/scripts/background.js

Affected code:

```
function setWebrtcOption() {
    let enableStates = ['connecting', 'reconnecting', 'connected',
'disconnecting', 'connection_error'];
    // checking if the current browser supports chrome.privacy
    if (typeof chrome.privacy === 'undefined') {
      return;
    }
    // update webRTCIPHandlingPolicy setting
    // first get the setting access permission and usage details
    chrome.privacy.network.webRTCIPHandlingPolicy.get({}, (details) => {
      // check if we can modify it
      if ((details) && ((details.levelOfControl ===
'controllable_by_this_extension') || (details.levelOfControl ===
'controlled_by_this_extension'))) {
        let isEnabled = (prefs['chrome.prevent_webrtc_leaks'] &&
       (enableStates.includes(currentInfo.state)));
        let settingValue = isEnabled ? 'disable_non_proxied_udp' : 'default';
        chrome.privacy.network.webRTCIPHandlingPolicy.set({
            value: settingValue }, () => {
          // check if we managed to change the setting successfuly.
          // Not sure when this can ever fail if we passed the
```

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

```
            'controllable_by_this_extension' check
          if (chrome.runtime.lastError !== undefined) {
            // console.error(chrome.runtime.lastError);
          }
        });
      }
    });
  }
```

PoC:
https://pwn.af/c53/ex/webrtc.html

To mitigate this issue, Cure53 recommends setting *privacy.network.peerConnectionEnabled.set*[1] to *False* in Firefox to prevent IP address leakage.

This issue was assigned a **3.1** rating, as stipulated in the breakdown of each scoring component offered below:

https://www.first.org/cvss/calculator/3.1#CVSS:3.1/AV:N/AC:H/PR:N/UI:R/S:U/C:L/I:N/A:N

CWE: https://cwe.mitre.org/data/definitions/200.html

[1] https://www.expressvpn.com/webrtc-leak-test#:~:text=disable%20WebRTC%20in-,Firefox,-on%20

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

Conclusions

The impressions gained during this report - which details and extrapolates on all findings identified during the CW39 and CW40 testing against the ExpressVPN browser extension by the Cure53 team - will now be discussed at length. To summarize, the confirmation can be made that the components under scrutiny have garnered a positive impression, with only a couple of (albeit significant) findings to report.

In context, communication was achieved via a shared Slack channel, cross-team queries regarding certain findings and functionality were promptly answered, and the engineering team provided immediate assistance to the testing team when required. The testing team achieved strong coverage of the single WP in scope, particularly in relation to the following assessment areas:

- Firstly, the extension's Content Security Policy configuration was subject to rigorous evaluation. Positively, no bypasses were identified that could potentially facilitate JavaScript execution.

- The resources listed on the *web_accessible_resources* manifest property were also deep-dive assessed. In this regard, an issue was discovered that could disconnect the VPN via clickjacking, as detailed in ticket EXP-12-001. Even though some degree of user interaction is required to successfully instigate this behavior, this still represents a significant flaw for a privacy-sensitive VPN application and should be addressed with utmost priority at the earliest possible convenience.

- Elsewhere, the content scripts were also subject to investigation. Particular scrutiny was placed on a selection of potential risk scenarios, including XSS, issues incurred via DOM clobbering, and Content Security Policy bypasses on the loaded web pages.

- In general, the Express VPN browser extension was assessed by the testing team to locate any client-side-related security issues associated with XSS, *postMessage*, and prototype pollution. In light of this, the testing team noted that the majority of the frontend utilizes the VueJS framework, which offers a battle-tested escaping mechanism that prevents a plethora of XSS issues by default.

- Finally, all privacy and security features offered by the ExpressVPN browser extension were examined to determine the presence of any bypass opportunities. This led to the detection of a minor issue in the Block WebRTC feature, as documented in ticket EXP-12-008.

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

All in all, following the completion of this audit, the testing team can only conclude that the ExpressVPN team has already established a strong security foundation for the browser extension in scope. Nevertheless, Cure53 recommends addressing and resolving the two identified issues to ensure comprehensively secure extension usage for all users.

Cure53 would like to thank Harsh S. and Brian Schirmacher from the ExpressVPN team for their excellent project coordination, support and assistance, both before and during this assignment.