Threat Modelling – Cisco OpenConnect VPN (September 2019)

SaiKiran Uppu, Debolina De, Arvind P Jayan

Abstract— OpenConnect had been initially made to support AnyConnect SSL VPN by Cisco. Now it also supports PaloAlto Networks Global Protect SSL VPN and Juniper SSL VPN which is also called as Pulse Connect secure. Any VPN Server that uses standard TLS/SSL, ESP and DTLS Protocols can be connected to using OpenConnect. This paper identifies the architecture of the OpneConnect VPN and analyzes the possible threats associated with its different components which is necessary to ensure that the present version complies with necessary security requirements.

Index Terms— VPN, SSL, HTTPS, LDAP, X509 Certificate, Cookie Tampering

1 Introduction

VPN is a service that creates an secure encrypted connection between the client and the server over an open unsecure network. As large number of companies are becoming mobile, employees need to connect to their systems over unsecured networks remotely, hence they are susceptible to use VPN to access their data. Attackers can target these services to breach the security of the company's network and access their private data. Hence, this became our primary motivation to assess threat scenarios related to an open-source VPN. Therefor we are doing threat modelling for imminent insecurities on our chosen open-source VPN Client – OpenConnect.

2 Architecture Overview

Open connect is an SSL VPN Client for secure communication. It connects through HTTP and SOCKS5 proxy including support for automatic configuration of proxy using libproxy library. It can detect both IPv6 and IPv4 addresses and routes.

The connection happens in two different phases: HTTPS connection for user authentication either by a certificate, password or secure id. Once authentication is done, an authentication cookie as a response allows to make the actual VPN Connection. This cookie in the second phase, connect to a tunnel using HTTPS through which data packets can be passed over. UDP tunnel can also be configured in case of UDP Traffic, which can be disabled if required.

2.1 Threat Model Diagram

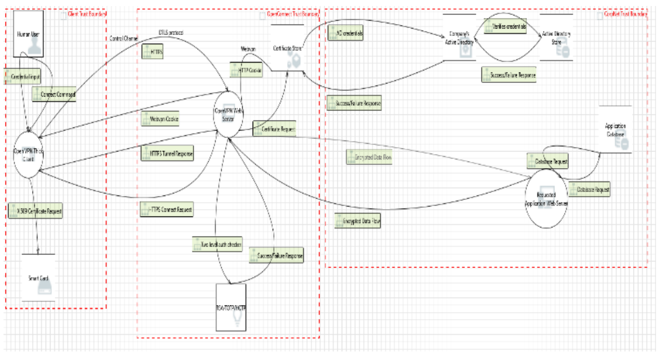
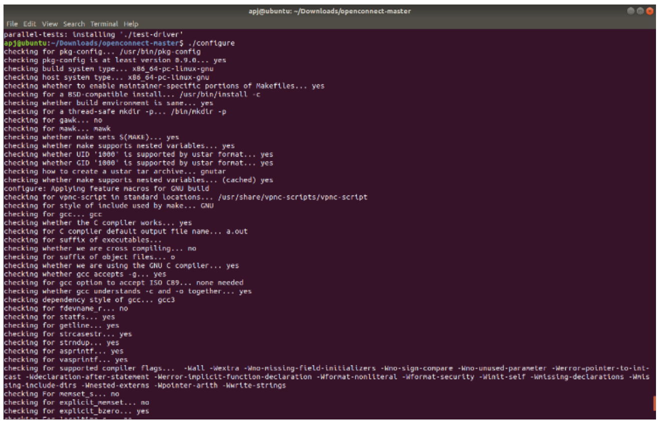


Figure 2: Threat Modelling Diagaram for Cisco OpenConnect VPN

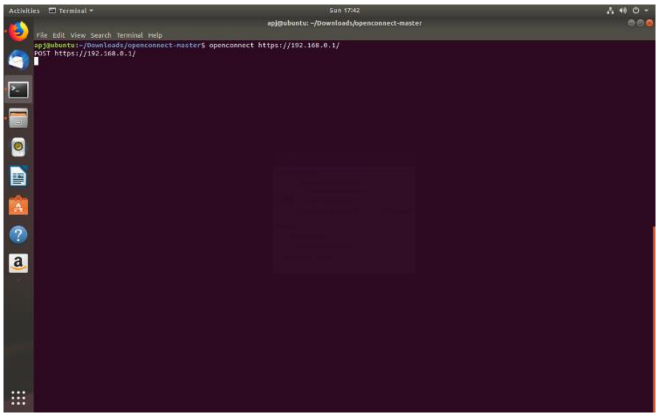
3 Application Analysis

To understand the working and connection flow of the OpenConnect VPN, we took the binaries and compiled them and installed the OpenConnect VPN as shown in the Figure below:


apj@ubuntu: ~/Downloads/openconnect-master
File Edit View Search Terminal Help

```
parallel-tests: installing './test-driver'
apj@ubuntu:~/Downloads/openconnect-master$ ./configure
checking for pkg-config... /usr/bin/pkg-config
checking pkg-config is at least version 0.9.0... yes
checking build system type... x86_64-pc-linux-gnu
checking host system type... x86_64-pc-linux-gnu
checking whether to enable maintainer-specific portions of Makefiles... yes
checking for a BSD-compatible install... /usr/bin/install -c
checking whether build environment is sane... yes
checking for a thread-safe mkdir -p... /bin/mkdir -p
checking for gawk... no
checking for mawk... mawk
checking whether make sets $(MAKE)... yes
checking whether make supports nested variables... yes
checking whether UID '1000' is supported by ustar format... yes
checking whether GID '1000' is supported by ustar format... yes
checking how to create a ustar tar archive... gnutar
checking whether make supports nested variables... (cached) yes
configure: Applying feature macros for GNU build
checking for vpnc-script in standard locations... /usr/share/vpnc-scripts/vpnc-script
checking for style of include used by make... GNU
checking for gcc... gcc
checking whether the C compiler works... yes
checking for C compiler default output file name... a.out
checking for suffix of executables...
checking whether we are cross compiling... no
checking for suffix of object files... o
checking whether we are using the GNU C compiler... yes
checking whether gcc accepts -g... yes
checking for gcc option to accept ISO C89... none needed
checking whether gcc understands -c and -o together... yes
checking dependency style of gcc... gcc3
checking for fdevname_r... no
checking for statfs... yes
checking for getline... yes
checking for strcasestr... yes
checking for strndup... yes
checking for asprintf... yes
checking for vasprintf... yes
checking for supported compiler flags...  -Wall -Wextra -Wno-missing-field-initializers -Wno-sign-compare -Wno-unused-parameter -Werror=pointer-to-int-
cast -Wdeclaration-after-statement -Werror-implicit-function-declaration -Wformat-nonliteral -Wformat-security -Winit-self -Wmissing-declarations -Wmis
sing-include-dirs -Wnested-externs -Wpointer-arith -Wwrite-strings
checking for memset_s... no
checking for explicit_memset... no
checking for explicit_bzero... yes
```

Figure 2 Compilation Steps to build Openconnect VPN Tool


Activities
Terminal
Sun 17:42
apj@ubuntu: ~/Downloads/openconnect-master
File Edit View Search Terminal Help

```
apj@ubuntu:~/Downloads/openconnect-master$ openconnect https://192.168.0.1/
POST https://192.168.0.1/
```

Figure 3 Executing OpenConnect Tool

4 Project execution

With this project, we urge our readers to think about the potential security concerns with OpenConnect VPN. To our knowledge, security threats of this service have not received enough attention. To investigate the security issues further, we begin by relating to documents and codes that are available. We focus on the operations of OpenConnect VPN in the Linux platform and test the possible imminent threats. Based upon the understanding of the segmented functions in the product, we create a threat model to get a detailed analysis of the architecture of OpenConnect. We decompose the application further to identify, document and rate the threats concerning each attribute. In doing so, more useful details unveil, and we gain an understanding of the overall Security of the System.

Threat Modeling Process:

1) Firstly, we identified the critical assets that the functional system must protect.
2) We used Microsoft threat modeling tool to document the architecture of OpenConnect, including subsystems, related subsidiaries, trust boundaries and the organized data flow.
3) Then we decompose the architecture further, to include the underlying network, protocols used, and the host environment design, to create a security profile for OpenConnect VPN. The aim of this profile was to uncover and identify vulnerabilities in the design, implementation and deployment configuration present in OpenConnect service.
4) We now identify the threats that could be associated to the application, with respect to an attacker and his possible outplay. The knowledge of the architecture and potential vulnerabilities of the application will help us in this process.
5) We now document each threat using an available threat template that defines important attributes to be captured for each identified threat.
6) We then rate these threats from most significant to least significant as per priority. The most significant threats present the most severe risk, and the rating process maps the probability of the threat against damage that could take place and hence this provides a more accurate standard.

5 Project observation

OpenConnect is a VPN that run on HTTP proxy and SOCKS5 proxy and it does several authentications including SSL certificates and TOTP/HOTP software tokens. Even though the service contains a lot of seemingly foolproof security mechanisms, we were able to develop a threat model for the application. The possibility of vulnerability still exists for any application, and by proceeding with threat modelling we were able to identify the assets that are to be considered for an attack surface like the users, web VPN server, thick client, OTP modules. We were able to understand the function overview of the OpenConnect service, we identified the 3 trust boundaries, the dataflow between the thick client, OpenConnect webserver and the application web server. We had a deep insight on the various security mechanisms used by the service. By threat modeling we were able to identify all the possible threats including spoofing, parameter manipulation and DOS and the most significant attack pattern possible for such a broad system.

By rating the threats, we discovered with OpenConnect service, we identified that one of the high-risk threats is spoofing the webvpn cookie provided by the VPN server. An attacker spoofing the webvpn cookie gain privileges of the client who is being granted the cookie. This allows a malicious attacker to obtain or mutate a company's private data through the tunnel. There are other threats of lesser risks such as weak credential storage, which even though if the attacker gains access to, it will be hard to obtain passwords from their hashed values.

6 Conclusion

We were able to successfully identify the threats associated with the selected target using Microsoft threat modelling tool. Also, we were able to classify the theats associated to OpenConnect VPN and rate them based on severity. These threats can be further generalized and can be associated to other VPNs like open-source commercial software OpenVPN, OpenSwan VPNs. All these products can also be susceptible to the same attack patterns due to the similar functionalities between the VPNs.

Acknowledgment

The authors wish to thank Reuben Johnston for his support for the successful completion of the Threat Modelling on the OpenConnect VPN using Microsoft Threat Modelling tool.

REFERENCES

[1] Open Connect Tool Getting Started, https://www.infradead.org/openconnect/building.html.

[2] OpenConnect VPN Tool Features, https://www.infradead.org/openconnect/features.html

[3] SmartCard integration with OpenConnect, https://www.infradead.org/openconnect/pkcs11.html

[4] OpenConnect 8.00 release". Lists.infradead.org. Retrieved 2019-01-05.

[5] "VPN Overview". openwrt.org. Retrieved 2018-03-15

[6] Virtual Private Network system and methods, https://patents.google.com/patent/US6055575A/en

Key Terms:

Threat: Threat is a possible risk that might exploit a vulnerability to breach security.

Vulnerability: A flaw in a system that leaves information expose to a threat.

Attack or Exploit: Code or data which takes advange of a vulnerability in a system for malicious intent.

Countermeasure: It is a mechanism for reducing the attack surface.

VPN: Virtual Private Network extends private network across a public network and helps users to access or mutate data over an open unsecured network.