SECURITY ASSESSMENT OF EXPRESSVPN WINDOWS APPLICATION (V10)

ExpressVPN

2022-03-18

CLIENT CONFIDENTIAL

Project Team

Assessment Team

Lead Consultant	Zach Hew
Consultant	Andy Tan

Quality Assurance

Document QA	Weiming Theh Ben Tudor
Document Technical QA	Sven Schleier Riaan Naudé

Contents

1 Executive Summary

F-Secure Consulting was engaged by ExpressVPN to conduct a penetration test targeting the ExpressVPN Windows Application (v10).

The scope of the penetration test was to conduct a "White-box" Windows Application security assessment targeting the ExpressVPN Windows Application (v10) within the Production environment, for 14 man-days (inclusive of reporting) between 29 November 2021 and 28 December 2021.

ExpressVPN Windows (v10) version 10.14.0.8 was used at the time of writing as the ExpressVPN client for Windows. It provides connection via various VPN protocols to the ExpressVPN service on the Windows platform.

The purpose of the assessment was to identify security weaknesses within the ExpressVPN Windows Application (v10) that could be exploited by an adversary to affect the confidentiality, integrity or availability of systems or data, or which could be used to cause significant reputational damage to ExpressVPN.

The main objectives of this assessment were to identify vulnerabilities relating to:

1. Information disclosure or IP address leakage
2. Remote code execution (RCE)

While the assessment was performed using a "White-Box" methodology, a "Grey-Box" approach was also used to evaluate components with no source code provided. To elaborate: F-Secure Consulting was provided with the QA build (debug production build) which facilitated testing, and the testing/development tools were out of scope.

The assessment was conducted from F-Secure Consulting's Singapore office, by the F-Secure Consulting Singapore offensive security team.

The approach to testing was conducted in line with accepted industry standards, such as Open Source Security Testing Methodology Manual (OSSTMM), Open Web Application Security Project (OWASP), CREST, and the CHECK guidelines produced by NCSC for testing UK Government protectively marked systems and networks.

The severity ratings used within this report are based on the Common Vulnerability Scoring System version 3 (CVSSv3) – please note however that this report does not contain the detailed vulnerability description or actual numerical CVSSv3 scores.

During the assessment, one (1) Low severity issue was discovered, along with five (5) Informational observations. A vulnerability was identified which can allow a local attacker to execute arbitrary code on a Windows machine running the ExpressVPN Windows Application (v10). There were no vulnerabilities identified which would directly result in information disclosure or IP address leakage. Below are the three (3) primary findings identified during this assessment:

- The use of .NET Remoting services resulted in the insecure deserialization of untrusted data. However, the attack surface is only limited to the user running ExpressVPN; it can neither be exploited remotely nor from another user on the same host. It also does not provide an attacker with additional access rights or privileges. Overall, there is no security impact observed on the exploitation of this issue.

- The application uses insecure C/C++ functions and was compiled with insufficient binary protections. While this increases ExpressVPN's susceptibility to vulnerabilities such as buffer overflow attacks, it was confirmed that the usage of these functions was not exploitable, as the input values have already been safely validated prior to usage (e.g. Null-terminated). This was highlighted due to deviation from secure coding practices.
- The minimum TLS version is not defined in several TLS configurations within `xv_engine` repository. This may result in the use of TLS 1.0 by default, which deviates from security best practices.

While the likelihood of a security incident occurring is low, ExpressVPN should consider implementing the recommendations made within this report to further reduce the potential.

In summary, F-Secure did not identify vulnerabilities which can be exploited to cause information disclosure, IP address leakage or RCE in the ExpressVPN Windows application. The application did not expose an excessive number of services to the network. However, the Named Pipes created could be exploited to run arbitrary code locally on the machine during the assessment. This access is, however, limited to the current user's privileges (i.e., no privilege escalation vector).

Retest Status (2022-02-21)

A retest assessment of the following six (6) issues was conducted on 21 February 2022 against the ExpressVPN Windows Application (v10) version 10.19.0.7. The following is a summary of the retest status for each issue.

Windows Application Assessment Vulnerabilities

- Insecure TLS/SSL Configuration – **Closed**
- Insecure .NET Remoting Services Configured – **Closed**
- Insufficient Windows PE Binary Protections – **Closed**
- Insufficient Windows DLL Binary Protections – **Closed**
- Insecure Functions Used – **Closed**
- Overly Permissive Folder Permissions – **Closed**

In addition, as discussed with ExpressVPN, the following two (2) issues have been added to the report to distinctly highlight the specific PE and DLL binaries that involves external dependencies to resolve.

- Insufficient Windows PE Binary Protections – External Dependencies
- Insufficient Windows DLL Binary Protections – External Dependencies

2 Approach, Findings and Recommendations

Overview

This section of the report discusses the main issues that were discovered together with high level recommendations as to how they may be best resolved. Further details on individual issues are given in Section 4 of this report.

The purpose of this engagement was to verify that security controls were effective against prevailing threats and vulnerabilities, through the identification of security weaknesses. Technical recommendations were also provided to ExpressVPN to eliminate or mitigate the identified risks.

The objective of the security assessment is outlined below:

- Identify security weaknesses that could be leveraged by an attacker to access or modify sensitive data or assets, or cause disruption to service as a result of installing ExpressVPN Windows (v10) application.
- Provide ExpressVPN with assurances with regards to the security posture of the in-scope application.
- Highlight areas where improvements could be made to reduce the risk of a security incident occurring, thereby helping to ensure the confidentiality, integrity, and availability of systems and data belonging to ExpressVPN or their users.

ExpressVPN Windows (v10) version 10.14.0.8 was used at the time of writing as the ExpressVPN client for Windows. It allows the client to connect via various VPN protocols to the ExpressVPN service on the Windows platform. In this assessment, the client application and its components, together with the various VPN protocols, were audited for vulnerabilities.

The protocols supported are listed below:

- Lightway (UDP and TCP)
- OpenVPN (UDP and TCP)
- IKEv2
- IPsec

The VPN configuration files and connections set up using the above protocols were reviewed for vulnerabilities that may expose information or compromise user privacy. The protocols themselves were not assessed during the time-boxed assessment. Inbound requests to the API servers (AWS), as well as the VPN servers itself, were out of scope of the assessment. However, traffic to/from the VPN servers was still considered in scope.

Windows Application Security Assessment

The ExpressVPN Windows application (`ExpressVPN.exe`) entails a minimal user interface design and does not require elevated privileges to run. Users may choose from a predefined list of countries and protocols for the VPN connection.

The VPN's configuration data were stored securely within `C:/ProgramData/ExpressVPN/v4` directory and requires elevated privileges to access. Free-form inputs such as website links and file paths were properly validated and launched with the same privileges as the parent process.

The ExpressVPN Windows service (`expressvpnd.exe`) was started with elevated privileges, and runs a JSON RPC server (using Golang http server) listening on TCP port 2015. For each action (e.g. Connect, Disconnect etc.) performed on the Windows application (the "client"), a HTTP request with a JSON payload is sent to the JSON RPC server (the "server").

Traffic between the client and server were limited to the local machine (i.e. localhost). As such, there was no SSL/TLS implemented by design.

The server was observed to perform sufficient input validation against the JSON parameters sent by the client. Different payloads were used to identify injection-based vulnerabilities (e.g. SQL injection, Cross-Site Scripting). However, no payloads were successfully executed during the assessment.

The business logic flow also matched the state machine's design and cannot be bypassed. For example, it was not possible to establish a VPN connection by calling `XVPN.Connect` method directly to the server after the user's subscription had expired.

Users were prompted to re-connect the VPN after changing the VPN connection protocol. When the Lightway protocol is used, the service would start `lightway.exe` with elevated privileges.

There were no vulnerabilities identified within the Windows service which could be exploited by a malicious adversary to execute arbitrary code with elevated privileges during the time-boxed assessment. The Windows service would also start a new process immediately whenever its process had been terminated ungracefully.

In a similar manner to the ExpressVPN Windows application, the ExpressVPN Notifications Service (`ExpressVPNNotificationService.exe`) runs without elevated privileges, and creates a Named Pipe for inter-process communication (IPC). It is noteworthy that Microsoft no longer recommends .NET Remoting services for IPC.

The Named Pipes created by both `ExpressVPN.exe` and `ExpressVPNNotificationService.exe` were found to be insecurely configured. The `BinaryServerFormatterSinkProvider` class was observed to be instantiated with `TypeFilterLevel.Full`, which allowed arbitrary deserialization of objects sent by clients[1].

This vulnerability was confirmed during the assessment; it was possible to send serialized data using ysoserial.net[2] payloads to the Named Pipes of `ExpressVPN.exe` and `ExpressVPNNotificationService.exe`, leading to local arbitrary code execution on the machine.

[1] https://labs.f-secure.com/advisories/milestone-xprotect-net-deserialization-vulnerability/
[2] https://github.com/pwntester/ysoserial.net

However, as this attack vector does not provide the local attacker with any additional access rights or privileges, nor can it be exploited remotely, the likelihood of exploitation and impact to ExpressVPN's client is deemed to be low.

Network communication to the VPN servers were established using TLS1.3, in line with security best practices.

- For the Lightway protocol, the configuration defined in `helium.conf` was deemed to be secure, with data transmitting securely using TLS 1.2 at the minimum.
- Similarly, the configuration defined in `config.ovpn` for the OpenVPN protocol was also securely configured, and in line with security best practices. These practices include the use of keys and certificates generated using at least 2048 bits and using TLS 1.2 at the minimum.
- VPN connections established using IKEv2 and L2TP/IPsec protocols also supported secure data-in-transit using Encapsulating Security Payload (ESP), which provides Data Integrity, Encryption, Authentication, and Anti-Replay security controls.

It was not possible to gain information about ExpressVPN's clients or out of the network traffic. Nor was it possible to execute code remotely through attacks such as, but not limited to, Man-in-the-Middle (MitM), TLS downgrading, packet injection.

The ExpressVPN installer required administrative privileges to run, but provided no avenues for privilege escalation attacks. Users were limited to clicking the "Install" button to complete the full installation process; the installer provided no other options for users to configure (e.g. installation path).

The files and folders in the default installation and `ProgramData` directories were also found to be securely configured, and did not provide any privilege escalation vectors, with the exception of `C:/ProgramData/ExpressVPN`, which had write access granted for all users. The `installer_config` file was observed to be read from the affected folder with elevated privileges during the product activation stage.

Although there was no successful exploitation during the assessment, an attacker may potentially overwrite a file path in the configuration file. In turn, the file path pointing to an attacker-controlled malicious executable may be started with elevated privileges during the next product activation.

The executables and DLL libraries in use by ExpressVPN were also found to be missing various binary protections offered by Microsoft Windows, such as Structured Exception Handling (SafeSEH), Control Flow Guard (CFG), Address Space Layout Randomization (ASLR), Data Execution Prevention (DEP), and High Entropy VA. The generation of binary files without these mitigating techniques makes it easier to exploit software vulnerabilities reliably.

Additionally, the source codes in the following repositories were reviewed under time-boxed conditions:

- **xv_helium_cli (Lightway)**
  - The connector that allows Lightway to be used as a VPN protocol
- **xv_engine**
  - Main logic that handles connections and application functionality
- **xv_win_v4**
  - Main Windows application code; handles setup and installation

F-Secure Consulting placed emphasis on code within `ExpressVpn.Client.Windows.UI`, `ExpressVpn.Utils.Wmi`, and `Install` directories as highlighted by ExpressVPN. Third-party libraries, dependencies, development/testing tools and files labelled for non-Windows OSs were considered out of scope.

C and C++ functions which does not safely validate user inputs were found to be used within the `xv_engine` repository. While this increases ExpressVPN's susceptibility to buffer overflow or Denial-of-Service (DoS)-driven attacks (i.e., application exceptions, undefined behaviour), it was confirmed that the usage of these functions was not exploitable as the input values were properly validated (e.g. Null-terminated strings) and not user-controllable.

Additionally, the use of insecure TLS/SSL configurations were also identified within the same repository, but only affects non-critical functions (e.g. speed test download), and not the main VPN connection itself. These findings were raised due to deviation from security best practices; there is no impact to the security of the main VPN connection.

High Level Recommendations

To prevent and mitigate security risks posed by the identified vulnerabilities, it is recommended that ExpressVPN adopt the following high-level recommendations:

- Use Windows Communication Foundation (WCF) protocol and deprecate the use of .NET remoting services
- Implement sufficient binary protections for all Windows executables and DLLs
- Use secure SSL/TLS configurations and secure C and C++ functions as part of secure coding/development
- Reconfigure the affected folder's permissions in line with the Principle of Least Privilege (PoLP)

3 Summary of Vulnerabilities

3.1 Total Vulnerabilities Found During this Test

The following table presents the total number of vulnerabilities discovered during testing, by severity.

Scope	CRITICAL	HIGH	MEDIUM	LOW	INFO	Total
Windows Application Assessment Vulnerabilities	0	0	0	1	7	8
Total	0	0	0	1	7	8

3.2 Summary of Vulnerabilities Found

Windows Application Assessment Vulnerabilities

Severity Level	Vulnerability Name	Issue Status
LOW	Insecure TLS/SSL Configuration	CLOSED
INFORMATIONAL	Insecure .NET Remoting Services Configured	CLOSED
INFORMATIONAL	Insufficient Windows PE Binary Protections	CLOSED
INFORMATIONAL	Insufficient Windows DLL Binary Protections	CLOSED
INFORMATIONAL	Insecure Functions Used	CLOSED
INFORMATIONAL	Overly Permissive Folder Permissions	CLOSED
INFORMATIONAL	Insufficient Windows PE Binary Protections – External Dependencies	OPEN
INFORMATIONAL	Insufficient Windows DLL Binary Protections – External Dependencies	OPEN

4 Windows Application Assessment Vulnerabilities

Insecure TLS/SSL Configuration

Windows Application	LOW
Issue Status	CLOSED

The application was observed to configure the TLS/SSL settings insecurely for speed test tools and sending of crash reports, where the `MinVersion` attribute was not defined in the TLS configuration, which defaults to TLS1.0.

The TLS connection may be established using a weaker and vulnerable TLS protocol (1.0 by default) specifically for speed test tools and sending of crash reports when the user is not connected to the VPN.

There is no impact to the security of the main VPN connection, as network traffic were observed to be established using at least TLS 1.2, which offers improved security, when the user is connected to the VPN.

Note: *The speed test option is not available when connected to the VPN.*

Retest (2022-02-21)

During the retest, the `MinVersion` attribute has been defined explicitly to TLS 1.3. Hence, this issue was deemed to be sufficiently remediated, and has been marked as 'Closed'.

Insecure .NET Remoting Services Configured

Windows Application	INFORMATIONAL
Issue Status	CLOSED

The ExpressVPN application was observed to use .NET Remoting services for inter-process communication (IPC).

The use of `BinaryServerFormatterSinkProvider` class with the `TypeLevelFilter` set to 'Full' allowed arbitrary deserialization of objects sent by clients.

However, it was observed that the named pipes:

- cannot be exploited by another user account on the same machine
- were not remotely accessible

As the attacker already possesses the ability to run system commands with the local user's privileges, exploitation of this issue does not provide the attacker with any additional access rights or privileges. Hence, the likelihood of exploitation is deemed to be low.

Retest (2022-02-21)

During the retest, it was observed that the affected resource was removed. It was further clarified with ExpressVPN that the application utilised named pipes directly without .NET Remoting. Hence, this issue was deemed to be sufficiently remediated, and has been marked as 'Closed'.

Insufficient Windows PE Binary Protections

Windows Application	INFORMATIONAL
Issue Status	CLOSED

Microsoft Windows offers mitigation techniques to make it harder to exploit software vulnerabilities reliably.

Mitigations such as Address Space Layout Randomization (ASLR), Data Execution Prevention (DEP), Safe Structured Exception Handling (SafeSEH), Control Flow Guard (CFG) and High Entropy VA have made reliably exploiting any vulnerabilities that do exist far more challenging.

The executables in use by ExpressVPN were found to be missing various binary protections, namely SafeSEH, CFG and High Entropy VA. The generation of binary files without these mitigating techniques makes it easier to exploit software vulnerabilities reliably. Other binary protections such as ASLR and DEP were observed to be implemented for all executables.

This issue was raised due to deviation from security best practices; there was no successful exploitation achieved during the time-boxed assessment.

Retest (2022-02-21)

During the retest, it was observed that appropriate binary protection configurations were applied to the affected files. Hence, this issue was deemed to be sufficiently remediated, and has been marked as 'Closed'.

Insufficient Windows DLL Binary Protections

Windows Application	INFORMATIONAL
Issue Status	CLOSED

Microsoft Windows offers mitigation techniques to make it harder to exploit software vulnerabilities reliably.

Mitigations such as Address Space Layout Randomization (ASLR), Integrity Checks, Data Execution Prevention (DEP), Structured Exception Handling (SEH), Control Flow Guard (CFG), High Entropy VA and DLL Signing have made reliably exploiting any vulnerabilities that do exist far more challenging.

The DLL libraries in use by ExpressVPN were found to be missing various binary protections, namely ASLR, DEP, SafeSEH, CFG and High Entropy VA. The generation of binary files without these mitigating techniques makes it easier to exploit software vulnerabilities reliably. Other binary protections such as DLL signing was observed to be implemented for all DLL libraries.

This issue was raised due to deviation from security best practices; there was no successful exploitation achieved during the time-boxed assessment.

Retest (2022-02-21)

During the retest, it was observed that appropriate binary protection configurations were applied to the affected files. Hence, this issue was deemed to be sufficiently remediated, and has been marked as 'Closed'.

Insecure Functions Used

Windows Application	INFORMATIONAL
Issue Status	CLOSED

The application's source code was observed to utilise insecure C and C++ functions which does not safely validate the input parameters (i.e. does not perform bounds checking).

While this increases ExpressVPN's susceptibility to buffer overflow vulnerabilities, application exceptions or undefined behaviours, it was determined that the values passed to these functions have been safely validated (i.e. Null-terminated strings) and not user-controllable. Thus, this issue was raised due to deviation from secure coding practices.

Retest (2022-02-21)

During the retest, the affected functions in the source code were either replaced with function that safely validate parameters or were removed. Hence, this issue was deemed to be sufficiently remediated, and has been marked as 'Closed'.

Overly Permissive Folder Permissions

Windows Application	INFORMATIONAL
Issue Status	CLOSED

Excessive permissions were observed to be granted to the `C:/ProgramData/ExpressVPN` folder.

This misconfiguration could allow a malicious user to read, modify and delete all files in the affected folder, such as executables, library files and configuration files.

In the context of the security assessment, a low privilege malicious user may be able to replace and overwrite files in the affected folder, due to the excessive folder permissions granted.

However, there were no vulnerabilities identified which can be exploited by a low-privileged user during the time-boxed assessment. Thus, this issue was raised due to deviation from security best practices.

Retest (2022-02-21)

During the retest, write access was observed to the affected folder was only granted to users belonging to the privileged 'Administrators' group. Hence, this issue was deemed to be sufficiently remediated, and has been marked as 'Closed'.

Insufficient Windows PE Binary Protections – External Dependencies

Windows Application	INFORMATIONAL
Issue Status	OPEN

Issues pertaining to binaries involving external dependencies were not remediated during the course of the retest conducted on 21 February 2022. The following statement was provided by ExpressVPN:

"As is common practice across the industry, ExpressVPN utilizes open source software in its products to provide common functionality, such as various .NET libraries and others. As these libraries are not currently compiled by ExpressVPN, the ability to add non-standard compiler options is limited. ExpressVPN is aware of this concern, and has taken steps to ensure the integrity and safety of the binaries we utilize within our applications to maintain the security of our applications and users. Where possible, we use strong named assemblies, pin the public key into our project and pin the DLL version. We are also working towards controlling the compilation of all open source binaries in future."

Microsoft Windows offers mitigation techniques to make it harder to exploit software vulnerabilities reliably.

Mitigations such as Address Space Layout Randomization (ASLR), Data Execution Prevention (DEP), Safe Structured Exception Handling (SafeSEH), Control Flow Guard (CFG) and High Entropy VA have made reliably exploiting any vulnerabilities that do exist far more challenging.

The executables in use by ExpressVPN were found to be missing various binary protections, namely SafeSEH, CFG and High Entropy VA. The generation of binary files without these mitigating techniques makes it easier to exploit software vulnerabilities reliably. Other binary protections such as ASLR and DEP were observed to be implemented for all executables.

This issue was raised due to deviation from security best practices; there was no successful exploitation achieved during the time-boxed assessment.

Insufficient Windows DLL Binary Protections – External Dependencies

Windows Application	INFORMATIONAL
Issue Status	OPEN

Issues pertaining to binaries involving external dependencies were not remediated during the course of the retest conducted on 21 February 2022. The following statement was provided by ExpressVPN:

"As is common practice across the industry, ExpressVPN utilizes open source software in its products to provide common functionality, such as various .NET libraries and others. As these libraries are not currently compiled by ExpressVPN, the ability to add non-standard compiler options is limited. ExpressVPN is aware of this concern, and has taken steps to ensure the integrity and safety of the binaries we utilize within our applications to maintain the security of our applications and users. Where possible, we use strong named assemblies, pin the public key into our project and pin the DLL version. We are also working towards controlling the compilation of all open source binaries in future."

Microsoft Windows offers mitigation techniques to make it harder to exploit software vulnerabilities reliably.

Mitigations such as Address Space Layout Randomization (ASLR), Integrity Checks, Data Execution Prevention (DEP), Structured Exception Handling (SEH), Control Flow Guard (CFG), High Entropy VA and DLL Signing have made reliably exploiting any vulnerabilities that do exist far more challenging.

The DLL libraries in use by ExpressVPN were found to be missing various binary protections, namely ASLR, DEP, SafeSEH, CFG and High Entropy VA. The generation of binary files without these mitigating techniques makes it easier to exploit software vulnerabilities reliably. Other binary protections such as DLL signing was observed to be implemented for all DLL libraries.

This issue was raised due to deviation from security best practices; there was no successful exploitation achieved during the time-boxed assessment.


F-Secure