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 macOS Client 06.-07.2022

Cure53, Dr.-Ing. M. Heiderich, MSc. S. Moritz, BSc. F. Fäßler, MSc. J. Moritz

Index

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

Introduction

"A VPN (virtual private network) is the easiest and most effective way for people to protect their internet traffic and keep their identities private online. As you connect to a secure VPN server, your internet traffic goes through an encrypted tunnel that nobody can see into, including hackers, governments, and your internet service provider."

From https://www.expressvpn.com/what-is-vpn

This report - entitled EXP-08 - details the scope, results, and conclusory summaries of a penetration test and source code audit against the ExpressVPN macOS client binary and codebase. The work was requested by ExpressVPN in June 2022 and initiated by Cure53 in late June and early July 2022, namely in CW26 and CW27. A total of twenty-five days were invested to reach the coverage expected for this project.

The testing conducted for EXP-08 was divided into two separate work packages (WPs) for execution efficiency, as follows:

- **WP1**: Source-code-assisted penetration tests against ExpressVPN macOS client binary
- **WP2**: Source-code audits and reviews against ExpressVPN macOS client codebase

In context, Cure53 has already audited the ExpressVPN macOS client during a previous test engagement held in June 2020 and documented under report EXP-03. This audit, therefore, marks the second iteration of testing against the client in question. Cure53 was provided with sources, binaries, pertinent documentation, as well as any alternative means of access and information required to complete the review. For these purposes, the methodology chosen was white-box and a team of four senior testers was assigned to the project's preparation, execution, and finalization. All preparatory actions were completed in June 2022, namely in CW25, 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 allowing an optimal collaborative working environment to flourish. All participatory personnel from both parties were invited to partake throughout the test preparations and discussions. One can denote that communications proceeded smoothly on the whole. The scope was well-prepared and clear, no noteworthy roadblocks were encountered throughout testing, and cross-team queries were kept to a minimum as a result. ExpressVPN 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 achieved via the aforementioned Slack channel.

Regarding the findings, the Cure53 team achieved comprehensive coverage over the WP1 and WP2 scope items, identifying a total of six. Two of the findings were categorized as security vulnerabilities, whilst the remaining four were deemed general weaknesses with lower exploitation potential. Generally speaking, the overall yield of findings is relatively small in comparison with similarly-scoped audits, which naturally constitutes a highly-positive indication of the macOS client's perceived security strength.

Furthermore, the fact that all discovered issues were ranked with a *Medium* or lower severity rating - with no greater attack surfaces or threats identified - attests to the positive impression gained.

Nevertheless, the testing team is keen to stress that all findings located within the defined scope originated from the macOS client binaries under WP1, indicating that the ExpressVPN team should invest additional resources into fully securing these components specifically. All in all, the ExpressVPN team deserves high praise for its efforts to provide an exceptionally secure macOS client. Only a few minor hardening improvements are required to elevate the platform's security posture to an exemplary level.

The report will now shed more light on the scope and testing setup as well as provide a comprehensive breakdown of the available materials. This will be followed by a detailed definition of the severity levels assigned to the findings throughout this report, as well as a chapter outlining the test methodology, which serves to provide greater clarity on the techniques applied and coverage achieved throughout this audit. 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 macOS client binary and codebase in focus, giving high-level hardening advice where applicable.

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

Scope

- **Code audits and security assessments against ExpressVPN's macOS client and codebase**
  - **WP1**: Source-code-assisted penetration tests against ExpressVPN macOS client binary
    - **Primary audit focus:**
      - ExpressVPN client application for macOS
      - Version: v11.2.0.67174
    - **Secondary audit focus:**
      - Integrated Lightway Client
    - **In-scope items:**
      - Functionality that runs through *"xv_engine"*
      - Possible RCE Attack vectors
      - Possible Privilege-Escalation attacks
      - Possible Information Leakages (IP leaks, DNS leaks, etc.)
      - Configuration and setup of VPN protocols
      - Possible MitM Attack Vectors
    - **Out-of-scope items:**
      - The utilized wolfSSL Library
      - DoS via memory corruptions
      - All developer only parts & tests that are not shipped with production release
      - Split-tunneling features
      - ExpressVPN's APIs, but MitM attacks were in-scope
  - **WP2**: Source-code audits and reviews against ExpressVPN macOS client codebase
    - In addition to listed in- and out-of-scope items in WP1, any source files labeled as "posix", "linux", "windows" were not considered in-scope,
    - In addition, files related to localization strings, "mock(s)", "test" and internal developer tools were also not in-scope
    - In scope however were all files labeled as "unix"
  - **Test-user accounts were created and activated for the auditing team**
  - **All binaries in scope were shared with Cure53**
  - **Test-supporting material was shared with Cure53**
  - **All relevant sources were made available for 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

Table of Findings

Identified Vulnerabilities

ID	Title	Severity
EXP-08-003	WP1: Favicon downloader facilitates fingerprinting	*Medium*
EXP-08-004	WP1: Lack of application firewall rules for VPN gateway	*Medium*

Miscellaneous Issues

ID	Title	Severity
EXP-08-001	OOS: Client-side redirect via Favicon downloader	*Info*
EXP-08-002	FP: Lack of certificate validation via InsecureSkipVerify	*Info*
EXP-08-005	WP1: Potential fingerprinting via Sentry requests	*Low*
EXP-08-006	WP1: Data share disabling does not apply immediately	*Info*

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

Test Methodology

This section documents the testing methodology applied during this engagement, shedding light on various components of the ExpressVPN macOS client. Further clarification concerning areas of investigation subject to deep-dive assessment is offered in addition, in lieu of significant security vulnerabilities detected.

Whilst the investigated areas were divided into two work packages targeting the macOS ExpressVPN binaries and the source code parts of the application, the following list includes both work packages for the sake of simplicity. The enumerated list of items offered below underlines all noteworthy tasks performed against the macOS ExpressVPN application and sources during the audit. Particular attention was also placed on methods typically used in white-box-based testing.

- Assessments were initiated to determine the method by which the frontend parts handle WebView-related content and user-controlled data is processed. Here, testing confirmed that WebView is not heavily utilized and content is sufficiently encoded by the relevant application parts, preventing client-side injections in this area completely.
- An explicit focus was placed on examining application parts that are executed with system privileges. Therefore, Cure53 checked the installation and uninstall scripts, which are processed with higher permissions.
- In the *postinstall* migrations, the observation was made that attacker-controlled files can be moved to the protected *com.expressvpn.ExpressVPN* folder located in the *Library* path, though no impact for further exploitation could be identified.
- The macOS installer was also assessed to confirm that temporary files are indeed written to a secure location.
- In addition to privilege escalation attacks, Cure53 checked the *expressvpnd* service, which also runs with system privileges. Priority was granted toward the JSON RPC protocol since it can be reached by any local user. The various command handlers were carefully reviewed with respect to logic bugs and injections into the various command executions.
- Using dynamic instrumentation with *dtrace* and code review, any alternative attack surfaces against the privileged helper were investigated. For example, the generation of config files or whether other insecure files and binaries are used.
- Every supported VPN protocol leads to different binaries or config files invoked by *expressvpnd*, thus each configuration was reviewed for general misconfigurations and privilege escalations.
- Despite an intensive investigation of used locations and resources, no issues were detected in this area that could be exploited by an unprivileged user.
- Additionally, the usage of shared libraries with its relative *@rpath* variables were assessed on all binaries for potential dylib hijacking attacks, though no findings

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

were identified in this regard. The defined paths only refer to protected *Applications* and *Library* folders, which prevents privilege escalation attacks. Also, no optional loading via *LC_LOAD_WEAK_DYLIB* was found to be in use.

- In addition to command injections and remote code execution attacks, Cure53 investigated the sources for potential sinks that embed user input into functions executing system commands. The engine application parts properly escape user input by inserting variables separately as new arguments to Go's *Command()* function. No string concatenation or argument injection could be spotted in this area.
- Additionally, the *Lightway* client based on C was also evaluated for potential command executions that could occur via common sinks, e.g. via *system()* calls or buffer overflows. Positively, no issues were detected in this area.
- In addition, ExpressVPN was assessed to determine potential usage of dangerous functions, which could result in code execution attacks via insecure deserialization of user input, for example. Positively, no issues were spotted in this area.
- The client application, daemon, and overall configuration were also carefully reviewed regarding the disclosure of sensitive information, which may affect user privacy. In this area, four issues were found that facilitate the fingerprinting of users (see EXP-08-003), allow leaking the client IP address to attackers with network sniffing capabilities (see EXP-08-004), a potential fingerprinting issue via the third-party service Sentry (see EXP-08-005) and the UX around disabling of sharing data (see EXP-08-006).
- Alternative areas that could lead to deanonymization or may affect the privacy of ExpressVPN users were found to be solid, since the application reliably prevents leakages that could occur via DNS or WebRTC communication for instance.
- Regarding network separation, testing was conducted to determine whether other connected VPN clients or internal servers can be reached via the established VPN connection, which may prove attractive for further attacks. However, all clients are sufficiently separated from each other and no other pertinent internal hosts could be reached during the testing phase.
- Cure53 also verified that the application accordingly validates the IDs of the corresponding ExpressVPN web extensions, which are permitted to communicate via *Native Messaging* to the socket.
- Furthermore, it was examined whether external interaction via the browser with the local sockets is possible. However, the default configuration of the JSON RPC server relies on a Unix socket and the event server on a UDP socket, which did not allow any opportunity for remote exploitation.
- Additionally, testing was initiated to determine whether malicious applications are able to perform higher privileged actions, such as disconnecting the user from the VPN via the running JSON RPC server. All actions that could lead to a

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

change of the behavior or state of the ExpressVPN application were deemed correctly protected.

- Assessments were made to confirm how the application handles local files and whether sensitive information is written to unprotected locations. In addition, it was checked if file writes and reads can be abused for potential path traversal attacks, though the sufficient validations in place successfully deterred any exploitable issues in this regard.
- Regarding cryptography, the codebases were checked for usage of outdated or weak crypto algorithms, ciphers, protocols or hash functions, though no findings were identified in this area that could weaken the macOS ExpressVPN application. The areas wherein non-secure randomness was utilized were deemed security irrelevant.
- Moreover, Cure53 searched for potential weaknesses that could increase the application's susceptibility to typical MitM attacks.
- This also includes checking whether the application properly verifies the SSL certificate for the API or other requests. Here, the testing team found that the captive portal check can be intercepted, which is to be expected. However, the ability to abuse this for deanonymization by setting a cookie or redirect was determined, though no significant issue was spotted in this area either.
- Aside from the aforementioned, testing was conducted to determine whether dependencies in the defined scope are prone to known vulnerabilities or weaknesses. Fortunately, these efforts yielded a lack of issues in addition.

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

Identified Vulnerabilities

The following sections list 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 simply given in brackets following the title heading for each vulnerability. Furthermore, each vulnerability is given a unique identifier (e.g., *EXP-08-001*) to facilitate any future follow-up correspondence.

EXP-08-003 WP1: Favicon downloader facilitates fingerprinting *(Medium)*

***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 ExpressVPN application allows users to configure shortcuts to applications or websites. If a user creates a shortcut to a third-party website, the client application attempts to fetch the website's Favicon and displays it in the UI. Here, the confirmation was made that this process can be abused by a malicious server to fingerprint or deanonymize the user if fetching the icon fails. This is possible since the HTTP client stores the server cookies in a session and submits these cookies for each icon download request. Notably, the icons are fetched in both a connected and disconnected VPN state without the user knowing that a request had been issued. Furthermore, the HTTP client submits a user agent header with detailed version information.

Steps to reproduce:

1. Add a shortcut to a self-controlled server (e.g. *https://my.server*). When requesting the URL, the server should set a unique cookie in the response.
2. Open the shortcut preferences when the VPN is in a disconnected state. The application will attempt to fetch the icon with the user's clear IP.
3. Observe the incoming request on the server's side from the clear IP containing the unique cookie.

   Incoming request (clear IP):

   ```
   GET /favicon.ico
   Host: my.server
   Accept: */*
   Cookie: uniqid=test
   User-Agent: ExpressVPN/67174 CFNetwork/1128.1 Darwin/19.6.0 (x86_64)
   Accept-Language: de-de
   Accept-Encoding: gzip, deflate, br
   Connection: keep-alive
   ```

4. Connect to an arbitrary VPN location. The application will attempt to fetch the icons again using the public VPN IP.

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

5. Observe the incoming request from the VPN public IP containing the same unique cookie.

   Incoming request (VPN IP):

```
GET /favicon.ico
Host: my.server
Accept: */*
Cookie: uniqid=test
User-Agent: ExpressVPN/67174 CFNetwork/1128.1 Darwin/19.6.0 (x86_64)
Accept-Language: de-de
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
```

To mitigate this issue, Cure53 advises configuring the HTTP client to ensure cookies are not stored. In addition, the *User-Agent* header should not disclose detailed version information to third-party websites.

EXP-08-004 WP1: Lack of application firewall rules for VPN gateway *(Medium)*

***Note from ExpressVPN**: There are multiple preconditions required for there to be any security impact on this finding, some of which include social engineering or getting the user to visit a malicious website. Furthermore, any attempts to remediate this finding is likely to worsen security as a result of the large complexity associated. The only impact to users exists as a result of social engineering, where a user is tricked into visiting a malicious website. In this case, there are significantly more damaging actions an attacker would likely try to take.*

Whilst reviewing the routing tables for any potential leaks, the discovery was made that the VPN gateway route can be abused to leak a user's clear IP. For proper functionality, VPN clients require a route to forward encrypted packets to the VPN gateway via the user's ethernet or Wi-Fi interface. However, this route should only be used by the ExpressVPN daemon. If a malicious actor with network-sniffing capabilities tricks the user into sending an unencrypted request to the VPN gateway, the user's clear IP will be leaked.

Steps to reproduce:

1. Initiate Wireshark to sniff on the ethernet or Wi-Fi interface and filter for port 80 (*tcp.port == 80*).
2. Trick a VPN-connected user into visiting a website that contains the following Proof-of-Concept script (see below).
3. Observe the outgoing packet in Wireshark. The unique token is transferred unencrypted to the VPN gateway so that the attacker can read the clear IP and the token.

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

The following packet was captured within a local network, but highlights that the packet remains unsent via the VPN tunnel:

Captured packet:

```
Source          Destination       Protocol Info
192.168.0.37    45.95.243.214     HTTP     GET /?uniqueToken123123 HTTP/1.1
```

The excerpt of the routing tables offered below demonstrates that packets to the VPN gateway (*45.95.243.214*) are sent over the ethernet interface.

Routing table excerpt:

```
Routing tables

Internet:
Destination         Gateway             Flags         Netif Expire
0/1                 10.49.0.5           UGc           utun3
default             192.168.0.1         UGSc          en0
[...]
45.95.243.214       192.168.0.1         UGH           en0
```

The following script submits unencrypted HTTP requests to the */24* subnet of the user's public VPN IP. This is required since the public IP differs from the VPN gateway IP but resides in the same subnet.

PoC script:

```
<script>
async function leak(unique){
  const res = await fetch("https://api.ipify.org");
  const pubip = await res.text();
  var ips = [];
  var ip = pubip.split('.');
  for(let i=0;i<256; i++){
    ips.push(ip[0]+'.'+ip[1]+'.'+ip[2]+'.'+i.toString());
  }
  await Promise.all(ips.map(async (ip) => {
    const res = await fetch('http://'+ip+'/?'+unique);
  }));
}
leak("uniqueToken123123");
</script>
```

To prevent leaking the clear IP of VPN users to attackers with network sniffing capabilities, it is recommended to configure an application firewall rule that only allows the ExpressVPN daemon to use this route. Furthermore, one should consider disabling services that allow unencrypted connections on the VPN gateways.

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 lead to an exploit but might 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 way to be called. Conclusively, while a vulnerability is present, an exploit might not always be possible.

EXP-08-001 OOS: Client-side redirect via Favicon downloader *(Info)*

***Note from ExpressVPN**: This finding was marked out of scope. The security impact of the issue is negligible, while any change would impose limitations on the favicon users can set. Any exploitation of this issue would require the attacker to control a website that serves favicons, induce a user into using these favicons, and then redirect them to a different host as a result.*

The discovery was made that the HTTP client responsible for downloading icons of shortcut links follows redirects of the server. In particular, the hostname and IP address of the redirect URL remains invalidated, which allows a malicious server to redirect the client to an unintended host. This can be abused to submit GET requests to internal services or can be chained to exploit issue EXP-08-004.

Steps to reproduce:

1. Initiate a TCP socket on the local machine (*nc -l 127.0.0.1 8080*) to confirm that the redirect is followed.
2. Insert a shortcut to a URL that redirects the client to an internal service (e.g. *http://localhost:8080*).
3. Observe that the request is received by the local socket.

Incoming request:

```
GET /favicon.ico HTTP/1.1
Host: localhost:8080
Accept: */*
Accept-Language: de-de
Connection: keep-alive
Accept-Encoding: gzip, deflate, br
User-Agent: ExpressVPN/67174 CFNetwork/1128.0.1 Darwin/19.6.0 (x86_64)
```

Notably, the requests are sent again if the ExpressVPN client has established a connection to the VPN. To mitigate this issue, Cure53 advises strictly validating the hostname and IP address of all redirect URLs to match the host configured by the user.

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

EXP-08-002 FP: Lack of certificate validation via *InsecureSkipVerify* *(Info)*

***Note from ExpressVPN**: The issue does not affect the audited macOS ExpressVPN client and should be considered a false positive, as the code file this was identified in is not used by the macOS application, and all interprocess communication has proper certificate validation implemented.*

During the assessment of the sources, the discovery was made that *InsecureSkipVerify* is set to *true*, which in practice does not validate the certificate of connected servers. This could induce the risk of exploitation in several MitM attack scenarios, whereby the client will trust any server connected to.

Affected file:
xv_engine/apps/xvpnd/jsonrpc/client/client_tls.go

Affected code:

```
func TLSClientConfig(certDir string) (*tls.Config, error) {
   tlscfg := &tls.Config{
      //nolint:gosec // FIXME XVCE-1129: verify server certificate
      InsecureSkipVerify: true,
}
```

To mitigate this issue, Cure53 advises desisting use of *InsecureSkipVerify* and alternatively always validating the certificate of the server. By doing so, clients generally should only establish a connection if the identity of the server can be trusted.

EXP-08-005 WP1: Potential fingerprinting via Sentry requests *(Low)*

***Note from ExpressVPN**: As described in the finding, any correlation is only possible if an attacker has compromised sentry.io's infrastructure. In addition, users have to opt in to share these analytics. As a result of these preconditions, we consider the risk associated with this issue to be negligible.*

The macOS ExpressVPN application allows users to opt into sending analytics information to third parties, which can be toggled in the application's GUI. In case the option is activated, the observation was made that requests are sent to *sentry.io* over the normal interface and also over the VPN tunnel after a connection has been established.

Since a static *did* and a string containing *"ExpressVPN"* are included in both requests, a connection could be made to the user's clear IP. The requests are also sent within a short timeframe, which increases the likelihood of receiving a positive match. This behavior could be leveraged to potentially obtain the user's clear IP, if an adversary or state-actor retrieves access to Sentry's data or systems for an investigation.

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

Affected file:
xv_engine/pkg/raven-go/client.go

Affected code:
```
req, err := http.NewRequest("POST", url, body)
   if err != nil {
      return fmt.Errorf("can't create new request: %v", err)
   }
req.Header.Set("X-Sentry-Auth", authHeader)
req.Header.Set("User-Agent", userAgent)
req.Header.Set("Content-Type", contentType)
res, err := t.Do(req)
```

Example HTTPS request (without VPN connection):
```
POST /api/1818020/envelope/ HTTP/1.1

Host: o137163.ingest.sentry.io
Content-Type: application/x-sentry-envelope
User-Agent: sentry.cocoa
X-Sentry-Auth: Sentry
sentry_version=7,sentry_client=sentry.cocoa/6.1.4,sentry_timestamp=1656590840,se
ntry_key=7419810809da4400b40551f026979d86
[...]

{"sdk":{"name":"sentry.cocoa","version":"6.1.4"}}{"type":"session","length":273}
{"errors":0,"status":"ok","started":"2022-06-30T12:07:20.914Z","did":"D74BAF5D-
661B-4390-A806-F98E4E1D3A7A","sid":"8BBF5FD1-8E17-4BD7-BBA1-
C70A875DCCD8","init":true,"timestamp":"2022-06-30T12:07:20.917Z","attrs":
{"release":"com.expressvpn.ExpressVPN@11.2.0+67174"},"seq":1}
```

Example HTTPS request (via VPN tunnel):
```
POST /api/1818020/envelope/ HTTP/1.1
Host: o137163.ingest.sentry.io
Content-Type: application/x-sentry-envelope
User-Agent: sentry.cocoa
X-Sentry-Auth: Sentry
sentry_version=7,sentry_client=sentry.cocoa/6.1.4,sentry_timestamp=1656591151,se
ntry_key=7419810809da4400b40551f026979d86
[...]

{"sdk":{"name":"sentry.cocoa","version":"6.1.4"}}{"type":"session","length":273}
{"errors":0,"status":"ok","started":"2022-06-30T12:12:31.557Z","did":"D74BAF5D-
661B-4390-A806-F98E4E1D3A7A","sid":"0B5C4B27-BE39-46E5-8689-
065C653E335F","init":true,"timestamp":"2022-06-30T12:12:31.560Z","attrs":
{"release":"com.expressvpn.ExpressVPN@11.2.0+67174"},"seq":1}
```

However, this issue strongly depends on the security posture of Sentry's systems and also the political stance, in which way the service provider would react in those

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

situations. Thus, this issue was only assigned to the *Miscellaneous* section. Nevertheless, Cure53 recommends blocking third parties from establishing a connection in order to potentially leak the clear IP of macOS ExpressVPN users. Generally speaking, one should stop the application from being able to communicate to third-party services. If this is considered infeasible, the volume of requests sent to third parties should be reduced to ensure that only one interface is used for the communication.

EXP-08-006 WP1: Data share disabling does not apply immediately *(Info)*

***Note from ExpressVPN**: The finding identified here is intended behavior. We expect the app to be restarted if the user modifies their "Help improve ExpressVPN" preferences, as these options are loaded at application start.*

During the investigation of the Sentry integration with respect to privacy issues, the observation was made that disabling "Help improve ExpressVPN" does not immediately halt any requests sent to Sentry.

Steps to reproduce:

1. Ensure that "Help improve ExpressVPN" is enabled at launch of the ExpressVPN app.
2. Open preferences and disable data sharing.
3. Close the ExpressVPN window by clicking the X (do not fully quit the app).
4. Wait 1-2 minutes.
5. Open ExpressVPN again and observe that another request is sent to Sentry.

A full restart of the app is required to enforce the settings alteration. Generally speaking, no full violation of the privacy policy was found; however, one can recommend warning the user that an application restart is required for the setting to take effect.

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

Conclusions

In this audit, the next iteration of the ExpressVPN application for macOS was examined by Cure53. Specifically, four members of the Cure53 team completed the project over the course of twenty-five days in June and July 2022. A relatively low volume of issues were located during the audit: two of which remain exploitable and were added to the Vulnerabilities section, whilst four were considered as hardening recommendations or best practice implementations and subsequently added to the Miscellaneous section.

Two *Miscellaneous* issues were adjusted during the testing phase and were marked with "OOS" since they do not match the defined scope shared with Cure53 (see EXP-08-001 and EXP-08-002). However, it is nevertheless recommended to address these issues in order to harden the affected areas considered out-of-scope.

Cure53 was provided with a functional macOS ExpressVPN application, pertinent accounts, and sources for the audit. This significantly increased the effectiveness of the audit, allowing Cure53 to assess the applications for security vulnerabilities present in both the code and running environment.

The primary objective behind the Cure53 investigation of the ExpressVPN application for macOS was to determine whether the existing functionality of the application and its running environment can be deemed healthy enough to withstand attacks by malicious applications and adversaries. With a specific focus on typical macOS application problems, the issues connected with various types of injection attacks and misconfigurations - which could compromise the application itself or the running environment - were investigated without significant success.

Corresponding API endpoints were not deemed in scope; thus, investigations were not performed on any backend application owned by ExpressVPN that communicated with the macOS application. However, testing was conducted to determine whether a network attacker could intercept any of the network requests.

As a result, only two exploitable issues were found, both related to the disclosure of user privacy. The first issue affects the handling of user-configured shortcuts. While the URLs are properly validated, minor improvements in the Favicon downloader are recommended to protect against fingerprinting users by third-party sites. See ticket EXP-08-003 for additional guidance.

The second issue was located during the examination of the running ExpressVPN daemon as well as the general configuration of the differing VPN protocols. Heightened focus was placed on the routing tables and firewall rules to determine if sensitive data can be leaked or locate any issues that could compromise user privacy. In this regard,

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

all components made a solid impression. However, a minor exception caused by a missing firewall rule was found and documented under ticket EXP-08-004. This issue could enable nation state attackers to deanonymize user's by leaking their clear IP. A third issue in this area was added to the report that could be leveraged to deanonymize ExpressVPN users via the third-party service Sentry (see EXP-08-005). In order to further reduce risk of deanonymization via adversaries, Cure53 highly advises addressing this issue in addition. The testing team also observed that disabling information sharing does not take effect immediately, which should be communicated clearly to the user. Further information regarding this issue can be perused in ticket EXP-08-006.

The examined codebase components of the macOS ExpressVPN application garnered a solid impression regarding the security posture. In general, the codebase adheres to and implements common best practices to facilitate an excellent security state. Static analysis tooling also seems integrated into the lifecycle, which further minimizes any leeway for erroneous behaviors. Overall, the architecture and separation of privileges was deemed soundly-designed with the attack surface highly minimal as a result. In relation to the issues previously detected by the Cure53 team, all relevant findings related to the macOS ExpressVPN application have been sufficiently addressed and do not incur any new problems or weaknesses. This is evidently a praiseworthy result and underlines the security awareness of the ExpressVPN team. In conclusion, this assessment of the latest ExpressVPN application for macOS iteration leaves an exceptionally solid impression as regards to security.

The average impact hovers between *Low* and *Medium*, thereby indicating that stable protection has been established against a host of attacks targeting the examined scope. However, the issues detected in this audit also highlight the essential requirement for improvement in some areas, particularly concerning possible information disclosure. This indicates that not all best practices and recommendations in this area are taken into account.

Nevertheless, as a result of the absence of major issues and strong impressions gained during the audit, Cure53 can only confirm that the ExpressVPN team instills due diligence in its efforts against the many and varying threats that modern VPN applications tend to face. The robust state of the examined codebase further corroborates this viewpoint. Once all relevant issues have been mitigated, Cure53 would be happy to confirm that the audited version of the examined ExpressVPN application and areas in scope are sufficiently secured for production use.

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