System Description and Risk Analysis

Qiyuan Liang Sophie Altenburg Matteo Oldani
Andrea Byku

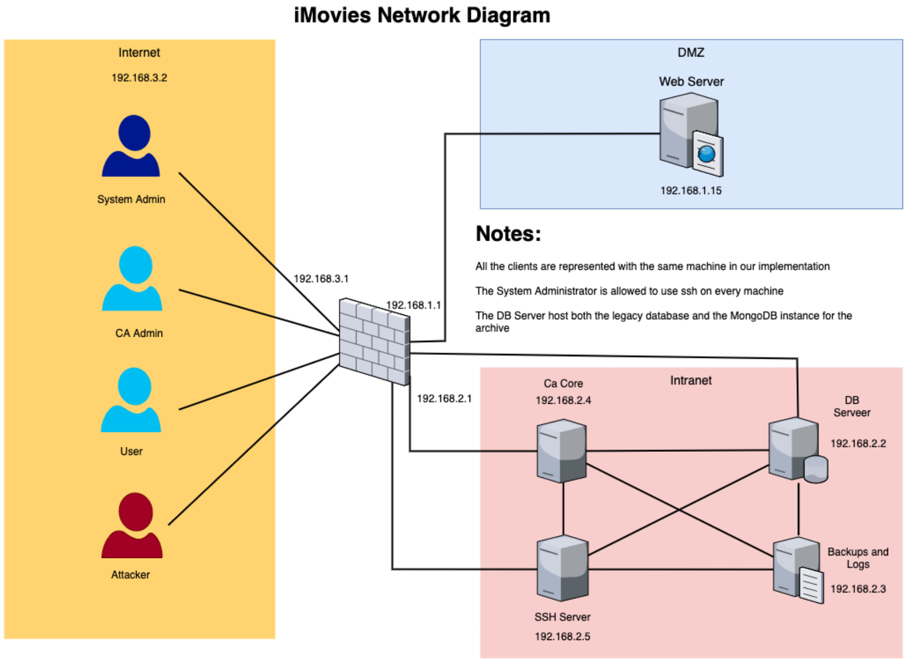
October 28, 2022

Contents

1 System Characterization

1.1 System Overview

The system has the goal to provide a certificate authority (CA) for iMovies, a movie production company that focuses on investigative reporting. Through this system, the company's employees are able to request and revoke digital certificates, that are used to communicate securely and to log in to the webserver by users.



Notes:

All the clients are represented with the same machine in our implementation

The System Administrator is allowed to use ssh on every machine

The DB Server host both the legacy database and the MongoDB instance for the archive

Figure 1: The system overview. All clients can connect from the Internet. For the development purpose, we cannot obtain a static public IP for our service. The internet is simulated by a different private subnet (192.168.3.0/24) from the DMZ and internal network. The traffic first goes through a firewall. The firewall also serves as a NAT. In other words, the outsiders only know the public IP address of the firewall, and the IP address of the webserver remains private. The webserver is placed within a demilitarized zone (DMZ). All core functionalities are placed within the intranet, including CA core, DB server, SSH server, backup & log server. Note that the system admin is allowed to connect to all machines via SSH through the SSH server. For the system admin to do so, he first connects to the SSH server (the jump host), then connects to all other machines.

The system is composed of 6 subsystems: firewall, webserver, CA server, SSH server, DB server, and Backup & Logging server. The system can be used in two different ways: from outside the company and from the company's internal network (usage limited to employees of the company). The central firewall separates the outside from the inside and blocks prohibited connections.

1.2 System Functionality

The system that is presented in this analysis is a simple certificate authority (CA) tailored for the company iMovies. It provides employees with digital certificates. The goal is to support the companies mission of investigative research, and therefore to help secure the company's e-mail communication for which the certificates will be used.

The CA system is based on the following roles:

- **Users** are all users that interact with the web page or the system in general. Authorized users for the CA functionality are the employees of iMovies and the technical staff maintaining the CA system.
- **CA admin** is a role assigned to an employee of iMovies giving additional rights to monitor the status of the certificate issuance.
- **System admin** denotes the technical staff that is taking care of the machines and software our system consists of.

The CA system offers the following functionality:

1.2.1 Access

1. Users have remote access to the certificate website and the possibility to log in either via their company password and username or via valid certificates.
2. The CA admin can access the CA admin panel showing the status of the existing certificates by logging in only via his/her certificate.
3. The system administrator has remote access via SSH to all machines to a user that has privileges for all sysadmin functionality, including the firewall, webserver, CA server, databases, and backups & logs. He/she first connects to the SSH server and from there to other machines.
4. The only entry point from outside the network is the SSH server which will grant access only via TLS based authentication using the certificate of the sysadmin.

1.2.2 Data management

5. The system is integrated with the legacy database containing the employees' user data.

6. An authenticated user is shown his/her account page showing all his/her data.

7. An authenticated user can add or edit his/her personal data (see 11 and 12).

8. The CA admin can additionally open a certificate overview page.

9. The certificate overview page shows the number of issued certificates, the number of revoked certificates, and the current serial number.

1.2.3 Certificate issuing

10. The system provides the possibility to issue certificates for all users based on their UID stored in the legacy database.

11. In case, a user's data is changed, the certificate assigned to the user stays valid as the UID is not changeable.

12. For every user, only the most recent certificate is valid: when a new certificate is issued the previous one is revoked and thus added to the certificate revocation list. This minimizes the attack surface with only one key per person to protect, applying the Minimum Exposure principle.

13. A user has the possibility to download the new certificate in the PKCS#12 format from the website.

14. After the user downloads the certificate, he/she can install the certificate in his/her browser, such that the login via certificate is enabled. The user has the option to select the certificate to be used when visiting the webserver later.

1.2.4 Certificate revocation

15. The system offers the possibility for users to revoke their own certificate on the account page.

16. The homepage of the company website publishes an up-to-date certificate revocation list (CRL). This list is available to everyone, even not logged-in users.

17. There are two CRLs, one for the intermediate CA (ICA), the other for the root CA. The ICA is in charge of signing certificates for users, see section 1.3.5. The root CA is in charge of issuing certificates for all machines and ICAs. If the ICA's private key gets compromised, the root CA can revoke the ICAs certificate and it will show up in the root CRL.

1.2.5 Backups

18. The system provides backups to ensure that (encrypted) data can still be accessed in case of loss of keys or servers, see section 1.3.5.

19. The system stores all (private) keys and certificates in a separate MongoDB database and keeps a dump of them in an archive on the backup server.

20. The system provides automated backups for the database content.

21. The backup server contains all server configurations updates by the sysadmin when he performs manual changes.

1.2.6 Non-functional requirements

22. The access to the CA functionality, data, configuration and keys is restricted to the authorized personnel.

23. There is regular security awareness training to teach employees how to store keys securely.

24. There is access control in place on all components of the system.

1.3 Security Design

1.3.1 System design in general

The system security design is inspired by the security principles presented in the book [1]. In particular, we adopted the compartmentalization principle and deployed our system on different machines. The minimum exposure and least privileges principles were both taken into account while creating the network. It is only accessible through a firewall and security is enforced by the user permissions on services that they can run. In our system, all the data is properly encrypted during the transit period with TLS. For data at rest, we use a strict file permission setup and encryption. The leading guidelines for the design are compartmentalization, least privilege, and limited communication.

1.3.2 Authentication

1. **Key management**: System administrators can access the machines via SSH to the SSH server (jump host) in the internal network from which they can SSH into the internal machines. This SSH connection from outside the company intranet only allows authenticated certificate login. Because of this, every sysadmin has a private USB stick, where his private key needed for the SSH connection is stored. The private key is also protected using a passphrase known by the sysadmin only. This practice increases security by lowering the possibility of having the system administrators' keys leaked online or the loss of the USB stick that leads to the loss of the

private key.
Regarding the user's key pairs, they are stored in an archive on the database server and are backed up after being encrypted that only the sysadmin possesses (in case of necessity to access the backup).
This and the private key used to create the root CA certificate is stored offline to prevent leakage, so applying the Minimum Exposure principle, see section 1.3.5.

2. **Session management**: Sessions are managed by the Django middleware. The session cookie will be stored locally in the client browser and the context will be stored on the MySQL DB in additional tables. Sessions automatically expire after 10 minutes to increase the security in case the session token is stolen.

3. **CSRF mitigation**: Our system uses the CSRF Middleware provided by Django to prevent CSRF attacks. Indeed, every POST request includes a CSRF token which is automatically generated by the middleware.

1.3.3 Access control

In all iMovies servers, the Role Based Access Control (RBAC) is applied with the least privilege principle. For the external users, an RBAC is associated with each user session while access to the resources is decided in the backend. All components of the system are configured to run with the least privilege required to fulfill the tasks as the least privilege principle suggests. There is also physical access control when it comes to the hardware of the internal network. It is kept in a server room at iMovie's and the keys belong to the sysadmin. The cleaning staff needs to ask for him to open the room.

1.3.4 Security of data in transit

We ensure the confidentiality and integrity of the data in transit via the use of appropriate protocols. All the communications between our system's components are done over a secured connection over SSL or SSH. Every machine has its own certificate that has to be used to establish SSL connections.

Client (employees) to web server. Confidentiality and integrity between client and webserver are ensured via HTTPS to protect e.g. the password and username, and any user information that is involved during the connection.

Web server to CA core. The webserver and CA core communicate to negotiate new certificates. The CA server supports an API running over HTTPS, listening on port 1390. Since it does not face outside, we have the freedom to choose any port as we want. Deviation from the standard HTTPS port makes it harder for an attacker (if any) to figure out the usage of the port, and attack it. For most attackers, it tries out the default ports first. To request a new certificate, the API call includes the user ID that should go into the certificate. The

CA core replies with the private key and certificate in the PKCS#12 format. Encryption for the connection is provided by HTTPS.

Web server and CA core to DB. The DB server hosts two databases, a MySQL database for the user data connected with the web server and a MongoDB for the private key archives used by the CA core. Both connections are run over TLS 1.3 authenticated with both the machines' certificates. The MySQL database is listening on port 6698 and MongoDB is listening on port 2799. Here the ports also deviate from the standard ports for both MySQL and MongoDB. It obfuscates what critical services are running on which ports, as attackers will initially attempt to exploit default values.

Communication to the Backup server and Logs. In regular intervals, the databases are creating dumps. The CA core additionally tracks all issued, requested, revoked and verified certificates. Also, there are operating system logs for the machines running the database, the CA core that are collected by the backup server. Backups, database dumps and logs are collected via SCP by the backup- and logging-server on port 22 every hour. Every day, backups that are older than a day are cleaned up automatically.

Sysadmin to all machines. Every machine will have a sysadmin user with root privileges. The system admin has SSH access to the SSH server (jump host) inside the internal network via his certificate. The login to SSH via password is disabled for the jump host. From there he can SSH into all machines with no certificate required. This lowers the attack surface as only the jump host is facing an open port to the internet, none of the other internal machines. This also follows the simplicity principle, it achieves the same functionality as enabling users to SSH to all machines. However, deploying the jump host facilitates management and control and improves usability. The sysadmin only needs to configure and protect the SSH server properly to offer security and accessibility.

1.3.5 Security of data at rest

The idea of least privilege is employed here. Only limited ports necessary for the functionality are open to allow access to the server, which minimizes the attack surface of the data stored in each of these hosts. The file permission is properly set up such that normal users are not allowed to read/write to the file content.

1. **Database server** is used to store the relational tables, including employee's id, name, email, and hashed password. It also holds the context and sessions for the webserver, including who has CA admin privileges. Storing issued certificate files in the format of PKCS#12 directly in the machine is neither integrity guaranteed nor easy to manage, therefore, we adopt MongoDB as the document database for the private key archive.

MongoDB is deployed on the same machine as the **MySQL DB** and we took some precautions to secure both servers:

- **Data protection.** The encryption for MySQL/MongoDB is not directly enabled. For a real implementation for a client, we would use MongoDB enterprise which supports database encryption unlike the free version we are using for our prototype [2]. Instead we deploy strict file permission for all database files. Only user mysql/mongo/root can access the data. We assume to have disk encryption to protect all our data against any physical access to the machine, e.g., stealing the computer and decrypting the content inside the drive. For implementation on physical machines instead of in Virtual Box, this would not cause memory space problems and disk encryption would be feasible and enforced.
- **Changing the database default settings.** (1) Change default MySQL / MongoDB port to other than 3306/27017. Attackers have to first figure out the usage of the port to perform an attack; (2) Setting up the username/password for the database properly. If the attacker wants to access the database to read all tables, he has to figure out the username/password at the first place; (3) Enforce password expiration and reduce the password lifetime for all users. In case there is a password leakage, the malicious users can only have limited time for accessing the database; (4) Configure client-server encryption with TLS 1.3 and do not allow downgrade attacks by limiting the server to only allow TLS 1.3.

2. **CA server** offers the certificate issuing/revocation service. To minimize the attack surface, we have the root CA's private key stored offline. The root CA's private key is stored on a USB stick that the system admin possesses and keeps in a safe. The USB is encrypted, which requires the system admin to enter the passphrase at every plugin of the USB to the computer. To maintain the functionality of the CA server, we introduce an intermediate CA different than the root CA. All the certificates are signed by the intermediate CA's private key. This mechanism is useful when the intermediate CA's private key gets compromised. The previous intermediate CA's public key will be revoked and all certificates signed by the previous intermediate CA are no longer valid. We regenerate the new intermediate CA's key pair using the root CA's private key. The clients do not have to change the root of trust, they can still rely on the root CA's public key to verify the validity of a certificate. The certificates themselves are not stored on the CA server. In the CA server, we only need to protect one file, which is the intermediate CA's private key. The best practice will be to rely on some hardware protection that offers tamper-resistant secure storage. However, such features are not available in the virtual machine, we delegate protection to the layer of the Operating System. The intermediate CA's private key is configured such that only the file

owner is allowed to read the file. This privilege is held by the sysadmin.

3. **Backup server** is the last chance that we are able to recover our database and services if anything goes wrong. It should not be accessed often such that only in rare cases it gets read by only the system admin. In case of complete destruction of the infrastructure, there's a GitHub repository available to the sysadmin with instructions on how to set up the system.

4. **Log server** does not require high confidentiality, while the logs' integrity is extremely important. The nature of the logs is that no one is supposed to change previously written entries. Therefore, we configure the files as append-only using the system flag. Since only root users are allowed to change the flag, the possibility of an attacker modifying previously written logs is limited. We further restrain the file permission that only file owners can read/write the logs, other users only have the read access.

1.4 Components

All the machines used by iMovies to host the system are placed inside a locked room thus physical access is restricted only to sysadmin which holds the keys. The cleaning staff has to ask for access.

Firewall The firewall is hosted on a machine running *Debian 11*. The firewall has two main functionalities: routing and proxy. First, it serves as a router between three different networks, namely the internal network, the DMZ network, and the internet. The traffic is filtered and routed between these networks. Second, it serves as a proxy. The webserver and the SSH server do not have their public IP addresses. For people from the internet, e.g., employees and sysadmins, to access these services, they have to contact the firewall's public IP. The firewall then translates the packet with the target IP for the webserver and SSH server. When receiving incoming packets with the public IP of the firewall as the destination IP and 443 as the destination port, the firewall will forward the packets to the webserver. Similarly, when receiving incoming packets with the destination IP of the public IP of the firewall and destination port 22, the firewall will forward the packets to the SSH server. The functionality is achieved by NAT using `iptables`. There is another security mechanism implemented in the firewall which relates to the principle of Fail-Safe Defaults. The firewall is following a white-list approach which means that unexpected traffic is by default denied. Should a rule in the firewall stop functioning correctly due to an update or another event, the firewall will rather block more traffic, but never allow unintended traffic.

Web Server The webserver is hosted on a machine inside the demilitarized zone (DMZ) of the iMovies network. It is responsible for handling all HTTPS requests from the internet. The webserver runs on a *Debian 11* using *Nginx 1.18.0* which connects to a *Django* web application through *Gunicorn 20.1.0*.

The web server is under a firewall that allows requests only on port 443 (for HTTPS connections) and port 22 (for sysadmin access). The firewall will allow access to the 22 port only from traffic inside the iMovies' intranet. HTTP requests are rejected. The webserver is also able to handle the connections with the MySQL server, which holds both the information of the users and the ones requested by Django (e.g. session ids), and with the CA Server to serve all the certificate management functionalities. To adhere to the traceability principle all access and errors are logged and then securely stored by the Backup Server.

Core CA The Core CA is a flask application deployed using Gunicorn and Nginx. The CA server runs on Debian 10 using *Nginx 1.14.2* with *Gunicorn 20.1.0*. It listens to the port 1390 with TLS connection. It handles certificate creation, revocation, and verification for the system. The Core CA is deployed with HTTPS, so the communication between the Web Server and Core CA is encrypted. Any wiretapper inside the network cannot sniff the traffic going on. The Core CA communicates with the MongoDB for storing the signed certificates and private keys of users. Similar here, the communication is protected with TLS connection.

MySQL and MongoDB Databases *MySQL 8.0.27* is listening on port 6698 holding the user data and the sessions and context for the web application. The MySQL database requires TLSv1.3 authenticated with the DB server's and webserver's certificates. We chose *MongoDB 4.4* as the version of our document database because of compatibility issues with version 5.0. Some CPU instructions are not supported by all Virtual Box environments we tested. The MongoDB is used as the private key and certificates archive and listens on port 2799.

Backup The backup server is hosted inside of the internal network. It is run on a Debian 11.1.0 machine

with basic functionalities that performs a cronjob by performing *rsync 3.2.3* via SSH connection. We perform scheduled backups every hour of the relevant files in the machines such as application logs, application configuration files. The pulling mechanism does not perform the backup of the machines at the same time, but each machine is being backed up at a certain minute within the hour. The Backup machine pulls files from:

- Web server
- CA server
- Database

Furthermore, we follow a non-incremental approach and we store in each designed directory older backups, consequently having different states during the day of the same backed-up machine. The files that need to be backed up are listed in different files inside the machine, which can be simply modified, allowing easy management. Nevertheless, we perform a cleanup mechanism which consists in deleting permanently backup folders that are older than 1 day, reducing the amount of data stored in the machine. To automate the whole process of backing up, we decided to make the SSH connection passwordless by adding the sysadmin private key in the .ssh folder. Thus, the system admin can undirectly reach the machine by SSH using his private key. Besides the data in transit that is being encrypted through the use of SSH, the backup data stored is not being encrypted because the hard drive which contains the data is being already encrypted. In conclusion, the system with the backup server fully follows the Traceability principle because logs multiple relevant data and saves the traces of the main activities performed, allowing the reconstruction of the whole system's history by a central administration and the Usability principle because the scripts that perform the backups are easy to understand and to modify eventually.

Client The client machine is built to simulate both a normal employee and a sysadmin. Regarding the employee, we installed the chromium-browser which can be used to test the webserver in all its functionalities. Indeed, a user can install his own certificate inside the browser. On the machine is also present the ssh client open to the internal network which should be used by the sysadmin to test the possibility to login in all the iMovies' machines. Furthermore, inside chrome we have installed the self-signed certificate of our CA, indeed the HTTPS connection is shown as secure. The client network interface is configured to be in a different network outside of iMovies' one.

SSH Server The SSH server is hosted on a machine running Debian 11. This is the only ssh entry point from outside the iMovies intranet for the sysadmin. The server has the public key of the sysadmin inside the authorized_keys and it stores the associated private key to allow him to ssh into other machines. SSH login with password is disabled for every user.

2 Risk Analysis and Security Measures

2.1 Assets

2.1.1 Physical Assets

1. **Network connectivity.** The cables, routers, and switches enable access from outside the company. The company's routers building the firewall are directing packets from and into the internet. Inside the company, there is a LAN, all machines are connected with cables and switches.

2. **Internal networks.** The webserver runs in a different internal network separated from the other machines to limit internet exposure to the latter. Both networks are based on Ethernet and are data layer switches. The

firewall routes traffic between the networks and the internet. The networks and the firewall are essential for the CA system to function and to be reachable by users.

3. **Machines for Firewall/Web server/Backup server/Database server/CA server.** Different services are provided using different machines located within the company. Except for the backup server which is a very important security mechanism, all machines are business critical.

    - **Firewall**: it is the entry point from the Internet into the internal network and restricts access to the intranet. The services that will be allowed to access from the internet are SSH with TLS certificates (port 22 on all the machines) and the webserver (ports 443 only on that machine).
    - **Web Server**: it is responsible for hosting the web interface that allows employees to request and use their certificates. It also contains the protected interface that CA administrators can use to manage certificates.
    - **Backup server**: it stores the database dumps for both the user (and webserver context) database and the private key archive. It contains copies of the configurations of all systems. Additionally, it holds log files of all applications and systems.
    - **Database server**: it holds two databases. One is the database that stores employees' information (also password hashes) as well as the webserver context and sessions. The other one holds the archive that stores copies of the private keys and all certificates issued.
    - **CA Server**: it the machine that hosts the software able to issue and revoke certificates.

2.1.2 Logical Assets

1. **User database (MySQL)**: users' personal data, password hashes, session tokens and the context of the webserver.

2. **Private key archive (MongoDB)**: this is the archive that contains every pair {private key, certificate} issued by the company.

3. **CA Core server**: it is the software responsible to issue and revoke certificates. It also provides a protected interface which only CA administrators can access where they can manage the existing certificates.

4. **Backup files**: These files contain both the backups for the employees' information stored in the database and the backup of the private archive.

5. **Logging entries**: these are the files that contain all the logs about the login in the Web server, certificates issued and accesses to backups, private archive and logs itself

6. **Web application**: managing the user flow being an interface for the system.

7. **Availability**: for the company the availability of the service is a relevant asset strictly related to the connectivity. The system must be able to operate for which the firewall machine, webserver, CA core and the databases are indispensable.

2.1.3 Persons

1. **Certificate administrators** In the previous text, a certificate admin was mentioned for better readability. For reasons of redundancy, there are multiple certificate administrators behind the singular term. They manage the current state of the certificate issuance. They monitor the procedures to make sure that the CA works as expected. Since CA Administrators require training, the loss of one of them comes with a high cost for replacement.

2. **System administrators** The same thing applies to the mentioned system admin. To avoid a single point of failure in case of e.g. sickness, there are multiple system administrators who maintain all the services and machines placed in the company. Indeed, they are the only ones able to access all the backups and to the logging server. Moreover, they are supposed to restore the private keys from the backup in case of loss of an employee's private key.

2.1.4 Intangible Goods

Reputation Reputation is an asset very hard to estimate in monetary impact. It often is the source of trust that allows a company to stay in business. Also the employees need to trust in the systems they are using or the system will fail its purpose. Since the certificates are supposed to later be used to encrypt email communication, the employees rely on the secure functionality. The employees must have strong confidence that their encrypted messages are indeed not visible to others to disclose confidential data in their communication which is critical for the business' mission. A leak of the database/backup with employees' private information and the possible decryption of the internal emails can lead to severe damage to the company and as well its public reputation. Competitors could gain a competitive edge by stealing the ongoing investigations as the competition is to release new and unique stories. They could as well diminish the credibility of iMovie's research. The company therefore requires to treat unpublished drafts confidentially relying on the security of the certificate authority.

2.2 Threat Sources

1. **Nature**: At the company, there are multiple running machines located. Earthquakes, floods, and blizzards should be taken into consideration for

possible unavailability of the service or the break of the system.

2. **Employees**: Although employees do not have special access to additional information, they can still leak all the communication that they participate in that including investigative reports or their access tokens. Possible bribing and job dissatisfaction can lead to employees damaging the company's assets.

3. **Script Kiddies**: The company web server is exposed to the internet, script kiddies can try to exploit the system motivated by the challenges or by destruction.

4. **Skilled Hackers**: Likely, some emails contain sensitive information about the investigative reports of the company. Skilled hackers can take the initiative to crack the system to gain inside information for monetary purposes. They can also attack the system for a challenge, but this possibility is low.

5. **Malware**: The system does not allow direct file upload, the possibility for malware to propagate inside the system or infect the internal server is low. However, the local device that users/system admins use to access the internal service can be infected and perform malicious attacks.

6. **Competitors**: The competitors of the company are considered a threat source here since they are interested in the state of the investigations or information sources of iMovies to gain an additional advantage over the competitor.

7. **Investigation targets**: Some investigations might target organized crime or highly influential people. iMovies does investigative journalism and therefore also reports stories that might offend them, e.g. publishing government corruption cases or investigating a missed person. iMovie's investigation targets want to evade legal consequences or might try to diminish the credibility of iMovies by leaking company-internal information.

2.3 Risks Definitions

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. Or the controls are strong enough to prevent an exploit from being executed.

[1]

Impact	
Impact	Description
High	The event (1) may result in highly costly loss of the company's tangible assets; (2) may lead to the full loss or long-term unavailability to the running service or internal data; (3) may lead to serious injury of the personnel; (4) may lead to severe damage to the company's reputation or interest.
Medium	The event (1) may result in a costly loss of the company's tangible assets; (2) may lead to the partial loss or medium-term unavailability to the running service or internal data; (3) may lead to injury of the personnel or acquainted persons; (4) may violate or harm the company's reputation or interest.
Low	The event (1) may result in some loss of the company's tangible assets; (2) may affect the company's reputation or interest.

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

[1] Risk definitions are closely inspired by the course book. [1]

2.4 Risk Evaluation

2.4.1 *Evaluation Physical Assets*

No.	Threat	Countermeasure(s)	L	I	Risk
1	A natural disaster breaks the connectivity and puts the company offline	Paid for by insurance	*Low*	*Low*	*Low*
2	A natural disaster destroys the hardware, including hard drives	The servers running the database and the web server are in a different office than the backup server so data can be restored. The hardware is paid for by insurance.	*Low*	*Medium*	*Low*
3	Accidental demolition by employees, e.g. cleaning staff and system admin	Redundancy as in No. 2 and insurance coverage	*Low*	*Medium*	*Low*
4	Physical access to servers via social engineering or by staff to physically break the system or to steal the hard drives	Building locking system and security awareness training. Configure the operating system to use hard drive encryption	*Low*	*High*	*Medium*
5	Aging of components leads to defects	Changing the components according to their status	*Low*	*Medium*	*Low*

2.4.2 *Evaluation Logical Assets*

Targeting all components

No.	Threat	Countermeasure(s)	L	I	Risk
6	Nature: destroys servers	Backup solutions in a different location than the database	*Low*	*Medium*	*Low*
7	Normal Employees misuse the system out of frustration	Normal users have limited access to the system and can only mess up their account. The system admin and CA admin are trusted.	*Low*	*Low*	*Low*
8	The CA admin misuses the system out of frustration	Backups and logs enable to restore the previous state and find malicious actions.	*Low*	*Medium*	*Low*
9	The system admin misuses the system out of frustration	There are two trusted system admins.	*Low*	*High*	*Medium*
10	A normal employee or the cleaning staff is bribed to steal a hard disk targeting the data on it.	Disk encryption avoids reading out the stolen disk.	*Low*	*Low*	*Low*

Targeting all components

No.	Threat	Countermeasure(s)	L	I	Risk
11	Hackers get access to the internal network via physical intrusion and attempt to get access to the machines	No sufficient host-based firewalls yet	*Low*	*High*	*Medium*
12	Malware infects the system admin's PC and from there the CA and web server, but **not the backups** and steals/compromises/encrypts it for a ransom	Patching systems as soon as an update is available, backups and a copy of the code base	*Low*	*Medium*	*Low*
13	Malware infects the system admin's PC and from there steals/compromises/encrypts everything **including the backups**	Patching systems as soon as an update is available	*Low*	*High*	*Medium*

Targeting the webserver

No.	Threat	Countermeasure(s)	L	I	Risk
14	Script kiddies gain control over the web server and make the software unavailable	System hardening and patching systems as soon as a patch is available	*Low*	*Medium*	*Low*
15	Hackers find or buy a zero day exploit for the Django version that we are using for the webserver. Is then able to control the webserver machine and compromise the functionalities. This will also lead to a compormised user database since Django has access to it.	Patching systems as soon as an update is available	*Low*	*High*	*Low*

Targeting the jump host

No.	Threat	Countermeasure(s)	L	I	Risk
16	Hackers compromise the jump host and gain access to every machine via SSH	System hardening and patching systems as soon as a patch is available	*Medium*	*Medium*	*Medium*

Targeting the databases

No.	Threat	Countermeasure(s)	L	I	Risk
17	Hackers get root access and find the data storage location of the databases. They then read out the private keys and make changes to the database	No database encryption yet	*Low*	*High*	*Medium*

Targeting the CA core and keys

No.	Threat	Countermeasure(s)	L	I	Risk
18	A hacker gains access to an employees private key through phishing and therefore can read and distribute all his mail and impersonate as him/her	Security awareness training for all employees	*Low*	*Medium*	*Low*
19	Hackers gain control over the CA core and issue themselves a certificate with which they can impersonate as a new or unknown employee	System hardening and patching systems as soon as a patch is available	*Low*	*High*	*Medium*
20	Hackers compromise the intermediate CA's private key and leak it. They then read out the private keys and make changes to the database	No database encryption yet	*Low*	*High*	*Medium*

Targeting the Backup Server

No.	Threat	Countermeasure(s)	L	I	Risk
21	Attacker who gains access to this key can remotely access to all the machines over SSH without password	System hardening and patching systems as soon as a patch is available	*Low*	*High*	*Medium*

2.4.3 *Evaluation Personnel Assets*

No.	Threat	Countermeasure(s)	L	I	Risk
22	The system admin is absent because of sickness or leaves the company	There are two system admins with equal rights so that no knowledge is lost. They keep good documentation.	*Low*	*Low*	*Low*

2.4.4 *Evaluation Intangible Assets*

No.	Threat	Countermeasure(s)	L	I	Risk
23	Competitors hire a hacker to gain an employees private key and spy on his/her mail traffic and publish the story earlier	System hardening and patching	*Low*	*Medium*	*Low*
24	A target of an investigation, e.g. organized crime or a government representative hires a hacker to expose company internal information to publicly diminish iMovie's credibility	System hardening and patching, security awareness training of employees	*Medium*	*High*	*High*

2.4.5 Detailed Description of Selected Countermeasures

Security design Our system is designed to support security by staying by keeping it simple and maintainable. There are no other applications introduced than the ones needed for the system functionality and any unnecessary ports are closed. This minimizes the attack surface. Applications are run with the smallest set of permissions and we protect sensitive files to be only readable by the intended user.

Network setup The network setup that we propose focuses on separation of the applications to different machines and networks. We have the webserver inside a DMZ, reachable by the internet and an SSH server inside the internal network reachable by the internet, but not our other applications and machines. Another advantage of this architecture is the separation of data and code as all functionality (sessions, user data, certificate creation) that can be separated from the web frontend is and is even located in another network. Even if an attacker gains control over the webserver, there is still additional effort needed to attack the internal network. We chose this setup as it minimizes the attack surface by keeping as many machines away from the open internet. It also supports maintainability as there are only two entry points which can be given special care regarding system hardening and not all of the machines are at risk the same way. We use TLS authentication over certificates for the more sensitive communications inside our networks between the webserver, CA core and databases to avoid MITM attacks and impersonation. Even if an attacker gains access to the internal network without compromising a machine, the damage is limited. We use a strong firewall to connect the networks with the internet and filter traffic. We went for the simpler implementation here to keep the setup simple. Instead of introducing a new system with a *pfsense* firewall, we decided for *iptables* because it is one system that the sysadmin is already familiar.

Security awareness training We take a global approach when it comes to secure design and also take the environment of the system into account, not just the code. To make use of our security mechanisms, it is important that the employees do not become the weakest part of the chain and eventually circumvent the system. Social engineering and phishing attacks become more and more sophisticated, therefore it is important to explain the risks and do regular security awareness trainings. Physical access to the machines is very critical, so we also rely on the employees of iMovie not to let strangers into the server rooms and always question the occasion of visits of external people to the company building.

2.4.6 Risk Acceptance

No. of threat	Proposed additional countermeasure including expected impact
4	Insurance. Afterward, a low risk remains.
9	The damage that can be caused by the sysadmin is substantial. However, there are multiple sysadmins. The other sysadmin has means to reconstruct the system after misuse. An additional copy of the backups is stored unavailable for both admins and there is a repository with the code to set up the system again. There will still be a notable downtime on the system in a few hours. A medium risk remains.
11	Additional countermeasures would be firewalls on all hosts, even inside the internal network. With that, a low risk could be achieved.
13	Sysadmin has very strong privilege in all machines. If his/her computer gets infected by malware, and the malware manages to get the sysadmin's private key, there is not much we could do here except for the additional backup of the system. An extra copy of all systems' configuration, code, and data is stored in an offline hard disk. If the system gets compromised, the stakeholder still has to possibility to restore all the information and services. The impact could be reduced.
16	System hardening of the jump host would be one possible countermeasure. Another would be to replace the jump host with a VPN connection, and deploy a strong authentication mechanism when login in via VPN. Then, the risk could be decreased.
17	Database encryption prevents attackers with physical access from reading out the database files. A low risk remains.
19	System hardening and patching. Additionally, an Intrusion Detection system could be good security support.
20	After the CA's intermediate private key is leaked, all user certificates need to be revoked, but not the machine's certificates as those were created with the root private key. Also, users need to log in once - other effects are neglectable.
24	A good company lawyer, system hardening and patching.

Possible improvements We are aware that our CA system only shows a prototype instead of a real system and there have been some decisions with the intend to keep the design simple, that we would have made differently for a production system with more time and resources. In a real system, logging is helpful, but due to the amount of data that a production system produces, we would set up automatic alarms for suspicious actions in the logs. For example, *fluentd* together with *elasticsearch* offer good logging support [3]. We would deploy an IDS in the firewall and also use more precise and specific firewall rules to reduce the attack surface even further. The enterprise version of MongoDB offers database encryption which we would happily use and then also encrypt the MySQL database. Backups could be done event-based depending on the traffic even after every change to the database or alternatively more often. The SSH server (or jump host) would be hardened very strictly, more likely not even be used and exchanged for a VPN connection into the internal network such that the internal machines have absolutely no interface to the open internet. Those are nonetheless time consuming and costly setups that go against the keep it simple principle for this course's setting.

References

[1] David Basin, Patrick Schaller, and Michael Schläpfer. *Applied information security: a hands-on approach.* Springer Science & Business Media, 2011.

[2] Inc. MongoDB. Mongodb manual - encryption at rest. `https://docs.mongodb.com/manual/core/security-encryption-at-rest/`, 2021.

[3] Fluentd. Fluentd elasticsearch. `https://docs.fluentd.org/output/elasticsearch`, 2021.