TunnelBear Security Assessment Summary 11.2019

Cure53, Dr.-Ing. Mario Heiderich & Team

Introduction

This technical summary report describes the results of a large-scale assessment of various components within the TunnelBear VPN complex. This project, which was carried out by Cure53 in November 2019, entailed a broadly-scoped penetration test, a security review, as well as auditing of sources pertinent to the selected items from the TunnelBear VPN scope. Both the project itself and the ensuing report on the security assessment's format, methods and findings were requested by the maintainers of the TunnelBear compound.

In terms of the objectives, the main goal was to gain an up-to-date image of the scope with an in-depth, extensive security review. In other words, the TunnelBear team has a continuing and well-founded interest in knowing how well their security promises actually hold against determined attackers. Cure53 assists them in these efforts by assuming the roles of attackers and attempting to find blind spots that might have slipped past the generally exceptional levels of security handling at the TunnelBear complex.

It should be noted that the Cure53-TunnelBear security-driven cooperation was established in 2016. The current report, stemming from the work conducted in 2019, is a third summary of this type provided by Cure53, with the initial one issued in the summer of 2017 and the second shared with the TunnelBear team in 2018. As a result of previous engagements, it can be said that the Cure53 team generally has a certain degree of familiarity with the TunnelBear compound. However, with ongoing projects like audits and tests against various items in the TunnelBear scope performed since the last assessment, Cure53 is happy to issue an updated verdict on the current security posture of the TunnelBear complex.

Test Summary & Methodology

Given the breadth and depth of this assignment, a budget dedicated to its completion stood at thirty-seven person-days in total. The Cure53 team inspected and analyzed the security posture of the TunnelBear scope over the course of Calendar Weeks 45 and 46 2019, signifying late autumn. With the multi-layered and multi-dimensional nature of the scope, ten members of the Cure53 team were drafted to take part in this project. Each tester/auditor had been chosen on the basis of the optimum, advanced and complimentary skills possessed by the members who all had a proven capacity of addressing the assessment's goals as efficiently and as comprehensively as possible.

Capitalizing on best practices implemented in previous testing installments Cure53 was granted a level of access deemed as necessary for reaching an expected level of coverage on all included parts within the software and infrastructure compound. The TunnelBear team made all relevant configuration data, tests-servers and test-user credentials, as well as relevant source code, available to Cure53. Further, the testing team could consult material deposited into GitHub repositories utilized by TunnelBear.

For this autumn 2019 assessment, six components of the TunnelBear complex were specially chosen as the main items to be examined. Based on this selection, Cure53 delineated six Work Packages (WPs) with specific test-targets. The WP1 centered on the TunnelBear client applications, spanning iOS, Android and Windows branches. All apps were subject to penetration testing and received dedicated code audits. Next, in WP2, Cure53 investigated the TunnelBear browser extensions, deploying same methods as above. Slightly altered methodology of a configuration review paired with penetration testing was adopted in the VPN infrastructure examination in WP3. While WP4 zoomed in on the TunnelBear FilterPods, WP5 shifted to the backend of both TunnelBear and PolarBear. Rounding the scope was Work Package 6, which concerned the TunnelBear frontend and public site.

The project started on time and progressed efficiently, with no noteworthy technical issues hindering the completion of the various steps within the investigations. As the investigations went on, Cure53 filed the discoveries into a JIRA instance made available for this assessment. In addition, communications between the teams were facilitated by Slack. To clarify, TunnelBear opened a dedicated channel on their workspace and invited the participating Cure53 members to join it. On Slack, Cure53 could ask questions, give feedback about the emerging findings, as well as discuss the process of fix verification executed by the in-house team at TunnelBear for the live-reported issues.

Audit Results

The findings underline the importance of a temporal perspective being adopted when discussing and assessing the security premise characterized by extreme complexity. Over several years since 2016, Cure53 has gained substantial knowledge about the complex and can issue an evidence-based verdict about the progress being made at the TunnelBear entities as regards security. The improvement observed within TunnelBear can certainly be attributed to the dedication and skill-level of the in-house team, which manages to have a good grasp over the security landscape, in spite of the noted size and intricacies of the compound they are protecting.

Even though there can be no doubt about the TunnelBear project becoming more and more secure with subsequent assessments, the testers from the Cure53 team still identified twelve security-relevant items across the executed Work Packages. Compared to the last round of testing, the total number of issues affecting the complex has been nearly cut in half, again ascertaining to a considerable amelioration.

Examining the problems at the meta-level reveals that six items should be seen as actual vulnerabilities, while those in the remaining batch can be classified as general weaknesses with arguably lower exploitation potential. All identified issues are presented in the table below. Quite clearly, attention should be drawn primarily to the notable issues with “Critical” and “High” severities.

Vulnerability	Description
Critical (2)	
WP1/OSX	The problem signifies local escalation of privileges resulting from a race condition. In particular, the affected helper is checking the signature of the OpenVPN binary and refuses to execute upon a failure to match. However, there is a time-of-check to time-of-use (TOCTOU) race condition between the signature verification and the execution of the binary. If the race condition is won, then the backdoor *openvpn* script is copied to a safe location at the beginning of the function, ultimately leading to a successful backdoor script execution.
WP5/Backend	Thanks to notable extra effort made by the TunnelBear staff, Cure53 received additional access to a staging version of the TunnelBear’s administrative web console. This was justified by suspicions Cure53 shared about potential issues hiding there. The initial clues were confirmed and resulted in proving XSS in the backend. Specifically, Cure53 observed that a user can issue feedback which gets logged into the account-details. Due to insufficient sanitization, it was possible to execute arbitrary JavaScript. The problem would allow attackers to, e.g., ban users or bypass fraud checks.
High (4)	
WP1/Android	Cure53 identified a vulnerability in one of the exported broadcast receivers. The registered receiver listens for actions and allows other apps to send content to the widget provider via intent calls. Parsing lacks proper validation or exception handling, thus making it possible for the attacker to send out a malformed intent. In essence,

	the TunnelBear Android app was proven prone to several DoS attack scenarios. In effect, users would have been rendered unable to operate the continuously crashing app.
WP5/Backend	A rather simple mistake was made when the wrong Java construct was employed for a string comparison and led to a CSRF bypass. More specifically, the check in use tries to assess if the request method is *POST* and then continues to validate the submitted token. Using the double equals operator in Java, it does not compare the actual value of two strings but just two object references. Under certain circumstances, the check might succeed because the Java VM references the same internal object to hold the simple phrase POST for multiple objects throughout the codebase.
WP2/ Extension	The Firefox browser extension relies on HTTP proxy to establish secure connection. The initial authentication proxy suffers from a flawed logic in the domain of automatic insertion of credentials. Any HTTP 401 response could capture sensitive data because the *browser.webRequestonAuthRequired* method intercepts it. As a consequence, VPN token can be leaked via HTTP 401 basic auth and the identity of the user could be revealed by associating Internet activity across websites with the same session credentials.
WP1/Windows	Another flaw relates to possible command injection in the obfuscation service on Windows, specifically the service offered through GhostBear. The parameters are passed to the *obfs4proxy* command via the *ProcessStartInfo()* function without validation. As the first parameter is inserted into the *sharedSecret* argument, an attacker might be able to insert payloads and execute arbitrary commands from there. Cure53 was unable to start a valid MitM attack and insert malicious commands into the parameter thanks to a well-implemented certificate Pinning check, hence positioning this finding in the category of general weaknesses.
Medium (1)	
WP3/VPN Infrastructure	It was noticed that the SlothBear service, which monitors VPN connections, handles dynamic input insecurely. The issue resides in the code for the *user.CaseID* variable, which is read from the incoming POST data. Since *caseID* is never actually sanitized for safe usage in command line statements, it is possible to prefix this variable with a dash character and inject additional command line arguments to the *tcpdump* command. Consequently, this item

	signifies argument injection in the backend software stack.
Low (2)	
WP5/Backend	Cure53 confirmed insecure handling of redirects in the TunnelBear backend with an Open Redirect problem located in the URL parameter of *redirectAction*. The affected route of */core2/redirect* within the *tbearDashboard2* component receives a URL and forwards it without proper validation. Therefore, it is possible to inject URLs pointing to hostnames other than *tunnelbear.com*, introducing possible Phishing or similar attacks.
WP3/VPN Infrastructure	Auditing the code of the *dnsproxy* backend software revealed it to contain a strict HTTP proxy functionality. This item attempts to block traffic entirely unless it is destined for *tunnelbear.com*, meaning that the DNSProxy vigilant-mode can be bypassed.
Informational (3)	
WP5/Backend	The code audit revealed multiple occurrences of improper error handling and printing of stack traces into log files. This might lead to unexpected disclosure of sensitive information, as the log files may contain IP addresses or other PII data.
WP5/Backend	When examining the Redis access logic, Cure53 discovered that the application was inconsistent in embedding user-input into Redis server keys. The utilized *VPNTokenManager* class suffixes a string constant to the user-input before writing it to Redis, so a malicious authenticated user can manipulate specific data in Redis.
WP5/Backend	The audit demonstrated that multiple docker files download resources via plain-text HTTP connections, which are then installed into the docker container. Assuming an MitM attacker, this signifies a possibility to replace the requested software with a malicious binary.

It is important to note that Cure53 assisted the TunnelBear team with developing some of the fixes, especially when more fine-tuned recommendations were needed because the choice of a solution was not immediately apparent. Nevertheless, it must be underscored that the Cure53 team did not perform a full and complete retest but rather engaged in a thorough and comprehensive fix verification, confirming that mitigation strategies deployed by the TunnelBear are sound and aligned with various modern and up-to-date recommendations. The main focus has been placed on devising processes that can help TunnelBear make sure that security resources are correctly allocated to regular audits and targeted checks, with the aim of preventing recurrence and regressions.

Conclusions

Finalizing the last report, which was issued in 2018, Cure53 has already stated that the TunnelBear today is a completely different complex than the one initially encountered by the testing team back in 2016. From a security perspective, the difference is quite extreme as just a few years proved to be enough to eradicate nearly all mistakes and security flaws The TunnelBear complex has been transformed from being just ‘average’ to becoming a clear frontrunner among its VPN competitors when it comes to security.

On this note, looking at the project in its entirety through a temporal lens cannot be downplayed, especially as the periodic and long-term engagement over the years made the Cure53 team equipped with substantial knowledge-base when it comes to the TunnelBear security. Not only the mutual trust, but also the methods, have been steadily developed and improved, with current approaches being extremely well battle-tested. As a result, the Cure53’s conclusions feature aspects of comparability, consistency and logic of examination that can only be achieved over time.

To reiterate, the results of this autumn 2019 Cure53 assessment of the TunnelBear VPN complex point to right direction of development being properly maintained. The benchmarks are better and better, while ambitious security milestones are being set, despite the TunnelBear’s increasing scale and complexity. After spending thirty-seven days on the scope in November 2019, ten members of the Cure53 team can conclude that the security posture of the tested TunnelBear components is sound and generally solid.

On the one hand, the presence of twelve security-relevant findings, inclusive of several rather severe flaws - marked as Critical and High, cannot be disregarded. Further, Cure53 managed to spot issues across all six Work Packages, meaning that every component of TunnelBear remained exposed to some risks. On the other hand, the findings must be read in the context of a pronounced vastness of this modern and ever-growing scope.

Taking all evidence and observations into consideration, the testing team is happy to report that both the total number and the spectrum of severities displayed by the spotted findings have diminished. Even though Cure53 covered a similar scope in 2019 as it did in the past assignment in 2018, the overall results indicate that TunnelBear benefits from more advanced security approaches and safeguards its users to a much greater extent. In that sense, Cure53 wishes to underline a definite and measurable improvement in the TunnelBear's security posture. The fact that this round of penetration testing required a considerable number of deep-dives to uncover potential issues is also significant, as it effectively means that exposure to classic, standard and typical security problems is a thing of the past for the TunnelBear complex.

In this context, Cure53 wishes to draw attention to both dedication and excellent skills displayed by the TunnelBear team over the course of this autumn 2019 assessment. One example of note was the professional and prompt communication during the test, which effectively led to the patch management being deployed in record time. As a result, Cure53 was able to review the first bug fixes when the tests were still ongoing

To summarize, the TunnelBear complex should be seen as always trying to have an edge over its competitors when it comes to security. With this third testing round completed in November 2019, Cure53 has gathered enough evidence to attest to the changing and improving posture of the numerous TunnelBear components. Despite finding a number of vulnerabilities and miscellaneous weaknesses that can be potentially turned into vulnerabilities, the Cure53 testing team is positively surprised about the directionality and pace of secure implementations becoming a standard at TunnelBear. All in all, TunnelBear is definitely on the right path and stands out as a mature application which clearly cares about their project's security posture and privacy of their users.