Welcome to ENPM693 Network Security Project

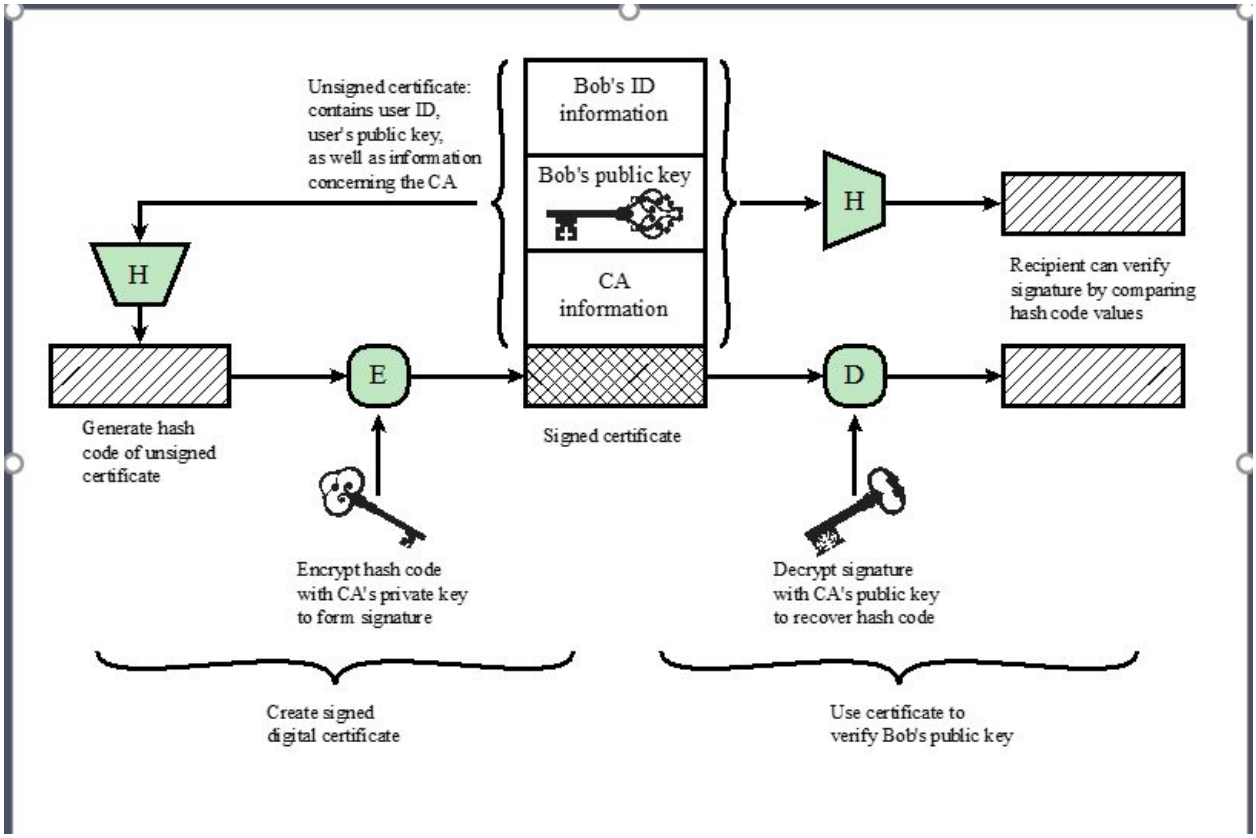
Khushali's guide to Public Key Infrastructure

By Khushali Dalal | December 2, 2017, 10:00 AM EST

Few weeks ago, Dr. Sohraab Sultani dropped a tempting idea in a Network Security class that those students who haven't performed well in midterms can cover up there grades by working on a course project. Of Course, I was super excited not for improving my GPA but, to work on something that is useful in real world. I approached Puyan Dadvar, Sr. Network Engineer at my workplace to help me pinpoint a project topic. He right a way threw many topic ideas on me and the one which caught my mind was PKI. I learned about it in one of Dr. Sultani's class however, there were many unsolved questions hovering my brain like How does pki works? Who has the bunch of public and private keys? On what basis does one get authorised? Can I have one key?

What is PKI?

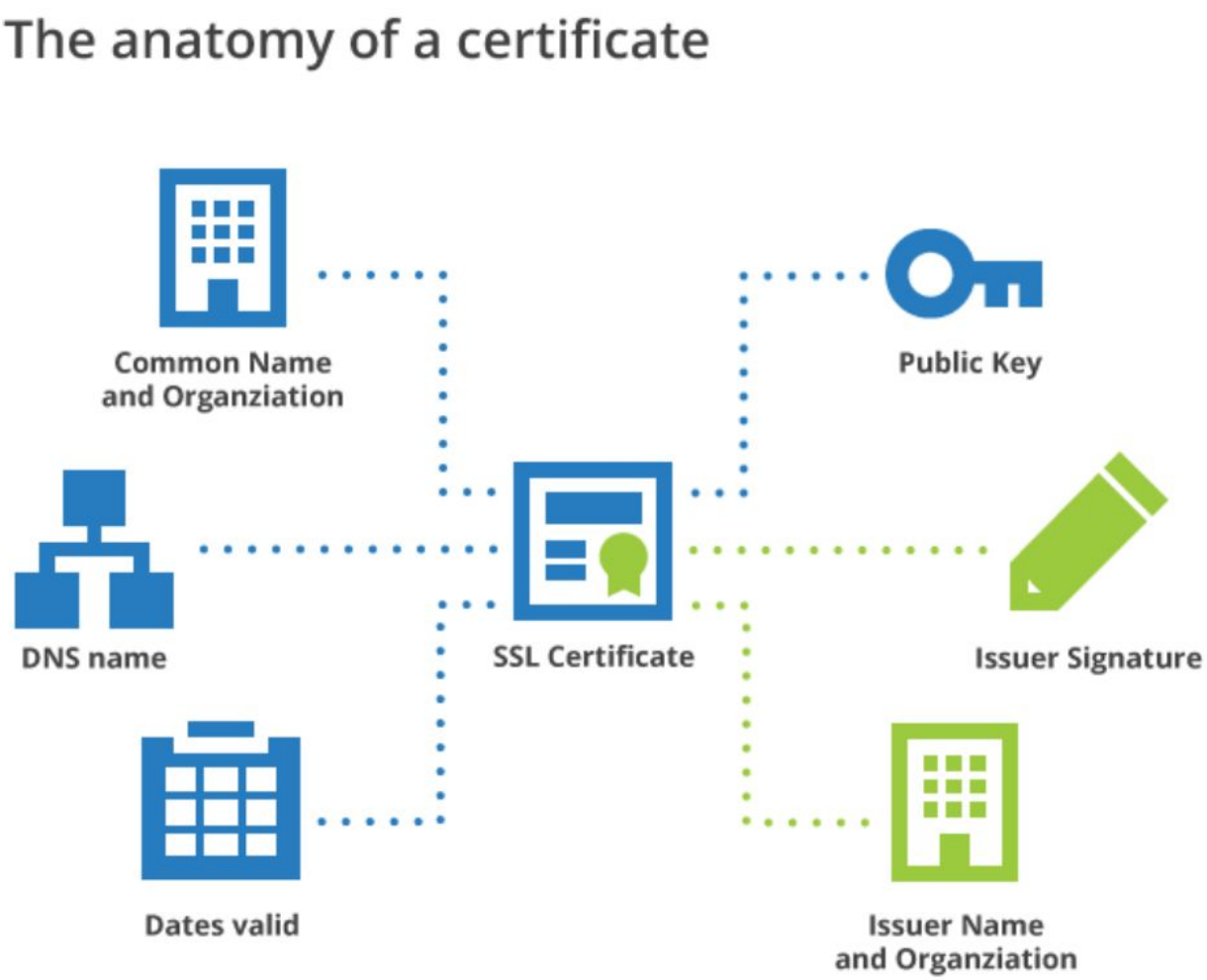
The biggest challenge with encryption is we may have very powerful encryption methods like AES or RSA but, unless both the parties know the key that is used to encrypt data, everything is useless. Public-key cryptography is an asymmetric encryption method where a key pair is used to encrypt and decrypt the data. When Bob wants to send message to Alice, he encrypts the message with Alice's public key and then Alice reads the message by decrypting it with her private key. PKI (Public Key Infrastructure) is used in multiple ways but, primarily for encrypting and signing data. Encryption is nothing but scrambling it in a way that it makes it unreadable except to authorised person.
Digital Signature is created by encrypting hash code with private key. It is used for authenticating both source and data integrity. Hence, the combination of public-key cryptography and digital signature fulfils authentication and confidentiality aspects.



The Anatomy of PKI

Now, let's see how Digital Certificate works. As the name implies Digital Certificate is one form of certificate used in the electronic world. In laymen terms, when I say "I am CCNA Routing and Switching certified" it means, the authority who offers the test Cisco, in this case, has certified me (authorized me) to be eligible of knowing basics of Routing and Switching. Similarly, if I want my website to be certified I need to get a digital certificate from a certificate authority called CA. CA generally checks their basic criteria to be eligible for the certificate and hence issue a certificate for some predefined time period. Digital Certificates are issued to computers, software packages, websites or do anything that we need to prove our identity.

Hence, Certificates are very useful in high-security situations. In the computer world, suppose that I want to securely transfer $500 from my saving account to checking account. How do I confirm that the site on which I am performing transaction activity is actual Capital One bank website and not an imposter? If I see that the URL on bank's website starting with **https://www.capitalone.com** then this is one positive sign that I am on a secure webpage. Secondly, if the certificate issued to Capital one is coming from a legitimate Certificate Authority, for instance, Veri sign then I can be double sure to accept the Capital one's identity to be trustworthy and perform the transaction securely without any hesitation.



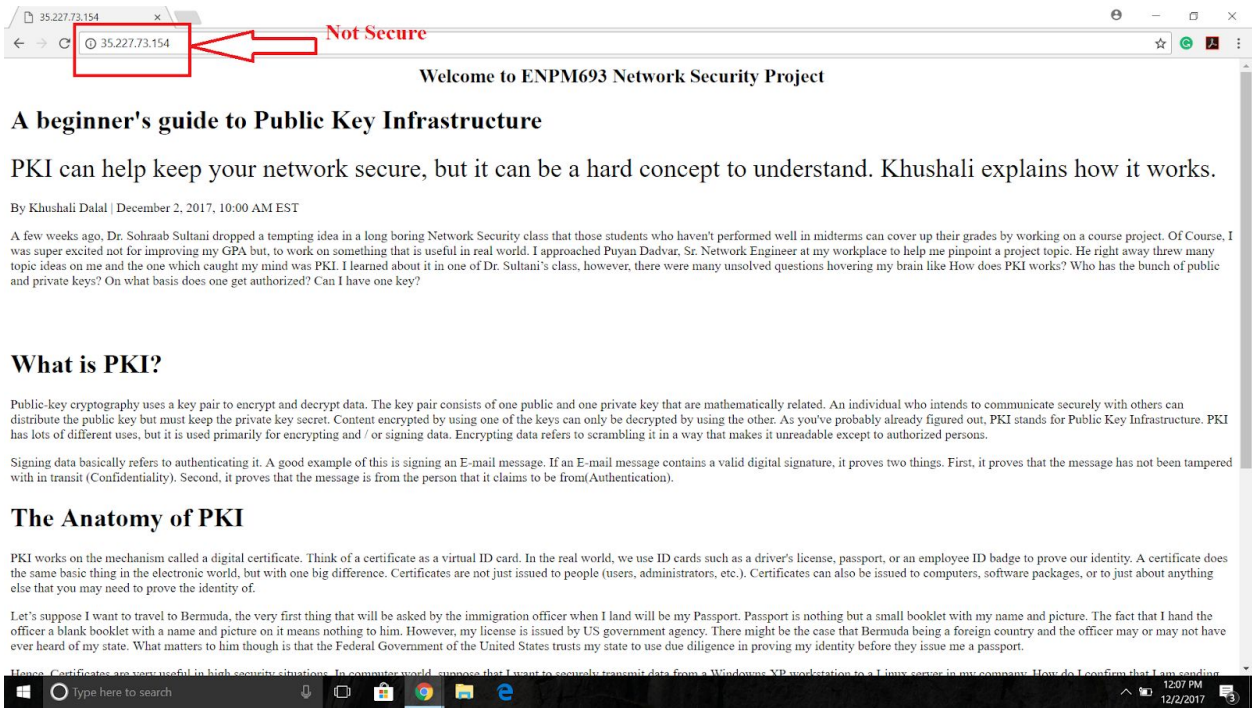
Using PKI to secure my Website

The steps I followed to build my own pki are pretty simple,

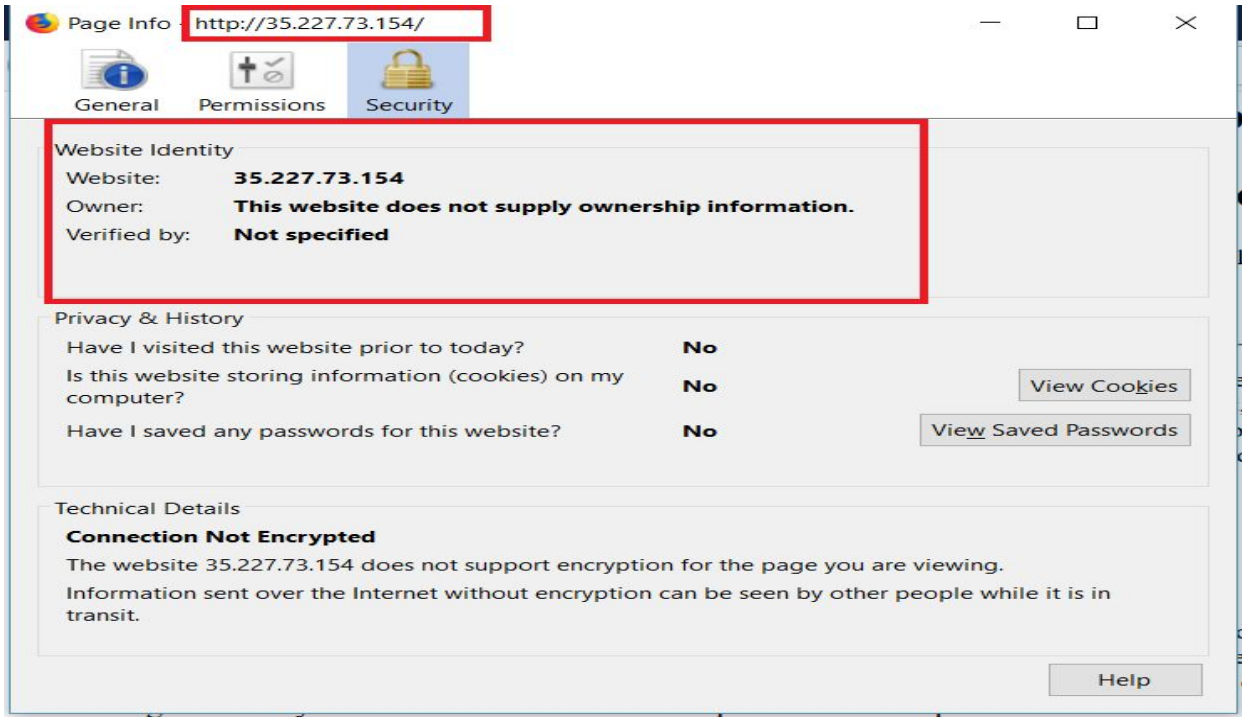
1. Searched a certificate authority (CA) who can provide me certificate for free. Running tight in budget. Let's Encrypt
2. Install an apache web server to get SSL/TLS to activated on it. Google Cloud Platform. Again Google's cloud service is offering free trial with $300 credit for 12 months.
3. Registered a domain name with Name Cheap.
4. Installed a client for certificates and issued a legit free certificate from a CA with the help of Certbot

Implementation Journey

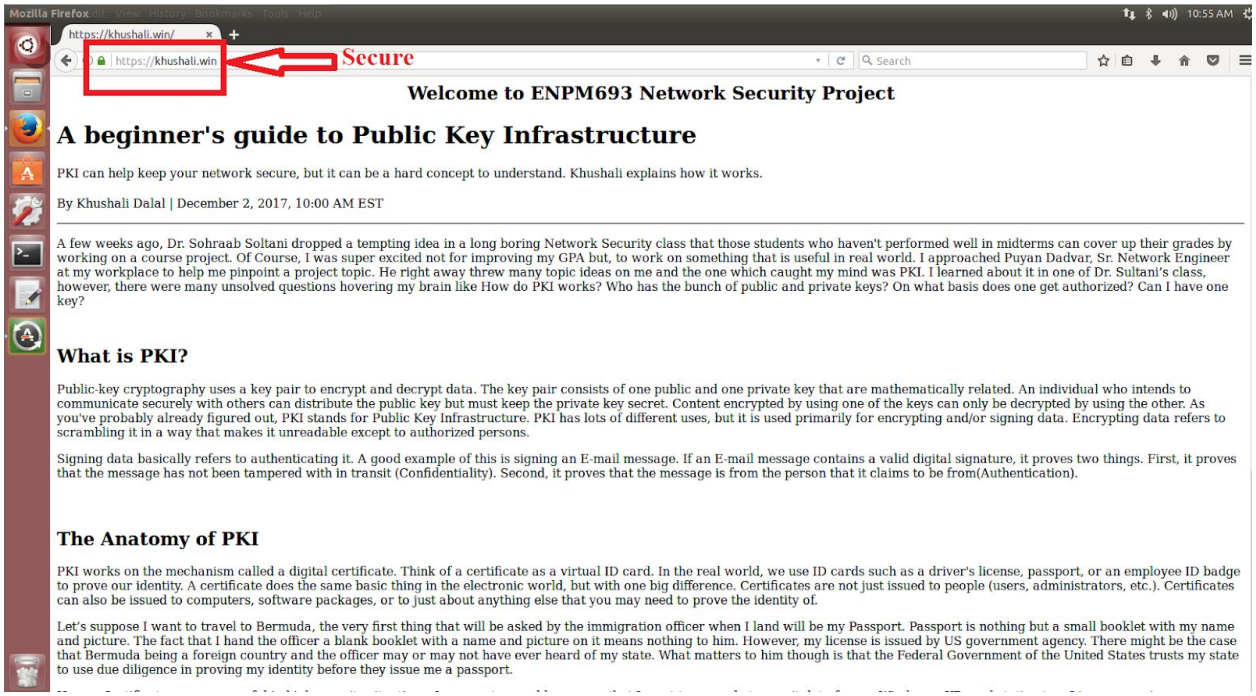
Before obtaining the certificate, I got the webpage as shown below



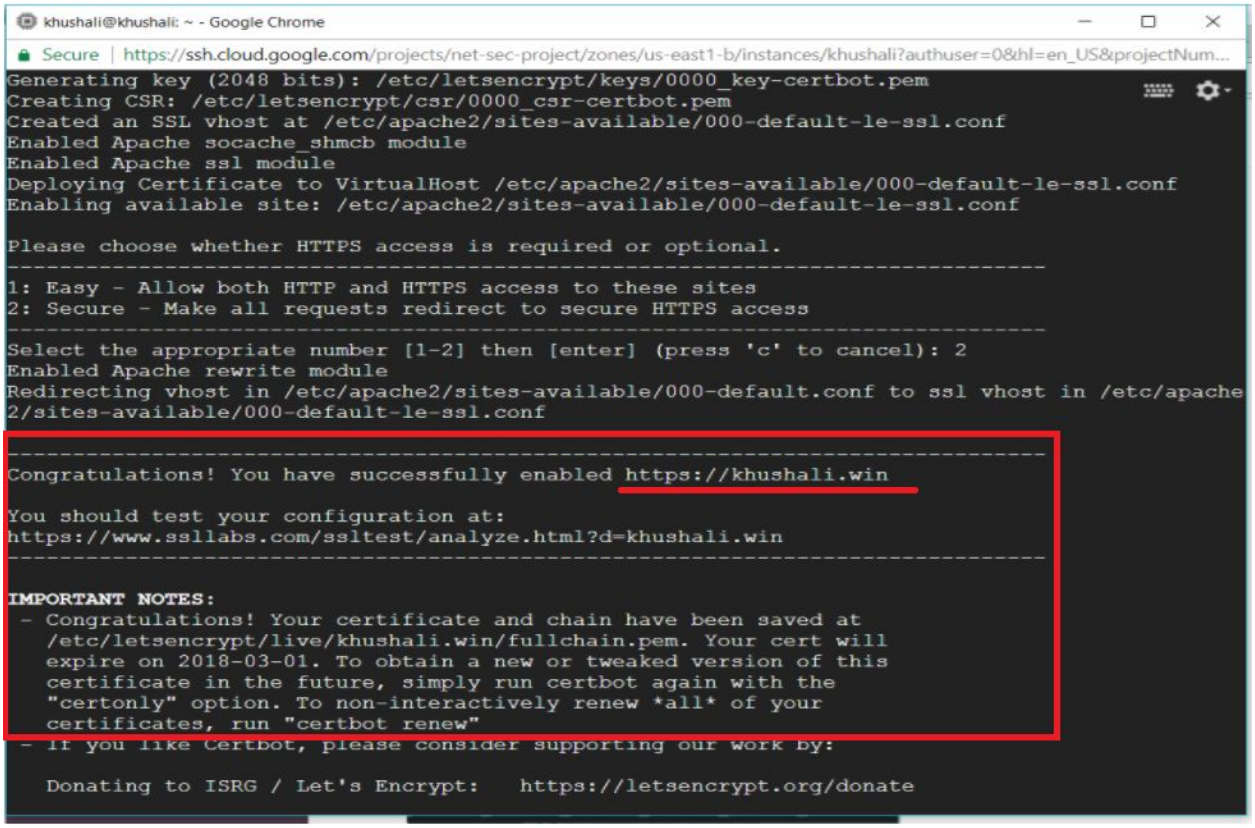
Page information for webpage not being secure



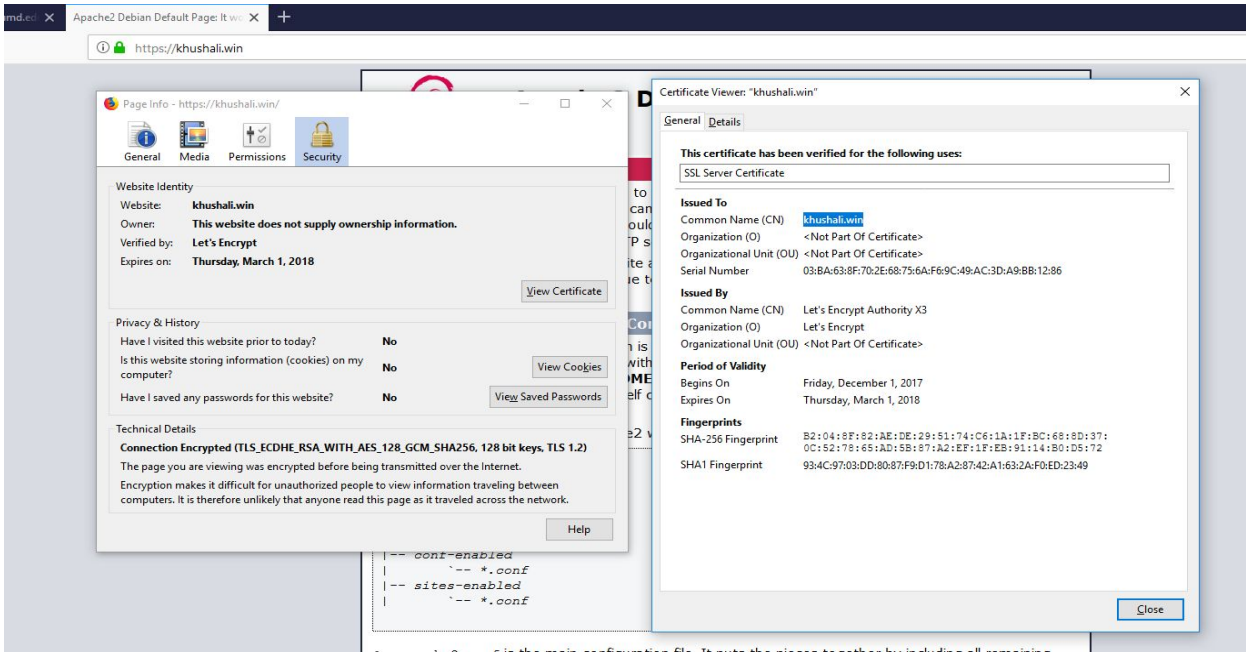
After getting the certificate, we can see that the url changed to **https.**



Confirmation message on enabling https://khushali.win



Certificate information page. [SSL Report]



To put it simply, I successfully created a secure encrypted site using public key infrastructure! In today's world, digital signature and PKI are used in many different applications like used to sign E-mail messages or secure webpage or to sign software packages and at many other places.

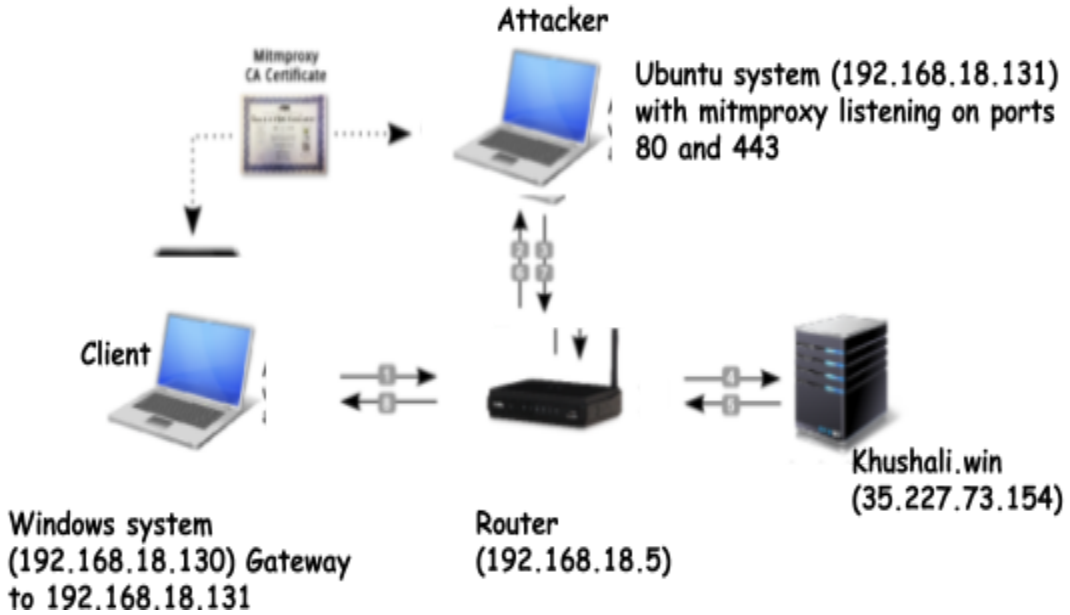
Using Custom Certificate Authority

Enough of legitimate stuff let's talk about fake Certificate Authority and cool hacking techniques. As we saw above once an authority gets legitimate certificate from authorised CA browsers trust the webpage and allow to navigate on that webpage securely. What if I make my own Certificate Authority and issue a fake certificate to my webpage and then eavesdrop all the communication my website does with the server. This scenario describes Man in the Middle Attack aka MITM attack.

Intercepting HTTP and HTTPS Traffic with mitmproxy

We know by now that unencrypted HTTP traffic can be attacked without any certificates and Certificate Authorities. Attacking encrypted HTTPS request and response between client and server is the real challenge. As the data being transferred is encrypted it is hard for a middle man (or a proxy) to decrypt the exchanged data packets.

When Eve wants to sniff into the secure connection between Alice and Bob, he needs to act like a fake Certificate Authority like mitmproxy. Mitmproxy generates certificates on the fly to the hostname needed for the connection. In simple words, if I want to connect to **https://khushali.win ,** mitmproxy generates a certificate for "**https://khushali.win**" and signs it with its own certificate authority. Hence, the I believe that mitmproxy server is in fact "**https://khushali.win**" . The figure shown below makes this scenario more clear by explaining transparent HTTPS proxying.



For this attack, two basic conditions needs to be met

1. Setting Mitmproxy as a standard gateway (HTTP and HTTPS) : In order to intercept the IP packets, the gateway ip for client device needs to be set to the mitmproxy server address.
2. The mitmproxy CA key should be added in the root directory of client's system for HTTPS proxying to work. Meaning, the client (Windows system) should know and trust the mitmproxy CA.

Implementation Path

Below is the screenshot of mitmproxy in my Ubuntu (Host) system

Changing the gateway of Windows (Client) system

Capturing HTTP/HTTPS requests and responses from https://khushali.win

The Certificate Authority changed dynamically from Let's Encrypt to Mitmproxy

To conclude, it is scary to imagine that someone having access to internet exchange points and a private key of a major root CA certificate of a trusted

certificate authority is equivalent to break every TLS/SSL connection passing through the respective interface.