# Secure Medical System

# COMP3433 Term Project

**Group members**

12132031D GENG XU

12131354D ZHANG BEIBEI

13088783D TAI SHUNWA

**1. Requirement Analysis:**

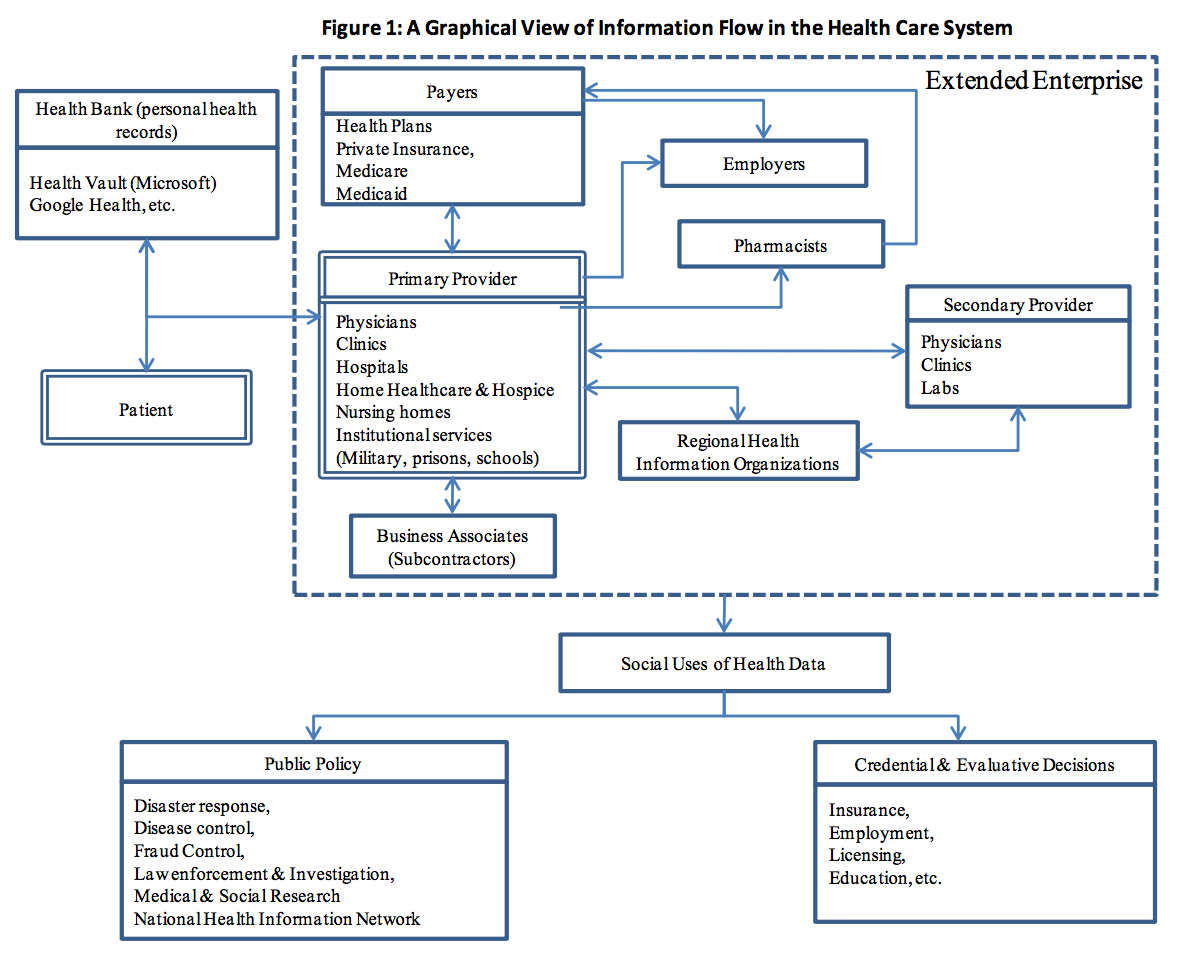
1.1 Research and study:

Security of medical system aims at providing necessary privacy for patients. Privacy is an underlying principle of a patient. Although sometimes patients may need to share adequate information with doctors for correct diagnosis and related assessment, some information should be divulged as a secret. For example, a patient may not want to share his or her marriage status since it is not quite related to the diagnosis. This is about the information confidentiality which is a measurement undertaken to prevent the disclosure of the sensitivity information.

What’s more, some information in the secure medical system should be modified or added only by the authorised person, for example, doctor. A patient record should only be modified by the related doctors and the modification should be consented by the patient. This is about the data integrity which concerns about the consistency, accuracy and trustworthiness of data over the data life cycle.

At last, the data availability should be endured by maintaining the hardware, we should make sure that hardware is fully functioning to support all technics used to build up the secure medical system. It is important to provide adequate communication bandwidth and prevent the occurrence of bottlenecks. Also, it is a must to keep all necessary system upgrades. This is about the availability.

A possible medical information system work flow is shown below:



According to the Privacy and Security Rules under HIPAA, we should build a simplified administrative system and the service should be continuous.

There are a lot of threats about the information privacy system including secure medical system. The threads can be further categorised as organisational threats and systemic threats. Organisational threads can be further categorised into[1]:

* Accidental disclosure
* Insider curiosity
* Data breach by insider
* Data breach by outsider with physical intrusion
* Unauthorised intrusion of network system

The systemic threats are basically from the inside of the information flow chain which means the access control generate some false access problem.

1.2 Scenario:

The administrator can add and delete the information from the secure medical system.

The doctor can login the system by a doctor account and view the adequate amount of information.

The patient can login the system by a patient account and view only the necessary information about him/her self.

Other stakeholders have different account with different access right.

The attacker may do a dictionary attack, a brute force attack, hijacking, XSS attack, CSRF attack and SQL injection to the login page. They aims at violate the confidentiality and integrity of the data.

The attacker may deploy DDoS attack, attack on the system, attack on the database.

1.3 Project requirement:

a. Non functional requirement:

The system should provide an adequate bandwidth.

The system should be intrusive, simple and usable.

The system should be stable and within the whole control of our developers.

b. Functional requirement:

When it comes to different kinds of users, we should have different kinds of user accounts. More specifically, build a secure medical system for doctors and patients’ easy access to their related medical records.(authentication)

When it comes to different user privilege we should let different stakeholder of the system have different information access right. More specifically, doctors can only access to the medical records diagnosed by themselves. Patients can only access to their own medical records. (access control)

The medical system is a web based application. (Web security issue)

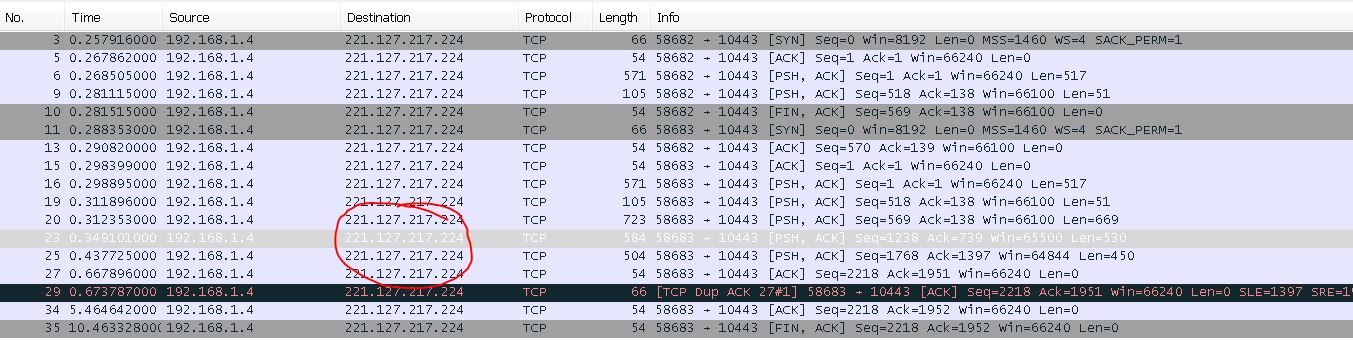
The operating system is secure under the prevention mechanism. (system security)

**2.Appropriate mechanism**

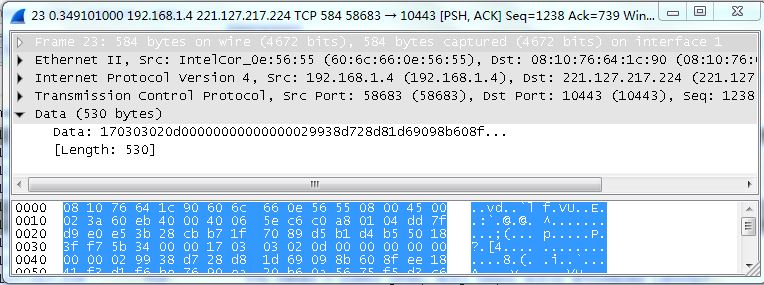
**2.1 network transfer security**

The web application is based on HTTPS protocol, so the channel is secured based on the SSL layer. We use wireshark to capture network packets to examine this:

Packet transfer between client and server:



Data encryption: (post doctor\_id = “D001”&password = ”12345”)



2.2 Authentication:

We use salt hash mechanism to provide more secured authentication process:

Database contains salt value and salt hashed password value for each account

Client will send plain text password via http POST.

The POST request will be encrypted by SSL layer as shown in 2.1

When server receive the password, server will salt hash it and compare with the one stored in database.

2.3 Access control:

Each account will be constrained to access particular tables inside the database, they cannot access unrelated tables under a designed database schema.

Doctor can only access to diagnosis and prescriptions of his patients.

Patients can only access to diagnosis and prescriptions of him own.

2.4 Web security

a. Brute force check on authentication

Check whether the authentication request is brute force in the login function by counting the times the authentication request that server receives.

If brute force attack is detected, the corresponding account will be locked.

b. Authentication encryption

Use salt hash method.

Store both salt and hashed password in the database

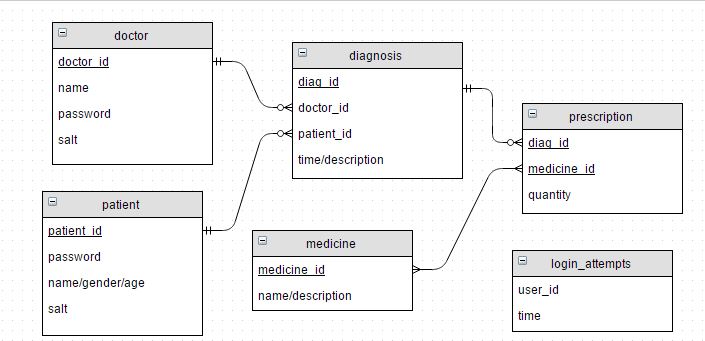
c. Eliminate SQL injection

Using php prepare() function when dealing with the SQL

d. Eliminate Session hijacking – XSS

Force session to use cookie only.

2.5 Database design

ERD: 

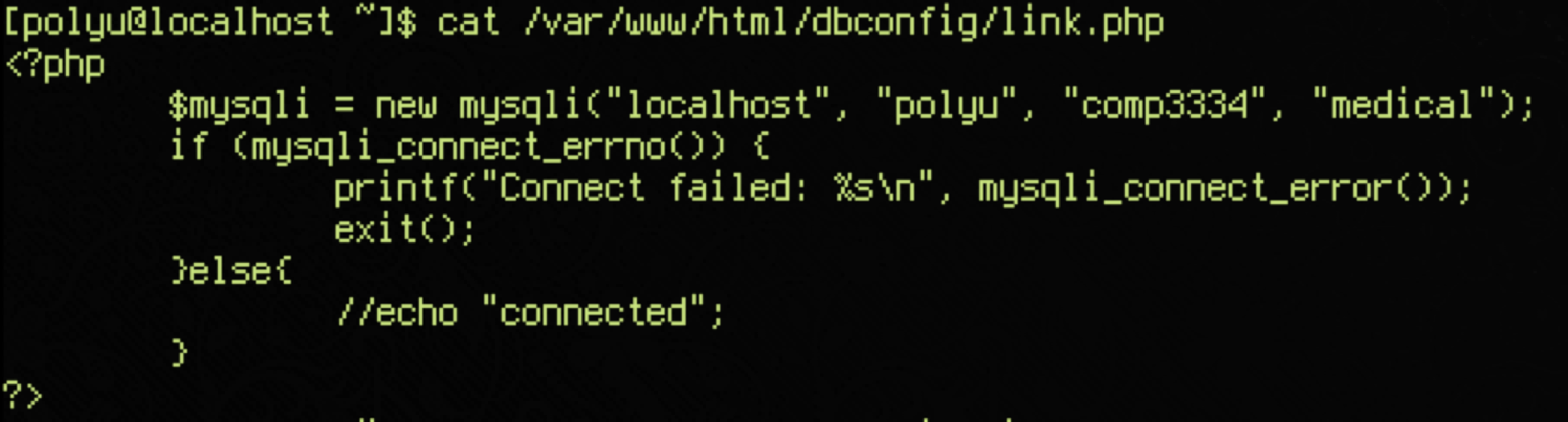
2.6 System security

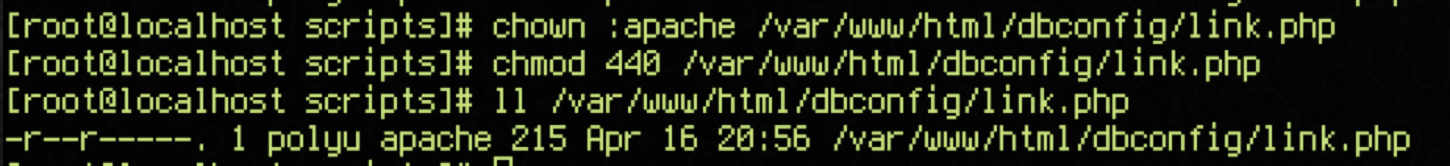
a. Disable the root account[2]

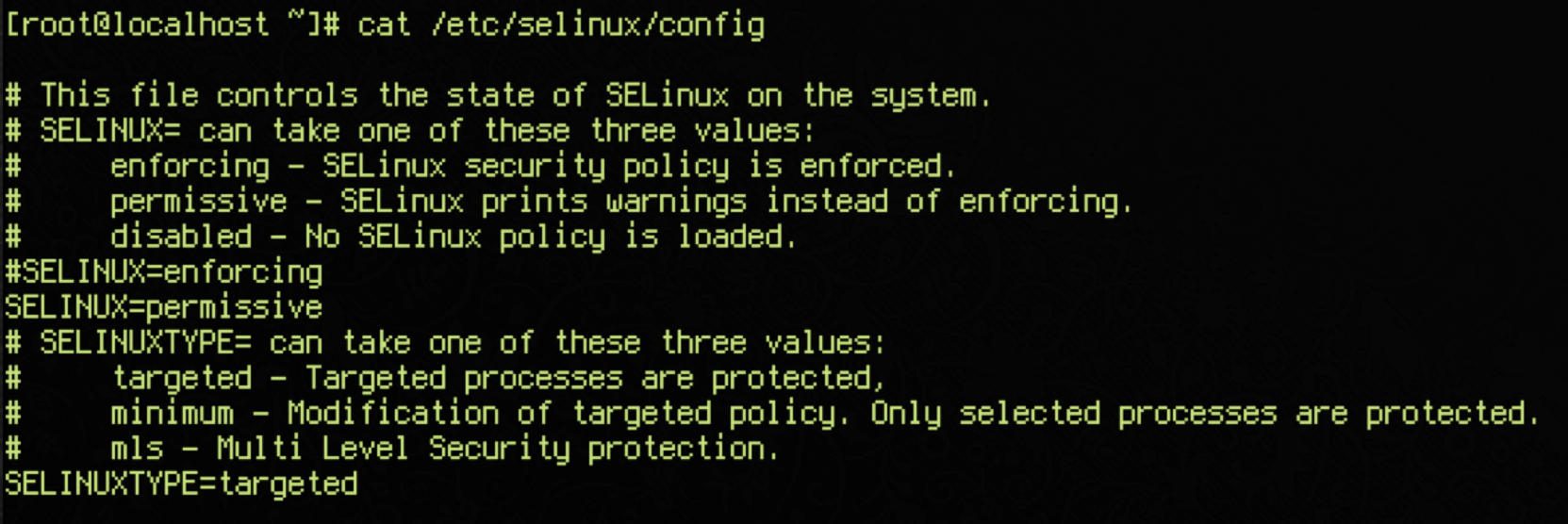
you can see that a “!” pasted before the hashed password means the account is locked. This can avoid the attacker try the root account:

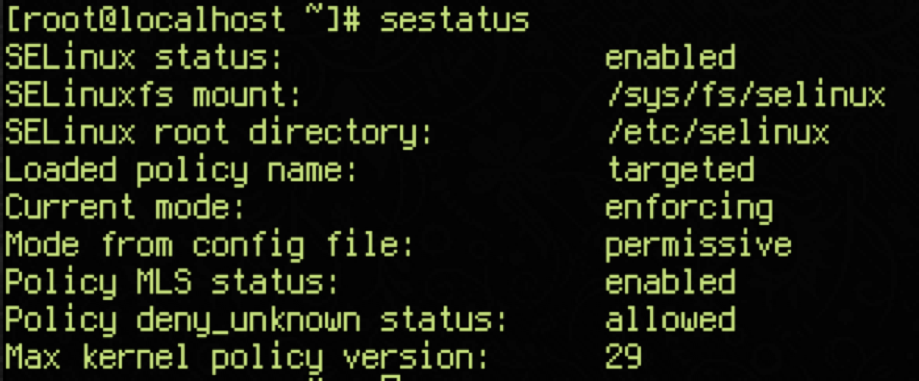
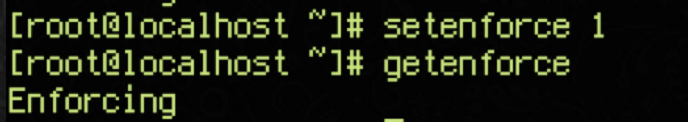
b. Proper permissions on the access control(DAC & MAC)

DAC - set the permission based on the user’s identity

e.g The following php file contain the database username and password:

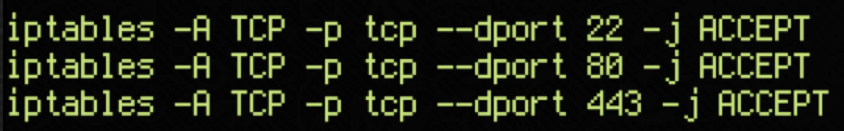
We set the permission to read-only by the owner and group:

MAC - enforcing the SELinux in the system:



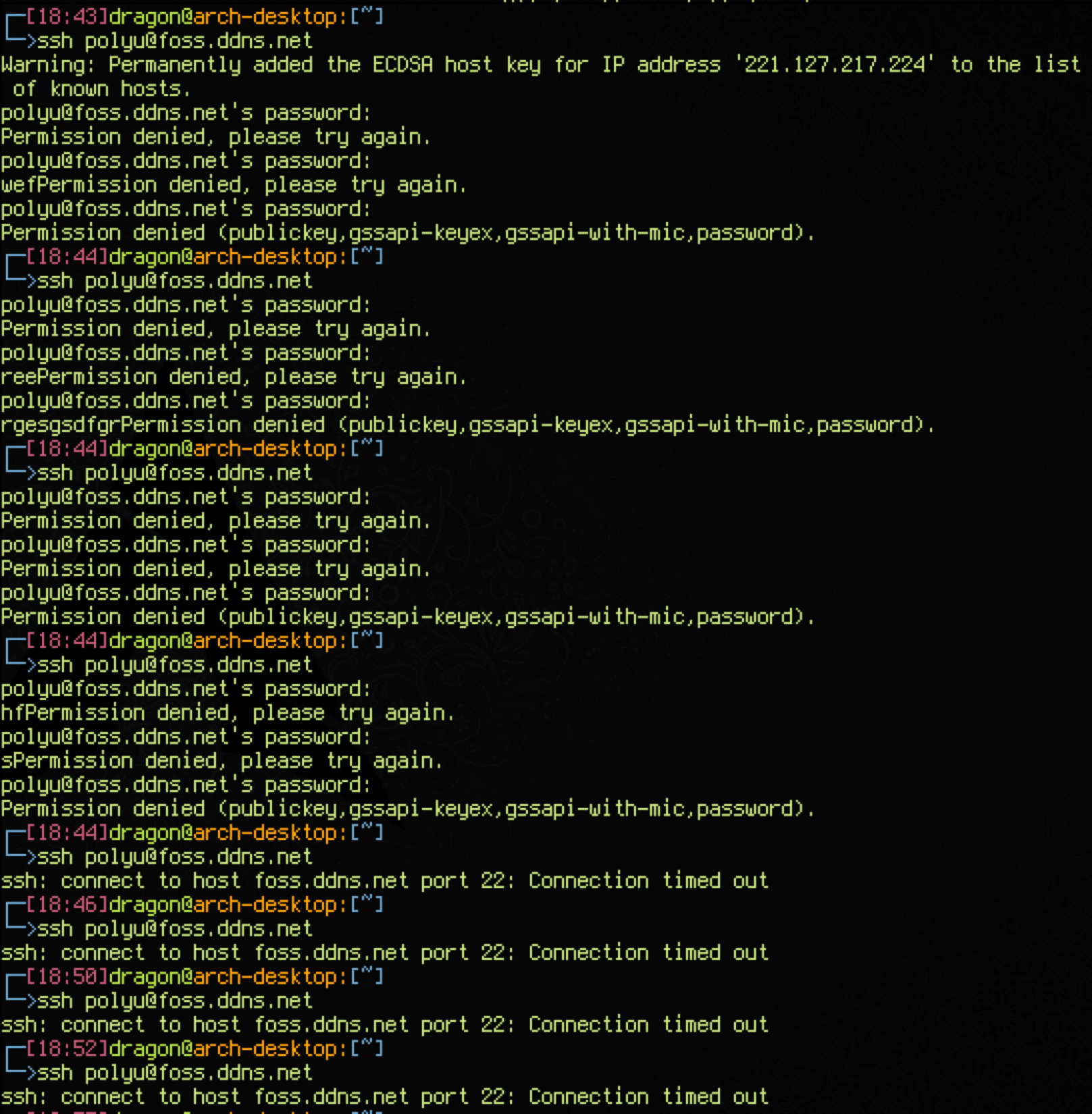
c. Limit the available ports by firewall[3]

First, we drop all the input traffics:

Then limit the ports for public access: only allow the SSH, HTTP, HTTPS:

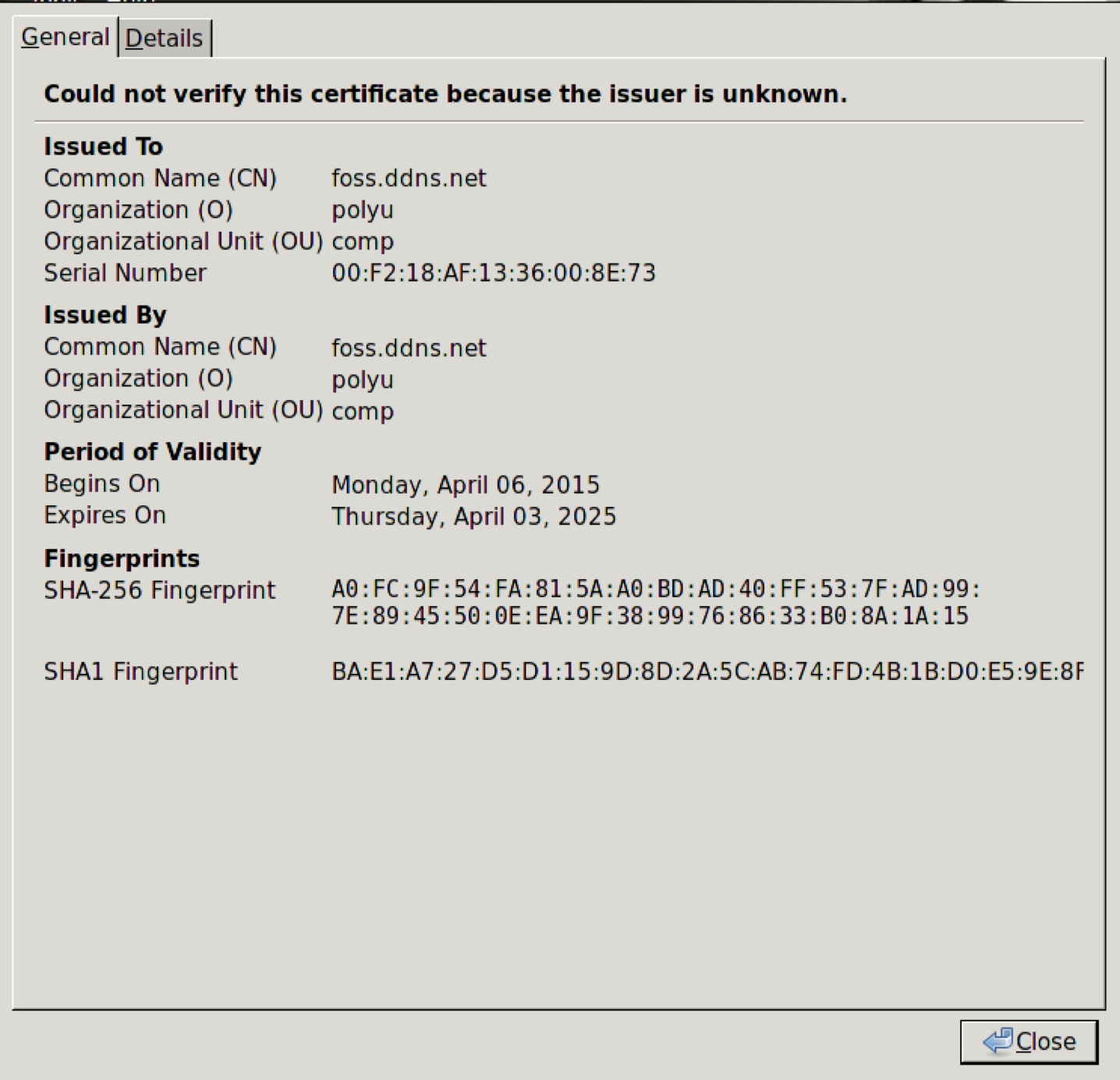
d.Prevent SSH brute force attack by configure the rules on firewall[3]

The following rules can drop the ssh connections when the attack attempted to login several times:

Result: After trying the password a number of time, the connection timed out:

5. HTTPS connection:

We used openssl to generate a self-signed certificate in order to establish

a secured connection :

The following 6 steps are how we make the cert in the system[4]:

>cd /etc/pki/tls/certs

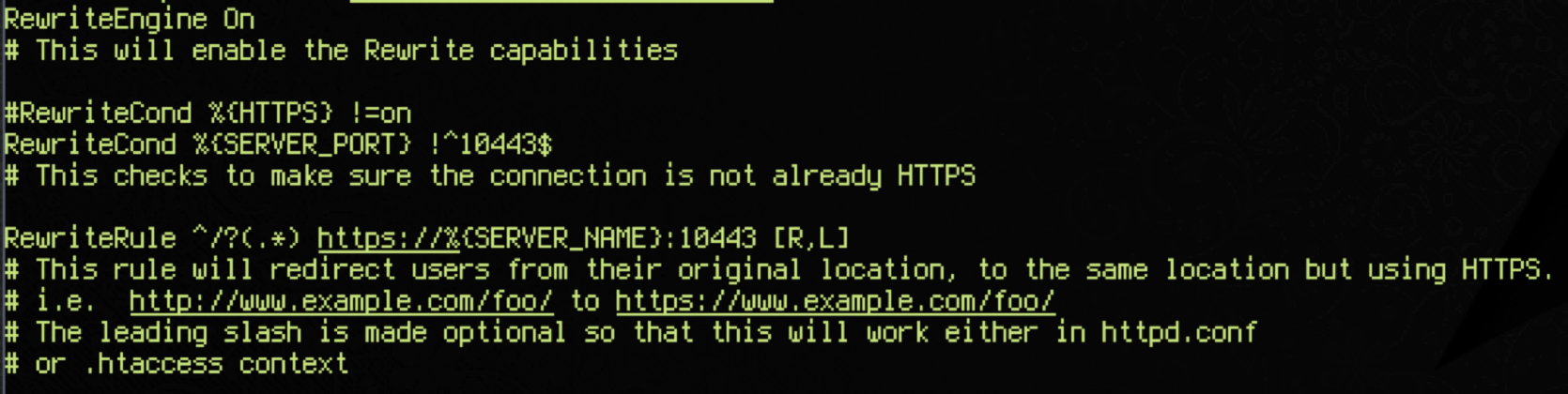
>make comp3334.key ## make a secret key

>openssl rsa -in comp3334.key -out comp3334.key ## make a RAS key

>make comp3334.csr ## make a CA cert

>openssl x509 -in comp3334.csr -out comp3334.crt -req -signkey comp3334.key -days 3650 ## finally generate the self-signed cert

>chmod 400 comp3334.\*

Then configure the apache config file to redirect to https[5]:

Technical Details

Hardware specification:

Asus EeeBox B202

hardware details:

<http://www.asus.com/EeeBox_PCs/EeeBox_PC_B202/specifications/>

OS:

Linux Fedora 21

Software details:

web server: apache 2.4.10-15

database: mariadb-server-10.0.15

ssl: openssl-1.0.1k

firewall: iptables-1.4.21-13

References:

1. NRC National Research Council (1997) ―For the Record: Protecting Electronic Health Information.

2. [https://wiki.archlinux.org/index.php/Sudo#Root\_password](https://wiki.archlinux.org/index.php/Sudo)

3. <https://wiki.archlinux.org/index.php/Simple_stateful_firewall>

4. <http://www.server-world.info/en/note?os=CentOS_6&p=ssl>

5. <https://wiki.apache.org/httpd/RewriteHTTPToHTTPS>