Opportunistic IPsec development using LetsEncrypt

Background

Opportunistic IPsec is an attempt to encrypt the internet at large. The idea is to build VPN tunnels directly to all internet hosts irrespective of the communication used. An initial proof of concept was created that leverages LetsEncrypt certificates for use with IKE and IPsec. The goal of this project is to turn this proof of concept into production quality code that makes it trivial to enrol and deploy on any server and any client. All the data flow between client and server should be encrypted ie. all the communication between the hosts should be encrypted and not just the communication between the two applications on the hosts.

Objectives

The project has the following objectives:

- Automate Client and Server configuration.
- Automate the enrollment and updating of LetsEncrypt certificates for use with libreswan.
- Implement a DNS / DNSSEC based method for advertising Opportunistic IPsec support using LetsEncrypt for servers.
- Design a non-DNS based method for publishing support of LetsEncrypt based Opportunistic IPsec for servers.
- Enhance libreswan to allow a secure fallback to the existing NULL Authentication method.
- Ensure the working of two behind the same NAT(Network Address Translator). Along with ensuring the working of two clients behind NAT, with one not using IPsec.
- Support the recovery from server OR client reboot/crashes. Along with ensuring no lockouts happen.

Technical - Components/Modules

This project has been divided into *6 components*, which are as follows:

1. **Automation of Client Configuration:**

   Creating scripts for(a - e):

   a. Installing libreswan(if not pre-installed), downloading the root CA file for LetsEncrypt certificates.

   b. Installing the LetsEncrypt CA certificates into the NSS DB(different locations for different Linux Distro).

   c. Configuring Libreswan by downloading and storing different required connection profiles eg "private-or-clear”.

   d. Testing and storing if a host supports Opportunistic encryption OR not.

   e. Restarting and testing if the configuration worked. It will test each time configuration is updated OR when the test is run by the user.

2. **Automation of Server Configuration:**

   Creating scripts for(a - f):

   a. Installing libreswan and apache/httpd(if not pre-installed) along with installing dehydrated
   b. Getting LetsEncrypt certificate through dehydrated. Along with allowing the user to enter configuration details.
   c. Confirming the success of the above steps and reporting it to the user.
   d. Generating and importing certificates into libreswan using OpenSSL configuration.
   e. Configuring Libreswan by downloading and storing different required connection profiles eg "clear-or-private".
   f. Restarting and testing if the configuration worked. It will test each time configuration is updated OR when the test is run by the user.

3. **Automation with LetsEncrypt certificates:**

   Creating scripts for(a - b):

   a. Enrollment of LetsEncrypt certificates using dehydrated.
   b. Updation(through dehydrated) and redeploying(in libreswan) the LetsEncrypt certificates.

4. **Add support to ensure the working of two behind the same NAT.**
5. **Add support for the working of two clients behind NAT, with one not using IPsec.**
6. **Support the recovery from server OR client reboot/crashes. Along with ensuring no lockouts happen.**

Note - In automation of Server, Client & LetsEncrypt configuration there will only be *a single configuration script*, that will call other scripts which will perform different required tasks(Reference). The user will not be required to run each and every script independently.

Timeline

In this, the project is divided into *5 stages*. Each stage constitutes various modules/components and is distributed over *12 weeks period(as per GSoC timeline)*.

Project Timeline		
Stage no.	**Stage Specifications**	**Time required**
Stage 1	Automation of Server & Client Configuration.	4 Weeks
Stage 2	Automation with LetsEncrypt certificates.	2 Weeks

Stage 3	Adding support for the working of two clients behind NAT, with one not using IPsec.	2 Weeks
Stage 4	Supporting the recovery from server OR client reboot/crashes. Along with ensuring no lockouts happen.	2 Weeks
Stage 5	Documentation & Debugging.	2 Weeks

EVALUATION

1. **Evaluation 1 [June 24 - 28, 2019]**

Duration	Tasks/Modules/Components
Week (1-4)	Creating scripts for(I - VII): I. Installing libreswan, downloading the root CA file for LetsEncrypt certificates. II. Installing the LetsEncrypt CA certificates into the NSS DB. III. Configuring Libreswan by downloading and storing different required connection profiles. IV. Testing and storing if a host supports Opportunistic encryption OR not. V. Restarting and testing if the configuration is working. VI. Installing libreswan and apache/httpd along with installing dehydrated in Server. VII. Getting LetsEncrypt certificate through dehydrated. Along with allowing the user to enter configuration details.

2. **Evaluation 2 [July 22 - 26, 2019]**

Duration	Tasks/Modules/Components
Week (5-8)	Creating scripts for(I - VI): I. Confirming the success of the above steps(in server configuration) and reporting it to the user. II. Generating and importing certificates into libreswan using OpenSSL configuration. III. Configuring Libreswan by downloading and storing different required connection profiles.

	IV. Restarting and testing if the configuration worked. It will test each time configuration is updated OR when the test is run by the user. V. Enrollment of LetsEncrypt certificates using dehydrated. VI. Updation(through dehydrated) and redeploying(in libreswan) the LetsEncrypt certificates. VII. Add support to ensure the working of two behind the same NAT.

3. **Evaluation 3 [August 19 - 26, 2019]**

Duration	Tasks/Modules/Components
Week (9-12)	I. Add support for the working of two clients behind NAT, with one not using IPsec. II. Support the recovery from server OR client reboot/crashes. Along with ensuring no lockouts happen. III. Creating Documentation of the project. IV. Debugging.

Personal

1. I am a passionate computer Science and Engineering student at one of the prestigious institutes of India - National Institute of Technology Hamirpur (H.P.), India - 177005.
2. I love contributing to Open Source.
3. Enjoy blogging, loves travelling.
4. Linux user.

GitHub, Email, Website

GitHub Username: Rishabh04-02
Email IDs: rishabh0402@gmail.com , cs14mi508@nith.ac.in
Website: https://rishabhchaudhary.in

My Timezone

UTC/GMT +5:30 hours | Asia/Kolkata : IST (*India Standard Time*)

My comfort working with a remotely available mentor

I am comfortable working with a mentor who is several time zones away/located remotely. I have worked in this way before.

Am I selected in GSoC before?

Yes, I have successfully completed GSoC 2018 with **The Libreswan Project** and have developed **Libreswan Managing Interface**.

Tracking my progress

I have divided my work into stages and have developed a weekly schedule. This schedule will help me as well as my mentors to track my progress.

My native language

My native language is Hindi, but I've studied English and I am fluent in it. **English is my preferred language** when working on the project.

Am I likely to finish the project in the allotted time?

1. I've divided the project into stages and in this way, I can achieve my goals in a more convenient/sorted and planned way. Also in this way, it'll be easier for the mentors to track my progress.
2. I've have developed **Libreswan Managing Interface** for the organization in GSoC 2018 and have delivered all the components mentioned in the proposal and some other essential components which were not covered in the proposal. I have worked with OpenSSL certificates in the last project and have a good understanding of the project.
3. Also, the basic tasks required for this project involves *creating python/shell scripts, Working with LetsEncrypt and OpenSSL, creating and modifying configuration files, working with libreswan.* I've worked on all of the above-mentioned tasks, some in my projects(including Libreswan Managing Interface) and on some while contributing to other open source projects(*Reference).*