System Description and Risk Analysis

Constantin Gougas Christian Knabenhans Jiacheng Ma
Selma Steinhoff

July 10, 2020

Contents

1 System Characterization

1.1 System Overview

The system's mission is to provide trusted certificates to employees. To achieve this, iMovies provides a simple certificate authority that supplies authenticated users with a digital certificate. Users can authenticate to a web client where they can manage their personal and contact information, generate new certificate and corresponding private key, download their existing certificate, and revoke their certificate. A special user, the CA admin, can authenticate to a separate web client, and has access to statistics about the system, such as the number of issued certificates, the number of revoked certificates, and the current serial number.
The system consists of two machines - the main machine hosting a web server and the CA system, and the second one acting as backup for data. These two physical machines sit in the same server room, and communicate through a cable. The web server has access to the core certificate authority and to a database which stores user information (user id, first and last name, email address, and password hash). The backup machine stores copies of logs, databases, and CA information. Only the first machine is connected to the Internet, the backup machine is only connected to the first machine.
A visual summary of the system is shown in Fig. 1.

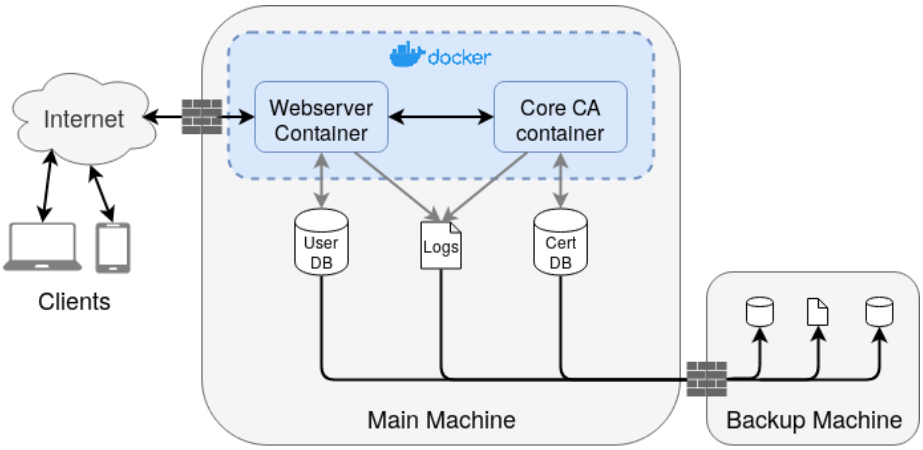

Figure 1: Components and network connection of the system. Black arrows represent encrypted channels, grey arrows represent insecure channels.

1.2 System Functionality

In the following section, we will discuss the systems functionalities. It is important to note that all described functionalities accessible to users outside the

company's network are.

1.2.1 Certificate Issuing Process

The user must first log in by entering his password and user ID in a web form or by providing his valid certificate. After a successful login, all information about the user which are stored in the user database will be displayed and can be altered by the user. We do not allow users to have more than one valid certificate.
If a user changes his private information, their previous certificate, if it exists, is revoked.
When a user wants to generate a new certificate, we check that he does not already have a valid certificate. If he does, he is given the option to download a PKCS#12 bundle[1] with his valid certificate and corresponding private key, encrypted with the passphrase he provided on generation. The user is also given the option to revoke his certificate. If the user does not have a valid certificate, he is prompted for a passphrase. A public-private key pair is generated, the user information (email address, first and last name) is tied to the public key in a certificate, and signed by the Client CA. A bundle in PKCS#12 format of this certificate and private key, encrypted by the supplied passphrase, can be downloaded as a file by the user, immediately after generation.

1.2.2 Certificate Revocation Process

After users change their personal data (name, email), the old certificate is revoked, as the data it contains is no longer valid. Users can also revoke their certificate after they authenticate on the user interface. When a certificate is revoked, the revocation bit is set in the database, a new Certificate Revocation List (which is signed by the Client CA) is generated and published immediately. Revoked certificates are not accepted by the client interface to authenticate a user.

1.2.3 CA Administrator Interface

The CA administrators can log in to the CA Dashboard through a dedicated web interface by providing their valid certificate. This interface requires the client to supply a valid certificate issued by the Admin CA. The dashboard contains information about the current serial number and the state of the CA (number of valid and revoked certificates).

1.2.4 Key Backup

Since the accessibility of the data from informants is crucial to the company's mission, a copy of all private keys and certificates is backed-up once per hour and securely stored on the backup machine. To ensure the confidentiality of

[1] `https://en.wikipedia.org/wiki/PKCS_12`

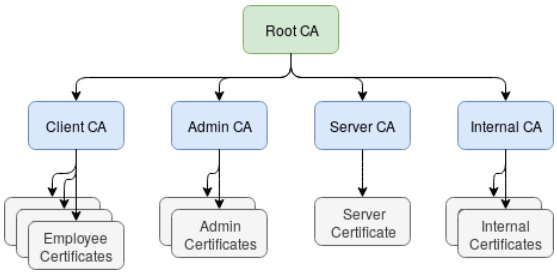

Figure 2: Certificate Authority Architecture

these information, the certificate and corresponding private key are stored in the PKCS#12 format encrypted using a user-chosen passphrase and the passphrase is encrypted using the public encryption key dedicated for passphrase encryption. The corresponding private key is stored offline.

1.2.5 System Administration and Maintenance

The system administrator has remote access to the the Web server and core CA using `ssh`. From this machine, he can also access the backup machine, also through `ssh`. Additionally, there are automated solutions in place to backup the log files and the user and certificate database and ease the life of the system administrator. All log messages are remotely logged on the backup machine over a secure channel. The backups of the databases are regularly (once per hour) created, locally encrypted using a public key dedicated for backup encryption and securely copied onto the backup machine. The corresponding private key is securely stored offline.
The CA certificates, private keys, and configurations are compressed, encrypted with a one-time password (and salted). The one-time password is encrypted with the CA backup public key, and is sent to the backup machine along with the encrypted CA files. The CA backup private key is stored offline, and can be used to restore the CA from the encrypted backup.
Similarly to the CA backup mechanism described above, the web server configuration files are also encrypted and backed up on the backup machine.

1.3 Security Design

In the following section, we will discuss the system's security design with a focus on access control, key and session management, and security of data at rest and in transit. Additionally, we will discuss protections against common web application vulnerabilities.

1.3.1 Access Control

We apply a white-list approach in the design of our system which follows the security principle of secure, fail-safe defaults. Access to system resources is de-

nied (on default) unless explicitly permitted.
Users have to authenticate themselves first, before accessing the user information dashboard and generating or revoking their certificates. This can be done by presenting the user ID and password which the system verifies against the information stored in the user database. Alternatively, a certificate-based client authentication over SSL/TLS can be used. The CA administrator has to authenticate himself before accessing the CA information dashboard by providing a valid certificate. Password authentication is not offered in this case. The system administrator must authenticate himself using his username and password before gaining access. This is independent of whether he wants to locally or remotely access the system. The backup machine can be securely connected to via `ssh` however, access is restricted to the system administrator.
Access to files is restricted using the file system permissions in Linux. The permissions are set to the minimal needed permissions for the system to successfully perform its jobs. An example of this rule is the restriction of write access to configuration files to root users only. A regular user should not be able to reconfigure a service. This should be restricted to the system administrator who has root access.

1.3.2 Key Management

Private keys are protected by passphrases, which only authorized process can access. The private key of the server (used by Apache for TLS) is the only key not protected by a passphrase. The access to this private key is constrained by file permissions. We use OpenSSL[2] to generate key pairs, certificates and certificate revocation lists. Since the CA private key and CA key passphrase have to be supplied to the OpenSSL command, we need to protect against information leakage from the command line. We encrypt the CA private key with the CA key passphrase which is stored in a configuration file with careful access control. Whenever possible, the private keys are stored offline and encrypted using a dual purpose algorithms that supports encryption and Message Code Authentication (MAC) to guarantee confidentiality and integrity.
We use different intermediate CAs each for a different purposes. This allows us to securely store the private key of the root CA offline and, in case of an compromised private key of an intermediate CA, use it to revoke the intermediate CA certificate. Only parts of the system will be affected by this and not the entire system. Figure 2 gives an overview of the chain-of-trust in our system.

The user can download its certificate and private key encrypted with a passphrase of his choice. He is responsible for distributing his certificate and private key on his devices. Users can retrieve the certificates of other users through a centralized repository. This was however not implemented, as it lies outside of the scope of the project.

- **User Private Key** After offering a secure download of the user certificate

[2] `https://www.openssl.org/`

and corresponding private key in the PKCS#12 format over `https` it is the users responsibility to store his private key securely.

- **Sub-CAs Private Key** The certificates and corresponding private keys of the intermediate CAs are generated during system setup. The private keys are encrypted using a passphrase which is stored using minimal file access permissions. The private key of the Client CA is needed to sign new user certificates and the revocation list, and is thus present in the CA Core container. Other CA private keys are not needed in the day-to-day business, and are thus stored on the host machine, but never copied in the running containers.

- **Root CAs Private Key** The root CA private key is used for signing and revocation of the Sub-CAs certificates. The private key is encrypted using a dual purpose algorithms that supports encryption and Message Code Authentication (MAC) and stored offline in a vault and in the backup machine. The private key needed to decrypt the backed up keys is stored offline, and can only be accessed with the approval of management.

- **Backup Encryption Key Pair** An asymmetric key pair that is used for encryption of the database and config file backups. The public key is stored on the main machine and used during the encryption process. The corresponding private key is stored offline and only used to recover backups in case of emergency (e.g. data loss on the main machine after breach or power outage)

- **Passphrase Encryption Key Pair** An asymmetric key pair is used to further protect the user-chosen passphrase with which the private key of the user is encrypted. The private key of the user should remain unknown to the company unless the user forgets his private key and passphrase or the user dies. In these cases the passphrase, and with that the private key of the user, should be recoverable such that the encrypted data from the informants is not lost. Following a management decision, the system administrator gets access to the private encryption key which is stored offline and can now recover the needed private key.

- **SSH Key** The SSH keys is used to provide a secure connection between the main and backup machines. This is an artifact of the virtual setup, as the connection would be provided by a physical cable in the real world.

1.3.3 Session Management

Sessions are created when the user logs in. The user ID is stored in a cookie and is used to identify the session. After the user logs out or after a time-out, the session is destroyed.

1.3.4 Data Security

We follow the security principle of complete mediation and protect sensitive information during transit and storage to further control access of sensitive data.

- **Data in transit** The communication channels between machines are encrypted such that an attacker cannot learn sensitive information by eavesdropping. This includes clients and server communication over `https`, remote logging over TLS and secure copy over ssh[3] (`scp`) of the database backup and config files.
- **Data at rest** Static data like private keys are stored encrypted and whenever possible offline to protect them from a network attacker. If private keys need to be stored on the machine, they are encrypted and with read-only access which is restricted to the system administrator. The backup of the database is encrypted and stored on the backup machine using an asymmetric encryption scheme as described in 1.3.2. The backup scripts are only modifiable by the system administrator and also log and config files have restricted read and write access to the system administrator.

1.3.5 Defenses against common Web App Vulnerabilities

- **XSS** User-supplied data is validated before it is processed further, and it is escaped when displayed back to the user (e.g. as a default value in a form).

1.4 Components

1.4.1 Platforms

- **Main machine** The main machine runs on Debian[4] 10. It contains the web server and the core CA system which run in two separate Docker[5] containers following the security principle of compartmentalization (the organization of resources into isolated groups of similar needs). The MySQL[6] database containing user information and certificates is stored on the main machine, and a socket to the database engine is mounted on the containers. This allows both the web server and the core CA to interact with the databases, while keeping all data on the main machine.
  - **Web Server** The web server uses Apache[7], PHP[8] and OpenSSL. It serves the user and admin interface and allows for password- or

[3] `https://linux.die.net/man/1/scp`
[4] `https://www.debian.org/`
[5] `https://www.docker.com/`
[6] `https://www.mysql.com/`
[7] `https://httpd.apache.org/`
[8] `https://www.php.net/`

certificate-base user authentication. The web server communicates with the core CA through an HTTP API, using GET and POST requests.

  - **Core CA** We separated the core CA into root CA and sub-CAs for security guarantees. The root CA is our trust anchor. Its certificate is self-signed and the corresponding private key is stored and used offline to protect it from a network attacker. The certificates of the sub-CAs are signed by the private key of the root CA. The purpose of the private keys of the sub-CAs is to sign and revoke client certificates. The core CA provides an API for certificate generation, revocation and look-up, as well as look-up of the current CRL. This API is accessible only from the web container.

    Both the web and core containers communicate with the MySQL database only through a socket, which is mounted when the containers are built.

- **Backup machine** The backup machine runs Debian 10. It is used to store backups of all relevant data, including user information, generated and revoked user certificates, user private keys, and CA certificates and private keys. The backup machine is only accessible from the main machine (through cable), or with physical access to the server room.

1.4.2 Applications

- **User Interface** The web server serves the user interface which allows authenticated users to manage their user information and request or revoke a user certificate. The user interface provides a way for users to interact with web server through browser in a secure manner.

- **CA Administrator Interface** The web server serves the CA admin interface which allows the CA administrators to authenticate themselves using their digital certificate and afterwards to consult the current state of the CA such as the number of issued and revoked certificates and the current serial number.

- **Backup scripts** The script to dump the MySQL database, encrypt it and securely copy it to the backup machine is hourly called by the time-based job scheduler `cron`[9]. The MySQL user password, which is required by `mysqldump`, is provided through a configuration file that has restricted file access permissions (can only be read by root user).

- **Remote Logging** We use `rsyslog`[10] to forward log messages from the main machine to the backup machine. The rsyslog traffic is encrypted using TLS and remotely stored on the backup machine in the dedicated folder /var/log/asl-server.

[9] `https://linux.die.net/man/8/cron`
[10] `https://www.rsyslog.com/`

- **Database** The databases consists of an MySQL server running on the main machine. Database which receives connections from the web server is used to read, insert and update user information, while database for core CA is used to store user certificates combined with corresponding private key stored in the format PKCS #12.
- **OpenSSL** OpenSSL integrated in PHP function to provide core CA functionalities.OpenSSL is a software library for applications that secure communications over computer networks. It is widely used by Internet servers. OpenSSL comes with fundamental cryptographic functions and provides various utility functions. PHP has extension which binds functions of OpenSSL library for symmetric and asymmetric encryption and decryption, including PKCS12, X509 and other crypto operations.
- **Firewall** Uncomplicated Firewall (`ufw`) is used on the main and backup machine to block all incoming connections except for traffic over ssh, http and https (for the main machine) and ssh and tcp on port 6514 which is used for remote logging (for the backup machine). This follows the security principle of secure, fail-safe defaults (deny on default) and eases maintenance due to the simplicity of the firewall rules. Additionally, it reduces the attack surface presented to a network attacker.

1.4.3 Data Records

- **User Information** Information about user such as first and last name, e-mail address, the user ID and the hash of the password are stored in the user information database and displayed to the user through a web interface.
- **User Credentials** Credentials such as user ID and password are used during the user authentication process and verified against the information stored in the user database.
- **User Certificates and corresponding private keys** The user certificate can be requested by the user after successful authentication. It can afterwards be used for certificate-based client authentication.
- **Intermediate CA Certificates and corresponding private keys** The private key of the intermediate CAs is used to issue and revoke user certificates. The certificate enables everyone to verify that indeed the intermediate CA and no other entity issued the certificate.
- **Root CA Certificate and corresponding private key** The private key of the root CA is used to issue and revoke certificates for all intermediate CAs. Its certificate is self-signed as it is our trust anchor.
- **Certificate Revocation Lists** Crl is a list of digital certicates that have been revoked by the issuing certificate authority before their scheduled

expiration date and should no longer be trusted. It contains mainly the serial number of certificates that are revoked and further signed by secret key of issuing authority to provide integrity guarantee of crl. Crl allows two states of certificate, revoked (irreversible state) or hold (reversible state).

- **Log files** Interesting system states and security-relevant system events such as error messages and warnings are written into log files. These files help the system administrator to detect operational errors or determine and investigate deliberate attacks.
- **Configuration files** All system configurations are stored in configuration files. They allow for a structured collection of all relevant configuration data and simple access control.
- **Database Backup** Regular backups of the MySQL database are stored on the backup machine in ciphertext. They allow for data recovery in case the primary database gets corrupted or is lost.

2 Risk Analysis and Security Measures

2.1 Assets

2.1.1 Physical Assets

- **Main machine** The machine is located in an air conditioned server room to which physical access is restricted to authorized and trained personnel (e.g. System Administrators) since the physical integrity of the machine has to be guaranteed. Access from the Internet is limited to the Web Server container. The machine is equipped with an redundant power supply to support the goal of availability by minimizing the chance of a complete computer shutdown or failure.
- **Backup machine** The backup server is located in the same server room as the web server. Physical access to the room is restricted to guarantee the physical integrity of the machine. Network access is restricted to an internal network to communication with the CA machine and the machine is equipped with an redundant power supply to reduce the risk of data loss or data inconsistencies due to a complete computer shutdown or failure.

2.1.2 Logical Assets

- **Software**
  - **CA machine: Operating system** The CA machine runs on Debian 10. The operating system and software stack are checked and updated regularly by a certified system administrator.

- **CA machine: Firewall** The firewall `ufw` is configured to deny all incoming connections except on port 80 (`http`), 443 (`https`) and 22 (`ssh`). The ufw log file is monitored by the system administrator for unexpected or misbehavior.
- **CA machine: Database Service** The database is running MySQL. The corresponding log file is checked periodically and inspected for unexpected or mis-behavior and updates are installed regularly by the System Administrator.
- **CA machine: Web application** The web application runs inside a Docker container using Apache and PHP. Updates are installed regularly by the System Administrator.
- **CA machine: Core CA software** The CA software is run inside a separate Docker container using Apache, PHP and OpenSSL software. Openssl provides functionalities such as generating certificate signing requests, signing certificates and generating revocation list. The core CA communicates to the Web application container and serves a request related to confidential information from the container. Software udates are installed regularly by the System Administrator.
- **Backup machine: Operating System** The backup machine runs on Debian 10. The operating system and software stack are updated regularly by the system administrator.
- **Backup machine: Database backup** Backups of the database are created using `mysqldump` and encrypted using the public key of the backup machine dedicated for encryption. The encrypted database is securely copied to the backup machine using `scp`[11].
- **Backup machine: Remote logging** The CA machine uses `rsyslog` to reliably forwards its logs to the backup machine via TCP. The backup machine listens on port 6514 for incoming logging messages and writes them file. The network traffic is encrypted using TLS.
- **Backup machine: Firewall** The firewall `ufw`[12] is configured to deny all incoming connections except on port 6514 (`rsyslog`) and 22 (`ssh`). The ufw log file is monitored by the system administrator for unexpected or misbehavior.

- **Information**
  - **User Credentials** We require that a password is only known by the corresponding user (confidentiality), that the usernames and passwords in the database cannot be tampered with (integrity), and that they can be read by the web server and CA system at all times (availability).

[11] `https://en.wikipedia.org/wiki/Secure_copy`
[12] `https://wiki.ubuntu.com/UncomplicatedFirewall`

- **Private Key of Users** We require that the private keys are only known by the corresponding user (confidentiality), that the stored private keys (encrypted using a user-chosen passphrase) in the certificate databases cannot be tampered with (integrity), and that they can be restored from a backup in case of data loss or corruption on the main machine (availability).
- **Passphrase of Users** The passphrase is used to protect the private key of a user during storage. It is provided by the user at the submission of the certificate signing request, and maintained by the OpenSSL software. No storage is required on the CA Core. The passphrase is encrypted using the public encryption key and stored in the certificate database. We require that the passphrase is only known to the corresponding user (confidentiality), that the stored cipher text of the passphrase in the certificate database cannot be tampered with (integrity), and that it can be restored from a backup in case of data loss or corruption on the main machine (availability).
- **Private Key of intermediate CAs** Private key corresponding to the certificate of the intermediate CA. The private key is encrypted using a password that is stored in a configuration file with restricted file access permissions. We require that the private key is only know by the intermediate CA (confidentiality), that it cannot be tampered with (integrity), and that the intermediate CA has access to it at all times (availability).
- **Private Key of root CA** The private key corresponding to the certificate of the root CA which is stored offline and in a vault. We require that the private key is encrypted using a dual purpose algorithms that supports encryption and Message Code Authentication (MAC) to guarantee confidentiality and integrity. The private key must only be made available to sign or revoke the certificate of a intermediate CA which is also done offline (limited availability).
- **User Certificate Signing Request** The CSR[13] is generated internally and temporarily, containing user information and to be signed by the Client CA (intermediate CA). The CSR is not stored and discarded right after usage.
- **Certificates** This includes certificates of user, intermediate CAs and the root CA. User certificates are signed by the Client CA, whereas all the intermediate certificates are signed by the root CA who in turn has a self-signed certificate. We require that each entity has access to its corresponding certificate, as well as any certificate of the chain-of-trust that is required in order to confirm the validity of a presented certificate (availability), and that it is not possible to modify the information stored in the certificate due to the digital signature (integrity).

[13] `https://en.wikipedia.org/wiki/Certificate_signing_request`

- **Certificate Revocation Lists** This is a list of all revoked certificates containing meta information about the certificate and its serial number. It is used whenever certificate verification is performed. We require the CRLs to be signed by the corresponding CA (authenticity and integrity) and to be available during certificate verification.
- **Backup Encryption Key Pair** The public backup encryption key is used to encrypt dumps of the user and certificate database. The corresponding private key is stored offline and placed in a vault. We require the private key to be encrypted using a dual purpose algorithms that supports encryption and Message Code Authentication (MAC) to guarantee confidentiality and integrity. The private key must only be available to the system administrator to restore the encrypted config files or database backup (limited availability).
- **Passphrase Encryption Key Pair** The public passphrase encryption key is used to encrypt the user-chosen passphrase before storing it in the certificate database. The corresponding private key is stored offline and placed in a vault. We require the private key to be encrypted using a dual purpose algorithms that supports encryption and Message Code Authentication (MAC) to guarantee confidentiality and integrity. The private key must only be made available after a management decision to restore the encrypted passphrase in the certificate database (limited availability).
- **Root Password** The system administrator needs access to the root password to reconfigure and maintain the system. We require the root password to be only known to the system admins (confidentiality). Any possible external backup or storage is done by system admin outside of whole system.
- **User and Certificate Database** The user and certificate database contains user information such as first and last name, e-mail, user ID and the hash of the password, as well as a passphrase encrypted bundle containing user certificate and corresponding private key and the passphrase encrypted using the public encryption key. We require that the information in the database cannot be read or modified except by the system administrator or the corresponding user (confidentiality and integrity), and that a recent backup of the database exists at all time to assure availability of the data.
- **Backup Database** Once every hour a dump of the database is created and stored on the backup server. We require that the database backup is encrypted at all times to guarantee confidentiality of the stored data.
- **Log Files** The log files contain information about various services running on the machine. We require that they are automatically written to by all running services and can only be read by the system administrator. A recent backup of all security relevant logs must exist

at all times such that important information about the system state can be recovered (for example, after an attacker tried to cover his tracks by deleting logs) and the effects and spread of an attack can be analyzed by the system administrator.

  - **Remote Log Files** These are backups of log files stored on the backup machine. We configure the file access permissions such that only the sysadmin may read the content of these files (confidentiality).
  - **Configuration Files** Configuration files for services such as MySQL, rsyslog or PHP. We require that only the system administrator can read and write to these files by restricting the file access permission (confidentiality and integrity), that an encrypted, recent backup of critical configuration files is stored on the backup machine at any times (availability), and that the backup can be restored by the system administrator.
  - **Backup of Config Files** These are the backups of configuration files stored on the backup machine. We require that the backups are encrypted using the public encryption key (confidentiality).

- **Connectivity**
  - **Internet Connection of Web Server** The connection of the web server to the internet should be guaranteed, to allow users to revoke their certificate.
  - **Connection between CA machine and backup machine** The main and backup machines are connected using a physical cable, whose physical integrity should be guaranteed.

2.1.3 Persons

- **Employees/Informants** Employees and Informants are users of the CA service. They use the issued certificates for secure e-mail communication.
- **CA administrator** The CA administrator can consult the current state of the CA.
- **System Administrator** The system administrator is responsible for the system configuration and maintenance and has therefore root access to the main and backup machine.

2.1.4 Intangible Goods

- **Employee's Confidence** Employees must trust that the information which they exchanged within the company and with their informants are secure.

- **Companies Reputation** If the reputation of the company is damaged, informants might no longer cooperate with the company.
- **Availability of Service** Employees should be able to log in, generate and revoke certificates at anytime. Such that, if a private key is compromised, a revocation certificate can be issued and published immediately
- **Accuracy of Information** The CA meta-information available to the CA admin must be accurately represent the current state of the CA.

2.2 Threat Sources

In section 2.1 we identified the different assets of the company iMovies. In the following, we will discuss the relevant threat sources for these assets and describe their motivation.

- **Nature** Physical components of the system (hard disks, power and network grid) may fail or get destroyed due to natural disasters like fires or floods .
- **Employees** iMovies employees (including CA Administrator and System Administrator) can threaten the system, either by ignorance (insufficient training) or by malice (e.g. seeking revenge against the company).
- **Investigated Entities** Investigated entities are highly motivated to stop the investigation, for example by threatening company or informants, and to prevent the movie release. Their skill-level varies however they are often people holding some position of power and therefore have the means to hire others with the necessary skill set.
- **Script Kiddies** Since the system is connected to the Internet through its web server, it is exposed to attacks by script kiddies. They are motivated by the sense of achievement from hacking the system, inserting or extracting data or disrupting service for other users.
- **Skilled Hacker** Depending on the nature if the investigative reports, skilled hackers might want to compromise the CA system to learn the current state of the investigation or reveal the secret identity of the informant. They could use these information to blackmail the informant or the company to prevent the release of movies.
- **Governmental Agency** Depending on the nature if the investigative reports, governmental agencies might want to compromise the CA system to learn the current state of the investigation or reveal the secret identity of the informant. They could use these information to advance their own investigations or pursue legal actions.
- **Malware** Undirected malware might make its way into the system. The goal of such malicious software is to intentionally harm the system, disrupt

the service or even blackmail the service owner for financial benefits (e.g. Ransomware).

Organized Crime and Terrorists do not seem relevant threat sources for the company and are therefore not be considered.

2.3 Risks Definitions

We define the notion of *likelihood* as proposed in the book [1]. For the definition of *impact* we update the 3rd case to include legal actions which can be a consequence that informants face if exposed. For both definitions we the use qualitative categories *low*, *medium* and *high*.

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 severe legal actions, 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 or legal consequence.
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

For the sake of readability, we will abbreviate *Script Kiddies* as *SK*, *Skilled Hacker* as *SH*, *Investigated Entity* as IE, and *Governmental Agency* as *GA* in this section.

2.4.1 *Evaluation Physical Asset: Main and Backup Machine*

No.	Threat	Countermeasure(s)	L	I	Risk
1	Nature: Components break (e.g. hard-disk defect)	Replace defect parts and recover data from backup	*Low*	*Low*	*Low*
2	Nature: Power outage	Redundant power supply	*Low*	*High*	*Low*
3	Nature: Network outage	Cable redundancy	*Low*	*High*	*Low*
4	Employee, IE: Physically damages machines	Restrict physical access, Log access to machines for accountability, Policy for server room (no beverages, magnets, etc.)	*Low*	*High*	*Low*
5	IE: gains physical access to machines to learn information	Restrictive access control for server room, Strong Door, no windows in server room	*Low*	*High*	*Low*

Both machines reside at the same physical location. Thus, in case of a power or network outage, both machines will be affected.
Physical damages due to natural events, such as, fires, flooding, earthquakes have to be assumed during the construction of the building housing the servers. Power failures as a result of earthquakes are included in threat 1. It is assumed, however, that the space housing the servers is properly constructed and maintained and includes temperature control and ventilation systems. Installation of the backup server in a separate (protected by physical firewall) can lower the vulnerability of the system due to fire(s) and/or other natural events. The provision of a separate power cable (powered by a separate power line from the switchboard) can, further, minimize the risk of simultaneous power failure.
For threat No. 1, we recommend that the data are stored in suitable disk servers where a failure of one (or more) physical disk will not result in data loss; since there is a data reconstruction capability.

For threat No. 2, we recommend to management the purchase and installation of suitable UPS (Uninterruptible Power Supply) in order to maintain power on both machines. Thus, we ensure that in case of power failure we can maintain the service for some time, or (of the power failure is prolonged) we can power down the machines preserving the data.
For threat No. 3, we transfer the risk by purchasing a special assistance plan from our network provider, which guarantees the dispatch of a technician within 4 hours of the incident.

2.4.2 *Evaluation Logical Asset: Web Server*

No.	Threat	Countermeasure(s)	L	I	Risk
6	SK, SH, GA: performs SQL injection attack and reads, modifies or deletes content of database	Sanitize user inputs, Backup of database, Regular software updates, Monitoring, alerting	*Med*	*Med*	*Med*
7	SK, SH, GA: performs XSS attack	Escape and validate untrusted data before displaying	*Med*	*Med*	*Med*

2.4.3 *Evaluation Logical Asset: CA Core*

No.	Threat	Countermeasure(s)	L	I	Risk
8	SH, GA: gains access to CA machine, generates forged certificates, revokes valid certificates	Regular Software Updates, Monitoring and alerting, System Hardening	*Low*	*High*	*Low*
9	SH, GA: gains access to CA machine & causing fatal damage or information leakage	Regular Software Updates, Monitoring and alerting, Abort system information, invalidate all credentials and create anew	*Low*	*High*	*Low*

2.4.4 *Evaluation Logical Asset: Backup Machine*

No.	Threat	Countermeasure(s)	L	I	Risk
10	System Admin: stops automatic backup solution and disables remote logging	Monitoring, alerting, Log access to machine, More than one system admin to detect misbehavior	*Low*	*Med*	*Low*
11	SH, GA: gain access to Backup machine	Access control, Log access to machine, Reduce attack surface by turning off unused services	*Low*	*Low*	*Low*

2.4.5 *Evaluation Logical Asset: Firewall*

No.	Threat	Countermeasure(s)	L	I	Risk
12	System administrator: Accidentally or intentionally misconfigures firewall	More than one system admin to detect misbehavior, Well-trained system admin and code reviews to prevent accidental misconfigurations, fail-safe defaults and simple rule set	*Low*	*Med*	*Low*
13	SK, SH, GA: causes malfunction or exploits misconfiguration to bypass firewall	Regular software updates, Firewall logs and monitoring	*Low*	*Med*	*Low*

2.4.6 *Evaluation Logical Asset: Connectivity*

No.	Threat	Countermeasure(s)	L	I	Risk
14	System Admin: misconfigures network	Experienced and well-trained system administrator, Back-up scenarios	*Low*	*High*	*Low*
15	SK, SH, GA: performs a DoS attack	External DoS prevention service	*Med*	*Med*	*Med*
16	Nature: Network outage	Cable redundancy	*Low*	*High*	*Low*

2.4.7 *Evaluation Logical Asset: User Credentials*

No.	Threat	Countermeasure(s)	L	I	Risk
17	Employee: leaks user ID and password	Employee training for correct password handling Report to system administrator to invalidate credentials and prevent further damage	*Low*	*Med*	*Low*
18	Employee: forgets user ID or passwords	Second authentication option using certificates	*Low*	*Low*	*Low*
19	System Admin: forgets admin credentials	Documentation of all configuration / machines	*Low*	*Med*	*Low*
20	System Admin: leaks admin credentials	Non-Disclosure Agreement	*Low*	*High*	*Low*
21	SH, GA: gains access to databases and reads/modifies user credentials	Regular Software Updates, Monitoring of database updates, System Hardening, Code Reviews	*Med*	*Low*	*Low*
22	SH, GA: hijacks user session to read user information	HTTPS to prevent Man-in-the-middle attacks, Use of long, non-guessable session IDs	*Low*	*Med*	*Low*
23	SK, SH, GA: guesses/brute forces user password through login interface	Strong password policy	*Low*	*Med*	*Low*
24	Malware: deletes or encrypts user credentials in database (ransomware)	Database Backups on backup server without internet connection, Regular software updates, Incident response policy	*Low*	*Med*	*Low*

2.4.8 *Evaluation Logical Asset: CA Administrator Credentials*

No.	Threat	Countermeasure(s)	L	I	Risk
25	CA admin: leaks/loses certificate used for authentication,	Contact system admin and generate new certificate, revoke old one	*Low*	*Low*	*Low*
26	SK, SH, GA: steal CA admin certificate	Monitoring login status, revoke certificate when necessary	*Low*	*Low*	*Low*

2.4.9 *Evaluation Logical Asset: Private Key of User*

No.	Threat	Countermeasure(s)	L	I	Risk
27	Employee: Accidentally leaks private key	Protect private keys with passphrase	*Med*	*Med*	*Med*
28	Employee: Intentionally leaks private key	Non-Disclosure Agreement, System admin revokes compromised user certificate	*Low*	*Med*	*Low*
29	SH, GA: gains access to certificate databases and reads private key passphrases and private keys	Private keys encrypted with passphrase, Passphrase is encrypted using dedicated pubic encryption key, Backup of certificate database	*Low*	*Med*	*Low*
30	SH, GA: hijacks user session to read user private key	HTTPS to prevent Man-in-the-middle attacks Use of long, non-guessable session IDs	*Low*	*Med*	*Low*

2.4.10 *Evaluation Logical Asset: Private Key of Intermediate CAs*

No.	Threat	Countermeasure(s)	L	I	Risk
31	SH, GA: gains access to CA core machine and reads CA private key	Regular software update protect private key by passphrase	*Low*	*High*	*Low*
32	SH, GA: gains access to database and writes certificate revocation list	software update Sign certificate revocation with secret key and protect from random writes regenerate CRL once broken	*Low*	*High*	*Low*

2.4.11 *Evaluation Logical Asset: Private Key of Root CA*

No.	Threat	Countermeasure(s)	L	I	Risk
33	IE: hires someone to break into company and steal private key	Place storage device of private key in vault, Have video surveillance	*Low*	*High*	*Low*
34	Employee: intentionally leak or sell private key	Place storage device of private key in vault, Have video surveillance	*Low*	*High*	*Low*

2.4.12 *Evaluation Logical Asset: Certificate*

No.	Threat	Countermeasure(s)	L	I	Risk
35	System Admin: issues user or CA admin certificate to illegitimate entity	Certificate revocation mechanism, Do background check during hiring process, Monitoring of logs	*Low*	*Med*	*Low*
36	Malware, SK: deletes or encrypts certificates of intermediate or root CA	Backup of all CA certificates on a backup server without internet connection, Regular software updates	*Low*	*Med*	*Low*

2.4.13 *Evaluation Logical Asset: Certificate Revocation Lists*

No.	Threat	Countermeasure(s)	L	I	Risk
	SH, GA: Inserts valid or removes revoked certificate entry from list	Restrict write access for root, System admin monitors system to detect abnormalities	*Low*	*High*	*Low*

2.4.14 *Evaluation Logical Asset: Private Key of Asymmetric Pair*

No.	Threat	Countermeasure(s)	L	I	Risk
38	Employee: intentionally leak or sell private key	Place storage device of private key in vault, Have video surveillance	*Low*	*High*	*Low*

2.4.15 *Evaluation Logical Asset: Configuration Files*

No.	Threat	Countermeasure(s)	L	I	Risk
39	System Administrator: accidentally deletes config files	Experienced system admin, Backup of config files	*Low*	*Med*	*Low*
40	SH, GA: gains access to main machine and reads/writes to config file	Restrict access to critical files to read-only for root Backup of config files System admin monitors system to detect abnormalities	*Low*	*Med*	*Low*

2.4.16 *Evaluation Logical Asset: Log Files*

No.	Threat	Countermeasure(s)	L	I	Risk
41	System Administrator: accidentally deletes log files	Experienced system admin, Backup of log files	*Low*	*Med*	*Low*
42	SH, GA: forges, modifies or deletes log messages	Restrict file access to root	*Low*	*Med*	*Low*

2.4.17 *Evaluation Personnel Asset: System Administrator*

No.	Threat	Countermeasure(s)	L	I	Risk
43	Suffers from illness or accident, unexpected unemployment termination	Contractual requirement for precise documentation of machine configurations and passwords	*Low*	*Med*	*Low*
44	Unintentionally misconfigures system, deletes data	Experienced and well-trained system administrator, Backup of data	*Low*	*High*	*Low*
45	Bribery, corruption, giving confidential data to third parties	Non-Disclosure Agreement, Access to encrypted backups only with authorization from management	*Low*	*High*	*Low*

2.4.18 *Evaluation Intangible Asset: User Confidence*

No.	Threat	Countermeasure(s)	L	I	Risk
46	SH, GA: Loss of user confidence due to theft of confidential information	Trained personnel, State-of-the-art security measures, External review of software and computer system	*Med*	*High*	*Med*

2.4.19 *Evaluation Intangible Asset: Companies Reputation*

No.	Threat	Countermeasure(s)	L	I	Risk
47	Employee: Loss of company reputation due to the disclosure of confidential information (e.g. identity of informant)	Non-Disclosure Agreement Log access to data Background check during hiring process	*Low*	*High*	*Low*

2.4.20 *Evaluation Intangible Asset: Availability of Service*

No.	Threat	Countermeasure(s)	L	I	Risk
48	SK, SH, GA: Loss of availability of service due to an DoS attack	External DoS prevention service	*Med*	*Med*	*Med*
49	System Administrator: disrupt service by misconfigurating the system	Experienced and well-trained system administrator, Test configuration first on test system	*Low*	*High*	*Low*

2.4.21 Risk Acceptance

No. of threat	Proposed additional countermeasure including expected impact
6	We lower the impact by in-time mitigation through monitoring logs and detect anomalies in use of services.We lower the likelihood of the threat by placing our web server behind a firewall that filters out malicious communication
7	Enables XSS filtering. Implement a strong Content-Security-Policy that disables the use of inline JavaScript.
15, 48	We lower the impact of the threat by hosting our web server in the cloud where we can scale the number of resources based on the number of requests
33	We lower the likelihood that the user confidence decreases by certifying our system to known ISO standards.

References

[1] David Basin, Patrick Schaller, and Michael Schläpfer. *Applied information security: a hands-on approach*. Springer Science & Business Media, updated 2019.