Fine penetration tests for fine websites

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

Pentest-Report VeePN Browser Extensions 03.2021

Cure53, Dr.-Ing. M. Heiderich, Dipl.-Ing. A. Inführ

Index

Introduction

> *“2500+ servers and 50 locations that VeePN covers ensure exclusive connection speed. Now you can effortlessly surf the internet and not worry about download caps or bandwidth limits. Our priority is providing our customers with the fastest internet experience possible.”*
>
> From https://veepn.com/

This report describes the results of a security assessment targeting the VeePN Browser Extensions for Chrome and Firefox. Carried out by Cure53 in the spring of 2021, the project entailed a thorough penetration test and a dedicated audit of the VeePN sources.

To offer some contextual details, the work was requested by VeePN Corp. in February 2021 and then promptly scheduled. As for the resources, a team of two senior testers was created in connection to the skills needed for the project. They were responsible for the project’s preparation, execution and finalization, with the core testing phase taking place in late March and early April 2021, namely in CW13. The budget stood at five person-days.

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

In order to optimally structure the work, three work packages (WPs) were delineated as follows:

- **WP1**: Penetration-Tests & Source Code Audits against VeePN Chrome Extension
- **WP2**: Penetration-Tests & Source Code Audits against VeePN FF Extension

The methodology chosen for this assessment was a white-box approach, so as to enable the best possible breadth and depth of coverage. Cure53 was given access to the uncompressed sources of the two extensions as well as all other necessary info, test user-accounts, builds and similar material.

All preparations were done in the week before the tests started, namely in CW12, so as to foster a smooth start for the Cure53 testing team. The project moved forward at a good pace. Communications were done in a dedicated and shared Slack channel which connected the workspaces of Cure53 and VeePN. Test discussions, questions and status updates were relayed efficiently; no noteworthy roadblocks were encountered during the test.

The Cure53 team managed to get very good coverage over the WP1-2 scope items. Only three security-relevant discoveries were made: two classified to be security vulnerabilities and one being just a general weakness with lower exploitation potential. One of the findings was given a *High* score, as it entailed a classic information leak in the Squid proxy error page. This is a very common finding for VPN and proxy software setups.

Other than that, no issues of noteworthy severity were spotted and the absence of *Critical* problems is a good sign. Moreover, all findings were discussed on Slack, alongside their fixes produced by the VeePN team during the testing period. It is important to note that Cure53 managed to verify the implemented mitigation before the finalization of this report.

In the following sections, the report will first shed light on the scope and key test parameters, as well as the structure and content of the WPs. Next, all findings will be discussed in grouped vulnerability and miscellaneous categories, then following a chronological order in the former category. Alongside technical descriptions, PoC and mitigation advice are supplied when applicable. Finally, the report will close with broader conclusions about this 2021 project. Cure53 elaborates on the general impressions and issues a verdict based on the testing team's observations and collected evidence. Tailored hardening recommendations for the VeePN Browser Extensions for Chrome and Firefox are also incorporated into the final section.

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

Scope

- **White-Box Penetration-Tests & Code Audits against VeePN Browser Extensions**
  - **WP1**: Penetration-Tests & Code Audits against VeePN Chrome Extension
    - All relevant Sources have been shared
  - **WP2**: Penetration-Tests & Code Audits against VeePN Firefox Extension
    - All relevant Sources have been shared
  - **Test-users were created for Cure53**
    - **Trial accounts:**
      - U: cure53-trial-1@gmail.com
      - U: cure53-trial-2@gmail.com
      - U: cure53-trial-3@gmail.com
    - **Premium accounts:**
      - U: cure53-premium-1@gmail.com
      - U: cure53-premium-2@gmail.com
      - U: cure53-premium-3@gmail.com

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

Identified Vulnerabilities

The following sections list both vulnerabilities and implementation issues spotted during the testing period. Note that findings are listed in chronological order rather than by their degree of severity and impact. The aforementioned severity rank is simply given in brackets following the title heading for each vulnerability. Each vulnerability is additionally given a unique identifier (e.g. *VEE-01-001*) for the purpose of facilitating any future follow-up correspondence.

VEE-01-001 General: User-information leaked in Squid default error page *(High)*

Note: *This issue was addressed and fixed during the assessment.*

During the assessment it was discovered that the utilized Squid proxy software displays the current user IP address, as well as the proxy authentication credentials, in case of a connection error. A malicious website can abuse this behavior to attack VeePN users by loading an HTML resource in an iframe, which is rejected by the attacker-controlled web server. Therefore, the aforementioned Squid error page is displayed instead and HTML content can be accessed by the attacker's JavaScript code as it is considered the same origin.

Example URL:
http://cure53.de:81/

Squid error page:

```
<div id="content">
<p>The following error was encountered while trying to retrieve the URL: <a
href="http://cure53.de:81/">http://cure53.de:81/</a></p>
[...]
<p>Your cache administrator is <a href="mailto:webmaster?subject=CacheErrorInfo
%20-%20ERR_CONNECT_FAIL&amp;body=CacheHost%3A%20container-31091%0D%0AErrPage%3A
%20ERR_CONNECT_FAIL%0D%0AErr%3A%20(111)%20Connection%20refused%0D%0ATimeStamp%3A
%20Tue,%2030%20Mar%202021%2007%3A26%3A28%20GMT%0D%0A%0D%0AClientIP%3A
%2084.112.217.173%0D%0AServerIP%3A%20cure53.de%0D%0A%0D%0AHTTP%20Request%3A%0D
%0AGET%20%2F%20HTTP%2F1.1%0AProxy-Connection%3A%20keep-alive%0D%0APragma%3A
%20no-cache%0D%0ACache-Control%3A%20no-cache%0D%0AProxy-Authorization%3A%20Basic
%20bnF4c2drd3Vzem53cmJkdzg0cWd3eXhmd3Bjdnd5YjQ6Y3VyZTUzLXByZW1pdW0tMUBnbWFpbC5jb
20%3D%0D%0AUpgrade-Insecure-Requests%3A%201%0D%0AUser-Agent%3A%20Mozilla
%2F5.0%20(X11%3B%20Linux%20x86_64)%20AppleWebKit%2F537.36%20(KHTML,%20like
%20Gecko)%20Chrome%2F89.0.4389.90%20Safari%2F537.36%0D%0AAccept%3A
%20text[...]US,en%3Bq%3D0.9%0D%0AHost%3A%20cure53.de%3A81%0D%0A%0D%0A%0D
%0A">webmaster</a>.</p>
```

It is recommended to modify the generic Squid error page and simply remove all user-related information.

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

VEE-01-003 WP1: Auto-Protect feature bypass via domain trimming *(Low)*

Note: *This issue was addressed and fixed during the assessment.*

The Google Chrome WebExtension allows its users to add domains which are always tunneled through the proxy tunnel. It was discovered that the extension removes the substring "*www.*" from the specified domain-name. This could lead to a user trusting that a potentially malicious domain is being tunneled despite a completely different domain being added by the extension.

Domain Example:
evilwww.abc.com

Trimmed URL:
evilabc.com

Affected File:
js/util/protectlist.js

Affected Code:

```
addItem(newUrl, newFavicon, restartProxy = true) {
[...]
const trimUrl = (longUrl = '') => {
const domain = longUrl.replace('http://', '').replace('https://',
'').replace('www.', '').split(/[/?#]/)[0];
return domain;
};
```

It is recommended to simply remove the highlighted code path above. This ensures that the WebExtension tunnels the domain, which was actually added by the user.

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

Miscellaneous Issues

This section covers those noteworthy findings that did not lead to an exploit but might aid an attacker in achieving their malicious goals in the future. Most of these results are vulnerable code snippets that did not provide an easy way to be called. Conclusively, while a vulnerability is present, an exploit might not always be possible.

VEE-01-002 WebExtension: XSS in pop-ups via server status code *(Low)*

Note: *This issue was addressed and fixed during the assessment.*

While checking the source code of the WebExtension for potential XSS sinks, multiple *innerHTML* assignments were investigated. In one instance, it was discovered that in case an unknown server error is encountered, the status text is included in the DOM via *innerHTML*, therefore potentially allowing the inclusion of HTML tags. It must be noted that the likelihood of this sink being exploitable to target specific VeePN users is very slim.

Affected Files:

- *js/eventhandler/templates/login/onSubmit.js*
- *js/eventhandler/templates/register/onSubmit.js*
- *js/eventhandler/templates/forgot-password/onSubmit.js*

Affected Code:

```
else {
const tParams = { statusLine: `${status} ${statusText}` };
errorDiv.innerHTML = t('UnexpectedServerResponse', tParams);
}
break;
```

Nevertheless it is recommended to replace the *innerHTML* property with a secure variant like *innerText*. This would allow to display the error to the user without risking the rendering of unintended HTML tags.

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

Conclusions

The overall impression gained of the Firefox and Google Chrome VeePN WebExtension is very positive. The Cure53's impressions - gathered by two senior testers over the course of five-days dedicated to the project in spring 2021 - hinge upon the low number of findings. In addition, the general verdict has to take into account that any issue reported via Slack was immediately addressed by the VeePN team. Since the fixes have been verified, Cure53 can only conclude this project with excellent outcomes for the VeePN complex. The following notes will describe the security observations in more detail.

Cure53 assessed the available attack surface in regard to potential injection vulnerabilities like XSS. Generally speaking, the code is very careful to avoid inclusions of user-controlled data. The *innerHTML* property is solely used with static strings, except for one minor debug call documented in VEE-01-002. Similarly, Cure53 neither spotted exploitable *postMessage* handlers, nor found help or error pages directly including a potentially user-controlled URL parameter in their DOM. The WebExtensions do not listen for any messages sent from other extensions either.

The next focal area concerned the proxy setup and the potential for bypasses or user-related information leaks. The establishment of the proxy connection, as well as the defined exception rules for private domains and IPs, was checked carefully for Firefox and Chrome, respectively. The code was especially studied for the presence of common mistakes in wildcard domains or local IPs checks, as these could allow bypasses of the configured proxy server. Despite extensive testing, no issue was discovered in this regard.

It must be noted that VeePN strengthened the security of the code by not determining whether a specified domain resolves to a local IP, which often introduces bypasses. Similarly, the supported Browser protocols and features like DNS prefetching were tested in the context of a running proxy-tunnel. All test-cases were correctly handled by the WebExtensions. However, the testers revealed that a configuration issue in the utilized Squid proxy leaked the current user's real IP address (see VEE-01-001).

Lastly, additional security features regarding WebRTC or Chrome's "Auto-Protect" were assessed. The VeePN WebExtension correctly disables WebRTC, which completely blocks the possibility to abuse this technology for leaking the actual IP addresses of users. In the end, only a minor implementation issue was spotted in the handling of user-submitted domains (see VEE-01-003).

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

All in all, the VeePN Browser Extension is on the right track regarding its security design. Common browser proxy mistakes have been successfully avoided by good design and implementation decisions, such as keeping the logic of excluded domains and IPs simple, properly using the available browser objects and functions to establish the proxy connection, as well as disabling potentially risky browser features. As long as VeePN continues on this path with careful development of new features, the WebExtensions can be seen as benefiting from a strong security model.

Cure53 would like to thank Anna Henderson, Dana Lee and Sergio Burret from the VeePN Corp. team for their excellent project coordination, support and assistance, both before and during this assignment.