System Description and Risk Analysis

Markus Ding Eric Jollès Simon Perriard Hédi Sassi

...

Page limit: 30 pages.

Contents

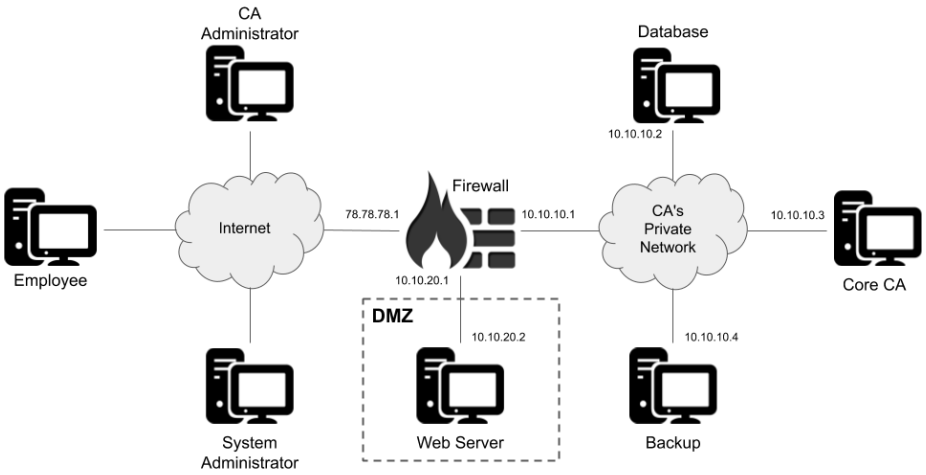


Figure 1: Components and network connection of the system

1 System Characterization

1.1 System Overview

iMovies is a company producing independent movies of various kind, but with a focus on investigative reporting. Due to the sensitive nature of such reporting, discretion is very important and must be taken into consideration when designing and setting up the system. The mission of its system is to provide a certificate authority (CA) which will manage digital certificates for secure (confidential + authenticated) email communication between employees. Employees must be able to request and/or revoke these digital certificates (DC). Each employee will have a maximum of one valid (non-revoked) certificate at a time. This limits the surface of attack since the amount of certificates is reduced.

The system's relevant technical and network components are depicted in Figure 1. This system is composed of 4 distinct parts:

- **Users** (inside or outside the company's network). They can be employees, CA's administrators or system administrators (learn more about their roles in Section 2.1.3).
- The **CA's private network**, composed of a **database** machine with users' information, a **Core CA** machine managing the creation and revocation of certificates and a **backup** machine containing copies of important files to prevent data losses and allow a quick recovery in case of disruptions.
- A **Web server** on which an employee can see his information stored in the system's database, edit his own information, request a new digital

certificate or revoke an older one. CA administrator can also consult CA's statistics.

- A **firewall** limiting connections between the internal (The CA's private network) and external network (internet). As the web server must be accessible remotely, a demilitarized zone (DMZ) containing the web server is created.

All communications are encrypted and authenticated in order to provide integrity and confidentiality of the transmitted data. In order to protect highly sensitive data, the company holds a secure vault in its basement. The vault is under constant surveillance (cameras and motion sensors) and can trigger alarms.

1.2 System Functionality

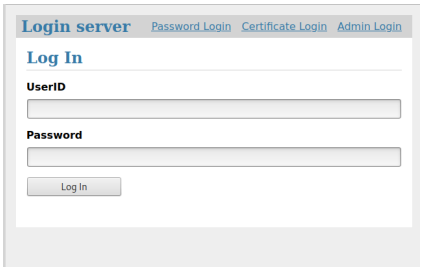
1.2.1 User Authentication

Employees (also called clients) must be authenticated to access CA functionalities. To do so, they must connect to the web server from their machine. This can be done in two different ways:

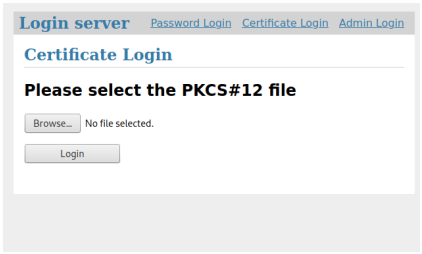
- The client can log himself via a web form (see Figure 2a) by entering his userid and his password. The username and the SHA-1 hash of the password (computed on the client side with JavaScript) are sent to the web server and compared with the one stored in the database (see Section 1.4.2).
- The client can also log in by proving that he owns a valid certificate. For this, the client has to select his PKCS#12 file (certificate + private key) (see Figure 2b). The client should never send his private key, because an attacker could decrypt his emails or impersonate him if he gains access to it. In order to avoid this, we decided to use a challenge-response scheme. The web server sends a challenge (nonce) in the web form to the client. To prove the knowledge of his private key without revealing it, the client signs this challenge and sends this signature with his certificate to the web server. The PKCS#12 file is thus never submitted to the server, it is only used to locally compute (in JavaScript) the signature and extract the certificate. Certificates must be valid (signed correctly and valid in time) and not present in the web server's certificate revocation list (see more in Section 1.2.3). If the signature is correct, the client is authenticated with the information contained in the certificate.

In order to provide one-way authentication towards the client, the Web server's certificate is signed by QuoVadis' CA. Once logged in to the web server, a user can:

- consult his information stored in the database (see Table 1). The password is not shown to the user in order not to unintentionally leak his password.



(a) Login interface with userid/password pair



(b) Login interface with certificate authentication (the CA admin login interface is the same but on another url)

Figure 2: Login interfaces

Attribute name	Attribute type	Description
uid	VARCHAR(64)	Unique user identifier
lastname	VARCHAR(64)	User family name
firstname	VARCHAR(64)	User first name
email	VARCHAR(64)	User's company email
pwd	VARCHAR(64)	SHA1-checksums of the user password

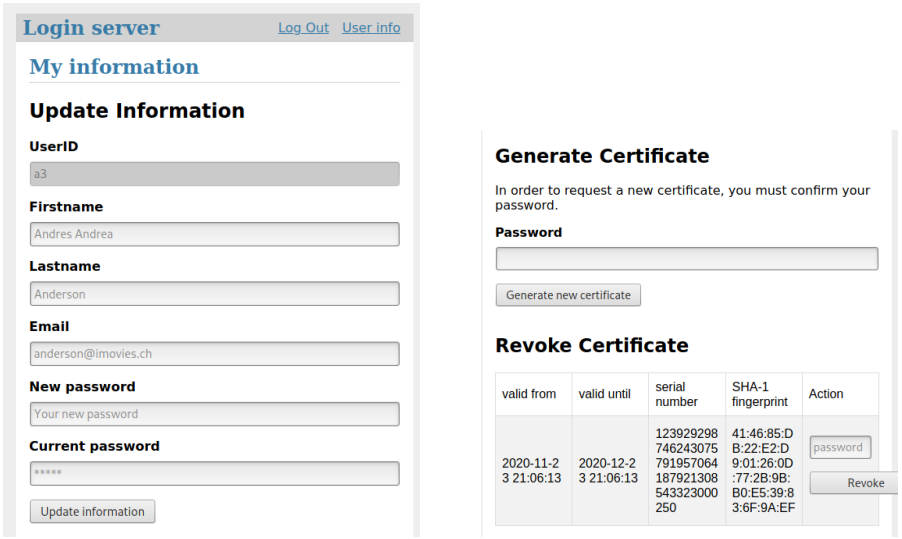
Table 1: Description of the *users* table of the MySQL database

- modify his information stored in the database (see Table 1). The user cannot change the userid field, since it is the primary key of the legacy MySQL Database. The user must enter his password if he wants to change his data to avoid modifications from an attacker with a stolen private key.
- request a new certificate (see more in Section 1.2.2).
- revoke a certificate (see more in Section 1.2.3).

We can see in Figure 3 the user interface after login.

1.2.2 Certificate Issuing Process

Once logged in to the web server as described in Section 1.2.1, a user can request a new certificate. This action must be confirmed by his password to prevent a certificate request by an attacker with a stolen private key. The request for a new certificate is sent by the user to the web server. The web server forwards the request for the new certificate, along with the user's information, to the Core CA. Once the request is received, the Core CA generates a new random RSA private key. A new certificate is created in X509 format containing information about the user (email, firstname, lastname and user id) and the public key from the new private key. The certificate is signed with the Core CA's root



(a) The first part of the user information page: the form to display and update his information

(b) The second part of the user information page: certificate issuing and revocation)

Figure 3: Login interfaces

private key. The serial number of the certificate is randomly generated. The new certificate and the corresponding private key are assembled in PKCS#12 format and returned to the web server. The user can then download it from the web server. The certificate is valid for 30 days. After that time period, the client must issue a new one. This way, an attacker that steals a PKCS#12 file doesn't have a long time access to the encrypted emails. The web server does not keep a local copy of the PKCS#12 file, thus the user cannot download this PKCS#12 at a later connection because being more restrictive with respect to the access to sensitive information diminishes the attack surface.

The new certificate and the associated private key (in PEM format) are also sent encrypted, using AES256-CTR[1] with the Core CA's backup key (see more in Section 1.2.5) to the backup server (see more in Section 1.3.4). The certificates are finally stored on the Core CA without the private keys, since we will not need them in the future on this machine (we will access the backup in case of key loss).

Since a user cannot have several valid certificates at the same time, issuing a new certificate will automatically revoke the previous one. The user must confirm this process through a pop-up alert when clicking the button. The revocation is done according to the details in Section 1.2.3.

[1]See more on: `https://en.wikipedia.org/wiki/Block_cipher_mode_of_operation#Counter_(CTR)`)

1.2.3 Certificate Revocation Process

If a private key is compromised/lost or if a user wants to change his information and issue a new certificate, he can manually revoke his previous certificate. To do this, the user must first log in to the Web server as described in Section 1.2.1. Then, he can revoke his certificate. This action must be confirmed by his password to avoid unwanted revocation. The revocation request is sent by the user to the web server. It is then sent by the Web server to the Core CA which will update its certificate revocation list (CRL) in X509 format and send it back to the web server. All future connections using a revoked certificate must no longer be able to connect to the Web server.

1.2.4 CA Administrator Interface

CA administrators can consult the CA's current state using a dedicated web interface, see Fig 4.
In this web interface, they can observe:

- the number of issued certificates,
- the number of revoked certificates,
- the current serial number (serial number of the last certificate that was issued).

CA administrators authenticate themselves with their digital certificate in the same way as explained in Section 1.2.1 but on the dedicated administrators' login page.

Figure 4: The statistics of the Certificate Authority that are only available to the CA administrators after the certificate login

1.2.5 Backup

We want to make sure that the backup data arrives at the backup machine and that the content of these communications is not visible to anyone as it can be sensitive; that is why we use TLS connections with mutual authentication (certificates required on both sides) between each machine and the backup server. Therefore, all machines have a TLS certificate to prove their identity. In the event of a machine failure, a copy of each of these certificates and the corresponding private key is kept in the secure vault.

We used a custom backup server instead of `rsyslog`[2] since we encrypt sensitive data on the fly during the backup process. In addition, `rsyslog` also has some vulnerabilities (see: course book[3] Section 5.3).

Key Backup : To ensure that encrypted data remains accessible even if an employee's certificate, private key, or even the employee himself is lost, a copy of all issued keys and certificates are stored in the backup server. The CA's root private key is also stored in the backup server. All this data is sensitive, so it cannot be saved as it is in the backup server. It is therefore encrypted on the Core CA's side before being sent to the backup server. The key used for the encryption, called **backup key** (256-bits AES key), is only known by the Core CA. To avoid data loss in the case where the Core CA loses this key, a printed copy is stored in the vault. The backup is updated each time a certificate is issued or revoked.

If an employee loses his private key, he can ask system administrators to recover it from the backup (after having checked his identity using his ID or employee card). To do so, the employee must go to the company's offices in order to recover it. We chose this design because it is better to have a copy in a relatively safe machine than to have an employee copying his private key everywhere because he's afraid of losing it. By doing so, we reduce the user-side attack surface.

Log backup : In the event of an attack on the system, we would like to be able to see the origins of the attack. To do this, we record the actions performed in the different machines with logs. After an attack, iMovies' forensic experts will be able to read these logs. Because an attacker could erase these logs on an attacked machine, they are also sent to the backup upon change. Saved logs are:

- SQL queries received by the database. Queries are not directly added to the log since they may contain private information. In the log, we keep: the date, the type of query (selection or update) and the user in question.
- any action (connection, function execution) with the web server. We keep in the log: the time, the action performed and the result of this action.

[2] See more on: `https://www.rsyslog.com/`

[3] Applied Information Security - A Hands-on Approach; by David Basin, Patrick Schaller, and Michael Schläpfer

- any action (connection, function execution) with the CA server is kept in a log file. We keep in the log the action performed and the time.
- any changes in the configuration files or in the code files for all machines in the system are tracked by the backup script. The date of modification is contained in the name of the backed-up file.
- the /var/log/wtmp, /var/log/lastlog, /var/log/syslog and /var/log/auth.log log files are saved for all machines in our system so we can trace who was active in the systems and we have access to security relevant logs.

As a rule of thumb, any investigation on the logs should be performed with the logs contained in the backup server and not the logs from the suspected machine since they may have been tempered with.

Machine backup : The scripts and configurations are stored on the backup server to enable a fast recovery in case of failure. They are sent to the backup machine once they have been modified.

Database backup : The content of the database is stored encrypted (using AES-CTR) in the backup once a day at 1 am. The symmetric key is only known to the database server and is stored in the vault.

Backup failure : To guard against a possible failure of the backup machine, backup archives are made on a daily basis. They are made by the system administrators following the 3-2-1 Backup Rule[4]: 3 copies of the data are made, 2 backup copies are on a different storage media (one in a hard drive the other two on magnetic tape), one of which is located off-site (in another iMovies office). By doing this, we don't have a single point of failure regarding the backup of the whole system.

1.2.6 System Administration and Maintenance

Remote administration can be made from the internet via the `ssh` protocol, which provides confidentiality and integrity over an unsecured network. `ssh` public keys are already installed on all machines for authentication. Login with password is disabled to limit the attack surface and prevent attacks based on password guessing. Since the public keys are different for all machines, administrators are only given the private key to access a machine based on the least privilege principle.

[4]See more on: `https://www.nakivo.com/blog/3-2-1-backup-rule-efficient-data-protection-strategy/`

1.3 Security Design

1.3.1 Access control

Processes of different machines run with the necessary privileges to perform their tasks, applying Least Privilege and Compartmentalization principles. Each machine has one root and one non-root user in order to enforce the Least Privilege principle. The servers and/or applications used in the system are running as non-root users; thus, if they get corrupted, the attacker is less likely to gain root access or to have a significant impact on the system.

By default, non-root users cannot modify the code of the servers that are running on the machines (e.g. CA_server.py on the core CA machine) and cannot temper with the keys and certificates.

- **Database** : The "database" user can only launch the database server and the backup script but cannot modify the scripts or libraries. He has write access to the database server log file (see Section 1.2.5) and read access for the certificates and keys used for and TLS.
- **Core CA** : The "coreca" user can execute the Python script that launches the CA server and can execute the backup script but cannot modify these scripts nor the python libraries they depend on. He has permission to read and write the certificate files that were issued and revoked and the files containing the statistics of the CA. It also has read permission on the certificates and keys used for the TLS connection. The "coreca" user has read and write permission on the CA's server logs (see Section 1.2.5).
- **Backup** : The "backupp" user can launch the backup server but cannot modify the python script, he can read and write backup files and has read permission for the TLS certificates and keys but cannot modify them. He has read and write permission on the backup server's log file.
- **Firewall** : The "firewall" user can launch the backup script but cannot modify it. He has read permission over the firewall configurations but cannot modify them. He also has read permission over the certificates and keys used for the TLS connection.
- **Web server** : The "webserver" user can execute the Python script that launches the server and the script that launches the backup but cannot modify these scripts nor the libraries they depend on. He has read permission for the certificates and keys used in the TLS connection. The "webserver" user also has read and write permission over the web server logs (see Section 1.2.5).

Access control for the web interface and its functionalities is ensured by the web server machine that is responsible for the authentication of the user using a password/username combination or a valid PKCS#12 file (see 1.2.1).

We need strong passwords of at least 20 characters (not containing words or sentences and containing special characters and numbers) for users in each

machine in order to make sure the access control mechanism is respected and no attacker can log in by guessing any password. A system administrator can only open an `ssh` session using an RSA private key file. `ssh` as root is not directly allowed. To gain root access, a system administrator has to use the `su -` command and then enter the root password. This limits the capabilities of an adversary that steals the private key needed for the `ssh` connection. Printed copies of these RSA public and private keys are stored in the vault (see Section 1.3.4).

1.3.2 Key management

All issued and revoked certificates (containing public keys) are stored in the Core CA and are also stored in the backup machine. This allows for rapid re-deployment of the entire system in case the Core CA is destroyed. This also allows to revoke all certificates and create a certificate revocation list in the case of a compromised Core CA. Private keys used to store encrypted data, CA's root key and private keys used for `ssh` connections are stored in the machines to which they belong and a printed copy is stored in the vault. Issued private key of users are stored encrypted in the backup but are not kept on the Core CA (see Section 1.3.4). This way, we limit the risk of losing keys and can re-deploy faster if one of the machines is destroyed by looking for the keys in the vault. TLS certificates (used in inter-machine communications) are valid for one year. They must be renewed every year. Thus, an adversary who may have stolen the certificates and keys can no longer perform exploits using these keys and certificates. In case of suspected security breach, all TLS certificates must be renewed.
A root private key (with an associated self-signed root certificate) is used to sign the TLS certificates for inter-machine communication. This root private key is kept in the vault and the root certificate is deployed on each machine so they can establish a TLS connection between them. For the TLS connection between the clients and the web server, we use a certificate signed by QuoVadis, which is renewed every year.
Private keys used to establish `ssh` connections are kept in the vault. If a system administrator needs to connect to the machine with `ssh`, he must obtain authorization from the information security officers, sign a chart and perform the required operations.

1.3.3 Session management

User sessions are created using HTTPS with TLS 1.3. The web server uses cookies to manage the user session, the cookies are signed so that they cannot be tampered with[5]. Cached challenge-response elements are also deleted to prevent replay attacks. On each page reload, a new challenge is issued.

Inter-machine communications are handled over TLS 1.3. System administrators use `ssh` with private key authentication to log in and maintain the

[5]See for more details: `https://flask.palletsprojects.com/en/1.1.x/tutorial/views/`

system.

1.3.4 Security of data at rest

- **Database** : We use authentication in the MySQL database, as well as encryption of data at rest using InnoDB[6]. This protects against an attacker breaking into the building and stealing the disks. The content of the database is also stored encrypted using AES256-CTR in the backup. This method allows the database to ensure integrity and confidentiality with respect to other system components and also provides integrity after key leakage. A printed version of the encryption key is stored in the vault. The MySQL database is not directly reachable from outside the machine. Only the database server (see Section 1.4.2) has access, with a limited set of queries, to the content of the MySQL database.

- **Core CA** : Must generate keys that are hard to break. Certificates and private keys must be stored encrypted using AES256-CTR in the backup as described in the previous point. A printed version of the encryption key is also stored in the vault.

- **Backup** : Sensitive data are encrypted before backup see 1.2.5.

- **Vault** : In order to protect highly sensitive data, the company holds a vault in its basement. The vault is under constant surveillance (cameras and motion sensors) and can trigger alarms. There are 3 different keys for the vault: one for the founder of the company, one for the head of the CA's administration team and one for the head of the incident response team. Two of the three keys are necessary to open the vault. This prevents a corrupted employee from stealing the data stored in the vault and complies with the "no single point of failure" principle. Each time the vault is opened, it automatically sends to all CA administrators and the information security officers a notification, using the company's internal mail.

1.3.5 Security of data in transit

All communications used between components are encrypted (either with TLS1.3 or `ssh`) or denied by the firewall. Since TLS1.3 and `ssh` are always configured using certificates, in the CA's private network (`ssh` using password is disabled), we have confidentiality, integrity and authentication for all data exchanges. Between the clients and the web server, we cannot guarantee integrity and confidentiality because the client is not authenticated to the web server, while the web server is authenticated to the client. Note that in case of a successful connection, confidentiality, integrity and authentication are ensured between the web server and the client.

[6]More details on: `https://dev.mysql.com/doc/refman/5.6/en/innodb-introduction.html`

Machine	Open port	Protocol	Use
Web Server	5000	TLS	Access to the website (HTTPS)
Web Server	22	`ssh`	Maintenance by System Administrator
Core CA	6000	TLS	Access to Core CA's functionalities
Core CA	22	`ssh`	Maintenance by System Administrator
Database	42069	TLS	Access to the database's functionalities
Database	22	`ssh`	Maintenance by System Administrator
Firewall	22	`ssh`	Maintenance by System Administrator
Backup	5555	TLS	Receive backup data
Backup	22	`ssh`	Maintenance by System Administrator

Table 2: Open ports in the system

1.4 Components

1.4.1 Platforms

On all of the following machines, system administrators should check for and install updates regularly (once a day). The system components are distributed on different machines that have specialized functions. By separating the tasks on different machines, we respect the principles of compartmentalization.

- **Firewall** : This machine is a Debian machine that handles the protection and routing of traffic between internal (The CA's private network) and external network (internet). It is maintained by system administrators via `ssh`. This firewall is configured to deny by default inbound connections to the internal network from the outside, except for system administrators, who can access it for maintenance purposes. This follows the "failsafe default" security principle. It denies all connections using invalid ports (See Table 2). For this purpose, the firewall has one DMZ connected to it (see Figure 1) that allows clients and CA administrators to visit the Web server while protecting the CA's private network. The firewall is managed by external system administrators, who can access it via `ssh` on port 22.

- **Core CA** : This machine is a Debian server that listen for incoming demands concerning certificates on port 6000 and is maintained by System administrators via `ssh` on port 22. This machine runs the CA script which can issue new certificates, revoke old ones and update the CRL. This machine also handles functions to verify a login and to retrieve and modify user's information.

- **Database** : This machine is a Debian server running a MySQL database containing user data and allows the Web server to access its data through an API (see more in Section 1.4.2). The MySQL database contains one table users (see Table 1) with the following fields `uid`, `lastname`, `firstname`,

`email`, `pwd`. The machine can only be accessed by the web server (see Section 1.4.2) via TLS on port 42069 and by system administrators via `ssh` on port 22.

- **Web server** : This machine is a Debian machine which handles client requests with a Flask server (on port 5000), it is maintained by system administrators via `ssh` on port 22. It only handles HTTPS packets. Using HTTP will result in a reset connection. Via TLS connection on port 6000 it can make certificate requests, certificate delivery and revocation requests for employees. It also has a CRL database, which is updated each time there is a new revocation request. When CA administrators are logged in, the web server can request CA's statistics (see Section 1.2.4).

- **Backup** : This machine is a Debian system that holds the backups of the other components of the system, runs the backup server (a Python script) on port 5555 and is maintained by system administrators via `ssh` on port 22. It saves contents, parameters, logs and states of the firewall, the Core CA, the database and the Web server (see more in section 1.2.5). For this purpose, on each machine, Python scripts returning archived data are triggered by each change in the folders containing the data that must be stored. The scripts are deployed on each system requiring a backup (see more in Section 1.2.5).

1.4.2 Applications

All code is open source in order to guarantee the open design principle.

- **Firewall** :

  Routing: the routes are configured in `/etc/network/interfaces` and we enabled IPv4 forwarding on the kernel.

  Firewall: we used `iptables` to enforce connections of Figure 5.

  The firewall is also protected from basic DOS attacks (e.g. SSH brute force protection) using `iptables` rules. The `iptables` configuration is done with a bash script containing all the rules. The default behavior for incoming packets is DROP.

- **Web server** : The web server has a Flask web server which is launched by a Python script that is always running to remain available at all times. The web interface is compatible with all modern browsers and provides the following functionalities (see more in Section 1.2):

  - User login with password: `https://10.10.20.2:5000/auth/login`
  - User login with certificate: `https://10.10.20.2:5000/auth/cert`
  - CA administrator login (with certificate): `https://10.10.20.2:5000/auth/admin`

  - Checking and updating employee info (if employee logged in): `https://10.10.20.2:5000/auth/user`
  - New Certificate issuing (if employee logged in): `https://10.10.20.2:5000/auth/user`
  - Certificate revocation (if employee logged in): `https://10.10.20.2:5000/auth/user`
  - See CA statistics (if CA administrator logged in): `https://10.10.20.2:5000/auth/stats`
  - Logout (clear the session): `https://10.10.20.2:5000/auth/logout`

  The web server communicates with the database and the core CA through predefined APIs to reduce its capabilities and thus reduce the attack surface of an attacker who would have taken control of the web server.

- **Core CA scripts**: In the Core CA machine, a Python script is always running. It communicates with the Web server. It can create new certificates, revoke old ones, verify the validity of a certificate and update the CRL (see more in Section 1.2.2 & 1.2.3).

- **Backup scripts**: A Python script constantly monitors each target folder and triggers a backup upon change. Machines push changes to the backup server. If the target file or folder is a log file, the backup server will append the changes. Otherwise, a version of the changed file (with the backup time in the name) will be stored in the backup server. Sensitive backups (private keys and database dumps) are encrypted in the backup script before being send to the backup server.

- **System administrator interface**: System Administrators can access all machines through `ssh` connections.

- **MySQL server** : This database contains user information (see Table 1).

- **Database server** : The database has a Python server which is launched upon startup that is always running to remain available at all times. We don't want to let the web server make SQL queries directly to the MySQL server because if the web server falls into the hands of an attacker, the whole database could be recovered or modified. For this, the database server is the only process that can access the MySQL Database. When the web server needs to communicate with the database (check password, get users' information), it sends commands to the database server. Here are the possible commands:

  - **check_password** : which checks if the username/password pair is valid. It checks the hashes locally and returns `true` or `false`, so the database's password hashes are never sent to the web server.
  - **get_user_data** : which sends the employee information back to the web server (see more in Section 1.2.1).

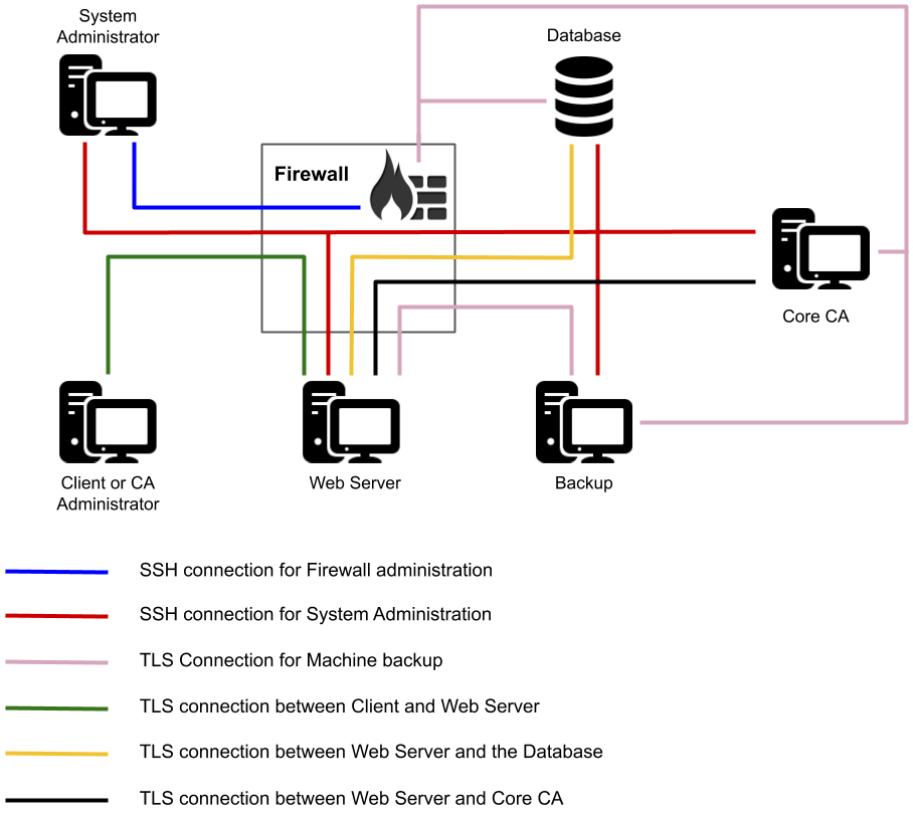


Figure 5: Connections allowed by the firewall

– **update_user_data** : which updates employee information.

All SQL queries and updates we used in this project are enforced with prepared statements in order to prevent SQL Injection attacks.

1.4.3 Data Records

- **User information** : User information stored in the database server (see Table 1).
- **Configuration files** : Configuration files of all machines which can be used to configure a new machine if a problem occurs.
- **Certificates and Private keys** : Certificates signed by the Core CA and their corresponding keys
- **Logs**: Logs of applications and machines.

1.5 Backdoors

1.5.1 Trivial Backdoor

The browser's user agent string allows servers to identify client information such as browser version and operating system. It is sent to the website in the "user agent" field of its HTTP header. This field can be modified by the client with Burp Suite or with user agent addons available on Firefox or Google Chrome. Our backdoor relies on the content of the user agent. Indeed, if the web server receives a user agent string corresponding to a mobile phone (i.e. containing the words "Android", "iPhone" or "Phone"), it automatically grants a valid CA administrator session and the CA administrator page is accessible without need to log in with a certificate. This breaks the security property that the specific page is only available to CA administrators after a successful login process. Most modern browsers also have tools to test the responsive design of websites and therefore allow to mimic a mobile phone (resolution, user agent, etc...) without having to install additional programs, which makes this backdoor attack simple to perform.

1.5.2 Non-trivial Backdoor

For this backdoor, we created a program with the same name and almost the same code as the Core CA server but it doesn't use the TLS connection (it creates the secure socket but never uses it) and instead of sending the CRL back upon revocation, it leaks the CA's root key. The code is hidden in the middle of other python libraries (in `/lib/python3/dist-packages/CA_server.py`) to make it difficult to find.

Since its behavior and the port it uses is the same as the Core CA server (we made the original server use port 6000 by default and added the 6001 if the other is taken and the "fake" server uses port 6001 as default and 6000 if the other is in use), it is difficult to notice that a different server is also running (one may think that it is a bug where two instances of the server are running). The port it mainly listens to (6001) is hidden using `iptables` with port knocking[7] on the firewall. Thus, the port 6001, that the "fake" server is using, appears to be closed and will be available only after the right sequence of port has been "knocked" (the right sequence is : 3306 (tcp), 8000 (tcp), 6000 (tcp), 6001 (tcp)).

To make things even harder, the "fake" server will be launched once a day during one hour using `cron`, between 1 am and 2 am. We tried to hide `cron` processes on the Core CA machine as much as possible. As a result, the output of `crontab -e` and `crontab -l` commands is obfuscated thanks to code that simulates their output in `.bashrc`[8].

This makes the backdoor very difficult to find in blackbox testing. It is possible to find the backdoor during whitebox testing if the testers are trying to

[7] See more on: `https://medium.com/secjuice/how-to-hide-your-ports-with-port-knocking-cb7f244849e7`

[8] as suggested in this post: `https://unix.stackexchange.com/questions/556149/hide-a-crontab-job`

find out which program is using the open ports of the system during the time the backdoor server is running. It is also possible to notice that the output of the command that displays the content of the crontable have been tempered with in ".bashrc".

Hide this subsection in the version handed over to the reviewing team by setting the flag `showbackdoors` at the top of this document to `false`.

2 Risk Analysis and Security Measures

2.1 Assets

We have divided the Assets in 4 different categories as is recommended by best practise (ASL book).

2.1.1 Physical Assets

For each machine, all administration is done online. To prevent non-network related attacks, all input/output components (CD/DVD-ROM, USB reader) must be disabled, except for the backup server which must be able to create archives on different media (see Section 1.2.5). The components (fan, processor, graphic card, etc.) must be in normal operations mode, otherwise the machines could be damaged, be less efficient or even stop working, threatening the availability of our system. This can be checked daily by 2 randomly selected employees (to avoid the single point of failure).

- **Core CA / Database server** : The Core CA machine and the database server are located on the engineering floor of iMovies in a locked room. They are on the engineering floor to allow employees to observe if someone enters it. They are located in two separate lockable racks to create a physical isolation between machines.
- **Firewall / Web server** : The firewall and the Web server are located on the same floor, in another locked room to increase physical isolation. For the same reason, they are located in two separate lockable racks.
- **Backup** : The backup machine is located in another locked room (also for physical isolation), on the same floor.
- **CA's private Network** : Internal networks are Ethernet local area networks (LANs). Ethernet sub-networks are distributed using layer 2 switches. The firewall routes traffic between the networks. The proper functioning of internal networks is essential for iMovies productivity. It provides network connectivity to the Internet and is necessary for the operation of the web server.

- **Internet Connectivity**: The router that connects to the service provider's network is placed in the same rack as the firewall. Internet connectivity is essential for iMovies, as it is the primary means of communication for iMovies employees to obtain certificates.

2.1.2 Logical Assets

Software

- **Firewall** : Debian 10.6.0 amd64, `iptables` for firewall & `ssh`.
- **Web Site** : Debian 10.6.0 amd64, `ssh`, Python3 & Flask.
- **Core CA** : Debian 10.6.0 amd64, `ssh` & Python3.
- **Database** : Debian 10.6.0 amd64, MySQL Community Server 8.0.21 with InnoDB engine for data encryption at rest, `ssh` & Python3.
- **Backup** : Debian 10.6.0 amd64 & `ssh` & Python3.

Information

- **Logs** : Log files of machines and applications. They can be used to detect suspicious activities or to reconstitute an attack.
- **Configuration** (needed to rebuild a machine after critical failure or destruction of the said machine):

  Network: network configuration in each machine `/etc/network/interfaces`

  Routing: firewall:`/etc/network/interfaces`

  Firewall: firewall:`/root/firewall_ruleset.sh`

  Database: network and encryption database config :`/root/imovies_users.sql` and `/root/mysql_server_conf.sh`
- **Web server's TLS private key** : Private key of the web site used to authenticate to the the client.
- **TLS private keys used for inter-machine communications** : Private keys used between the machines to establish TLS channels used in the backup script, CA server, web server and database server.
- **CA's root private key** : Key used to create employee certificates.
- **Self signed TLS root private key** : Key used to sign certificates for TLS communication between the machines
- **Certificate revocation list (CRL)** : It contains all the revoked certificates. It is updated at the Core CA and at the Web server each time a revocation is made.

- **User data** : Contains user information (see Table 1). Most important information is user id and password since it is used to connect to the Web server and make new certificates.
- **User's certificates and private keys** : User certificates with their corresponding private keys that can be used to secure emails or to login to the web server.
- **ssh keys** : `ssh` private keys allowing administrators to log in to the company's system.
- **Backup encryption keys** : Symmetric key used by the CA and the database machines to store sensitive data on the backup machine. They are stored on both machines and in the vault.

Connectivity

- **Internet connection** : The system is connected to the internet by optical fiber with a good bandwidth (1 Gbit/s).

2.1.3 Persons

- **Employee** : iMovies employees represent the users of the authenticated email server. They can issue or revoke certificates and also view or modify their information.
- **CA Administrators** : Through a dedicated web interface, they can view the current status of the CA (Number of certificates issued, Number of certificates revoked and current serial number).
- **System Administrators** : iMovies has system administrators who maintain the machines. Every system administrator has access to all critical system data.
- **Information Security Officers** : Responsible for the protection of the logical and physical assets of the company. They supervise the work of system and CA administrators and control access to sensitive material (private keys, vault, server rooms).
- **iMovies Managers** : Have the highest access level amongst all employees. Receive reports from the information security officers and take actions to ensure the safety of the system.

2.1.4 Intangible Goods

- **Employee confidence** : Employee confidence is a necessary prerequisite for a good working atmosphere.
- **Company's reputation** : Without its clients, the company is nothing. The company must therefore avoid any story that could tarnish its reputation in order to maintain the trust of its clients.

2.2 Threat Sources

- **Nature** : Depends on where our company is located. Possible problems: Water, fire, volcano, meteorite, tsunami, avalanche, landslide, earthquake. In addition, lightning and pollution should also be taken into account.
- **Employees** : Regular employees, CA or system administrators can be a possible threat. In addition to the technical staff mentioned so far, one should not forget the concierge and the cleaning staff who probably have physical access to all relevant equipment. One of their main motivations may be the money they can make from selling information.
- **Ex-Employees** : Employees who previously worked for the company, and who have been laid off or left. They can remember the company's infrastructure, credentials and secret files. They may have a desire for revenge, or they may be interested in money.
- **Script Kiddies** : As the company's system is connected to the Internet, it is exposed to attacks from script kiddies. They are interested in easy money.
- **Skilled Hacker** : Although skilled hackers are not the main source of threat to the iMovies company, they may be hired by iMovies' competitors who want to decrypt emails exchanged between iMovies employees.
- **Malware** : Malware (both directed and undirected) must be taken into account since they are widely used to extract information or compromise systems.
- **Organized crime, governmental agencies and terrorists** : iMovies produces independent movies focused on investigative reporting. Thus, depending on the subject, organized crime, government agencies or terrorists may be interested in knowing more about the emails exchanged and the identity of certain sources. They may want to modify or delete valuable information about their internal workings that iMovies could have discovered.

In the following risk evaluation, unless otherwise specified, we will use the term "attacker" to encompass the different skill levels of adversaries such as script kiddies, a skilled hacker or an organization with a greater reach and funds such as a governmental agency. Their chances of success vary, but if the countermeasures are the same, we have listed them only once under the term "attacker" and considered them to be the highest level of risk for the security analysis.

2.3 Risks Definitions

Define likelihood, impact and risk level using the following three tables.

Likelihood	
Likelihood	Description
High	The threat source is highly motivated and sufficiently capable of exploiting a given vulnerability in order to change the asset's state. The controls to prevent the vulnerability from being exploited are ineffective.
Medium	The threat source is motivated and capable of exploiting a given vulnerability in order to change the asset's state, but controls are in place that may impede a successful exploit of the vulnerability.
Low	The threat source lacks motivation or capabilities to exploit a given vulnerability in order to change the asset's state. Another possibility that results in a low likelihood is the case where controls are in place that prevent (or at least significantly impede) the vulnerability from being exercised.

Impact	
Impact	Description
High	The event (1) may result in a highly costly loss of major tangible assets or resources; (2) may significantly violate, harm, or impede an organization's mission, reputation, or interest; or (3) may result in human death or serious injury.
Medium	The event (1) may result in a costly loss of tangible assets or resources; (2) may violate, harm, or impede an organization's mission, reputation, or interest, or (3) may result in human injury.
Low	The event (1) may result in a loss of some tangible assets or resources or (2) may noticeably affect an organization's mission, reputation, or interest.

Risk Level			
Likelihood	**Impact**		
	Low	Medium	High
High	Low	Medium	High
Medium	Low	Medium	Medium
Low	Low	Low	Low

2.4 Risk Evaluation

2.4.1 Evaluation of Physical Assets

Hardware

No.	Threat	Countermeasure(s)	L	I	Risk
1	Nature: Water, fire, volcano, meteorite, Tsunami, avalanche, landslide, earth-quake, ...	One backup archive is saved offline in another place (encrypted). The location of the company should be low risk and the building must resist as many of these attacks as possible.	*Medium*	*High*	*Medium*
2	Natural decay: Components might break making the machine unusable (e.g., hard-disk defect).	Each machine has no special requirements and thus could be replaced. Parameters and data can be recovered with the backup system.	*Medium*	*Low*	*Low*
3	The physical access to the machine by insider (employees, administrators) or competitors.	There is physical protection of machines, well-defined room access (at least 2 people chosen at random among authorized personnel), access with badges and fingerprints, physical access archives stored in another secure room and alarm system.	*Low*	*High*	*Low*

CA's private Network Physical

No.	Threat	Countermeasure(s)	L	I	Risk
4	Cable cut by a disgruntled person or organization that has been or will be the subject of an iMovies documentary.	Cables are protected (buried in concrete), their state is verified once a week.	*Low*	*High*	*Low*
5	Wiretapping by a governmental agency.	Cables are protected (buried in concrete) and all communications are encrypted.	*Medium*	*Low*	*Low*

Information assets

No.	Threat	Countermeasure(s)	L	I	Risk
6	Logs modification or erasure voluntary (attacker) or involuntary (administrator error or machine failure).	Logs are send to the backup machine to have a second copy and to be able to confirm the integrity of the original one.	*Medium*	*Low*	*Low*
7	System misconfiguration by system administrators.	Have external audits every month to verify that there are no issues. Each modification goes through a modification review process and is double checked by the CA's administrator team.	*High*	*High*	*High*
8	Theft of certificate's private key or user password by an attacker.	Revoke the certificate as soon as a malicious usage of the certificate is detected. We must also raise awareness of the users to avoid these situations happening through internal training. The validity period of the private key/certificate is 30 days.	*Medium*	*High*	*Medium*
9	Theft of system administrator `ssh` key by an attacker.	System administrators have to follow a security seminar on proper key management and best practices.	*Medium*	*High*	*Medium*
10	Faulty backup.	Backup archives are made following the 3-2-1 rule (see Section 1.2.5).	*Medium*	*Medium*	*Medium*

Internet Connectivity

No.	Threat	Countermeasure(s)	L	I	Risk
11	Connection interrupted (by ISP, Denial of service, government, ...). This is committed by source threats that want to undermine the availability of our system.	A mirror of the whole system (including all security measures mentioned in this document) is running in Iceland using a different ISP.	*High*	*High*	*High*

2.4.2 Evaluation Logical Assets

Software

No.	Threat	Countermeasure(s)	L	I	Risk
12	Script kiddie takes control over an iMovies machine, potentially uses it as a relay host or file server, modifies relevant software.	Machines are properly maintained and not directly accessible from the Internet (firewall).	*Low*	*Medium*	*Low*
13	Skilled hacker takes control over one of the machines because of a software vulnerability, either in the operating systems or an application. Attacks can be multiple: installation of a root-kit, loss of confidential data...	System administrators are trained to notice irregularities on the machines. Machines are hardened, regularly updated are not directly accessible from the Internet.	*Medium*	*High*	*Medium*
14	Malware: Virus/worm spreads over the Internet/email possibly affects system files, possible restricted usability of machine, loss of data.	Proper maintenance of the machines, security patches installed on machines as well as antivirus software on email-server (different products), backup system, firewall shields internal network and machines from the Internet.	*Low*	*Medium*	*Low*

In case one system is corrupted, the information security response team disconnect the concerned machines. They perform forensic analysis on the corrupted systems or the backup to try to discover who the attackers are and how they gained access to the systems. Once the vulnerability has been identified and patched, the backups are used to re-deploy the corrupted machines.

Firewall

No.	Threat	Countermeasure(s)	L	I	Risk
15	Firewall misconfiguration by administrators. This misconfiguration can be used by attackers to gain illegal access to some machines.	Each change must be double checked by the CA's administrator team.	*High*	*High*	*High*

Backup

No.	Threat	Countermeasure(s)	L	I	Risk
16	Attacker steals the backup archive.	The sensitive archives (private keys and database dumps) are encrypted with a 256-bits symmetric key.	*Low*	*Medium*	*Low*

Database

No.	Threat	Countermeasure(s)	L	I	Risk
17	Attacker steals the database.	The database is encrypted using InnoDB.	*Low*	*Medium*	*Low*

Web server

No.	Threat	Countermeasure(s)	L	I	Risk
18	Attacker gains control on the Web server and issues requests to the CA.	System hardening performed on Web server to prevent intrusion. Secret keys are never stored on the web server. The client never sends his private key to the web server. If the intruder modifies the web server code or configurations, the backup server will receive the modifications and system administrators will notice the change. Upon connection, the attacker will trigger the backup of the `lastlog` or `auth.log` files and the access will be noticed.	*Medium*	*High*	*Medium*

Core CA

No.	Threat	Countermeasure(s)	L	I	Risk
19	An attacker gains full control on the CA system and forge certificates using the CA's root key and revoke certificates. This renders spoofing attacks possible and information exchanged in emails could be at risk.	The system administrators retrieve the copy of the CA's root key from the vault and revoke all certificates using the backups and publish the certificate revocation list.	*Low*	*High*	*Low*

2.4.3 Evaluation Person Asset

No.	Threat	Countermeasure(s)	L	I	Risk
20	An employee unintentionally misconfigures software such that one machine is unusable, possibly resulting in loss of data. This impacts the availability of the system.	Well-trained system administrators, restricted rights for users, backup system, users are regularly sent to training, experienced users.	*Low*	*Medium*	*Low*
21	System administrators install faulty software or do not update one of them. This could lead to unpatched or new vulnerabilities that can be used by an attacker to gain control over a machine or impact the availability of the system.	Double check what an administrator does, enforce an update policy as well as a review process before any installation or change in configuration.	*High*	*High*	*High*
22	Skilled hacker steals `ssh` key of an administrator (CA or System). He can after access sensitive assets of our systems and thus threaten the whole system. Confidentiality of emails is then not ensured anymore.	The disks of system administrators' computers are encrypted. It uses secure boot with a TPM chip and runs once a month a complete malware detection protocol	*Medium*	*High*	*Medium*
23	Bribery, corruption, giving confidential data to competitors that would harm the company's reputation or harm the confidentiality of the email exchanged.	Contractual commitment to obey non-disclosure policies for all employees even after they are terminated.	*Medium*	*High*	*Medium*

2.4.4 Evaluation Intangible Goods Asset

No.	Threat	Countermeasure(s)	L	I	Risk
24	Publication or theft of data by an attacker.	Encrypt all the sensitive data (keys, certificates and user's information) and never transmit it in the clear.	*Low*	*High*	*Low*

2.4.5 Risk Acceptance

List all medium and high risks, according to the evaluation above. For each risk, propose additional countermeasures that could be implemented to further reduce the risks.

No. of threat	Proposed additional countermeasure including expected impact
1	Contract insurance regarding natural disasters
7	Simulate VMs with new system configurations and run pentests.
8	Raise employee awareness about the protection of their sensitive data
9	Bug bounties between system administrators to retrieve ssh key from other administrators.
10	Another backup server on another subnetwork can be used. Backup machines uses RAID 6 to store the data
11	An additional web server can be added and load balancing can be implemented to reduce the impact of a DDOS attack
13	Log backups are screened once a week to detect intrusions. Information security officers receive a notification when someone logs into one of the machines
15	Schedule audit of the whole system once a month
18	When someone connects to one of the machines, the information security officers receive a notification and check that the access to the machine was authorized
21	Offer security seminars and formations for administrators
22	A secure enclave protected by a password could be used to protect the `ssh` private key. System administrators are also sent once a year for additional training on information security and malware prevention.
23	Give a bonus to employees that have a major role regarding the security