System Description and Risk Analysis

Jason Friedman Mei-Chih Chang Onur Veyisoglu
Sergio Roldán Lombardía

November 26, 2020

Contents

1 System Characterization

In this section, we introduce our system architecture, its functionality, its security requirements and the description of the elements forming the system.

Additionally, we include the available backdoors placed on purpose in our system.

1.1 System Overview

The company iMovies produces various kinds of movies, but specialises on investigative reporting. For that, the company needs to set up mechanisms to secure communication between company employees and their informants. Our goal is to set up a simple certificate authority (CA), that provides the employees of iMovies with digital certificates to secure their e-mail correspondence.

A graphical representation of our proposed system can be seen in Figure 1 at Page 4. The system is divided in diverse components including the Webserver, the Core CA, the Firewall, the Database, the Backup Server and the Setup Server.

- The Webserver allows the clients (employees/informants) to update their user information and issue or revoke certificates, while enforcing authentication and authorization in the process. Additionally, CA administrators can visualize useful data about the CA state by login in using its certificate.
- The Core CA is in charge of the actual issuance and revocation of the certificates, including the maintenance of the Certificate Revocation List. This component is the central actor of the system, as its stores and maintain information related to certificates.
- The Database is in charge of storing users information including hashed passwords, and this information can be updated by its rightful owner through the webserver.
- The Firewall is a simple machine in charge of the filtering and discarding of network packets depending on their target or port, among others. The firewall is configured such that we create a network separation between the demilitarized zone including the Webserver and the rest of the components of the system, such that security relevant components are not directly exposed to the internet.
- The Setup Server is in charge of the configuration of the internal components of our system. This allows for a single point of configuration without exposing internal machines directly to the internet.
- Additionally, the system includes the Backup Server in charge of storing backups of the Database, CA and the logs generated by the rest of the system components.

The Webserver will be accessible from any location over the internet via a standard web interface, strictly only allowing HTTPS connections using either a password or a certificate as client-side authentication. Users can visualize and modify their data, issue a new certificate or revoke an old certificate. Additionally, the server provides an additional interface for administrators to visualize confidential CA information, such as the last issued certificate serial number or the number of revoked certificates, this information is only accessible by a CA admin login in with its certificate. The Setup Server is the only other component reachable from the internet and can exclusively be accessed by system administrators. Additional host-based network rules are enforced in different components of the system following the "defense in depth" concept.

The system is defined in different components, each with a specific task following the compartmentalization principle.

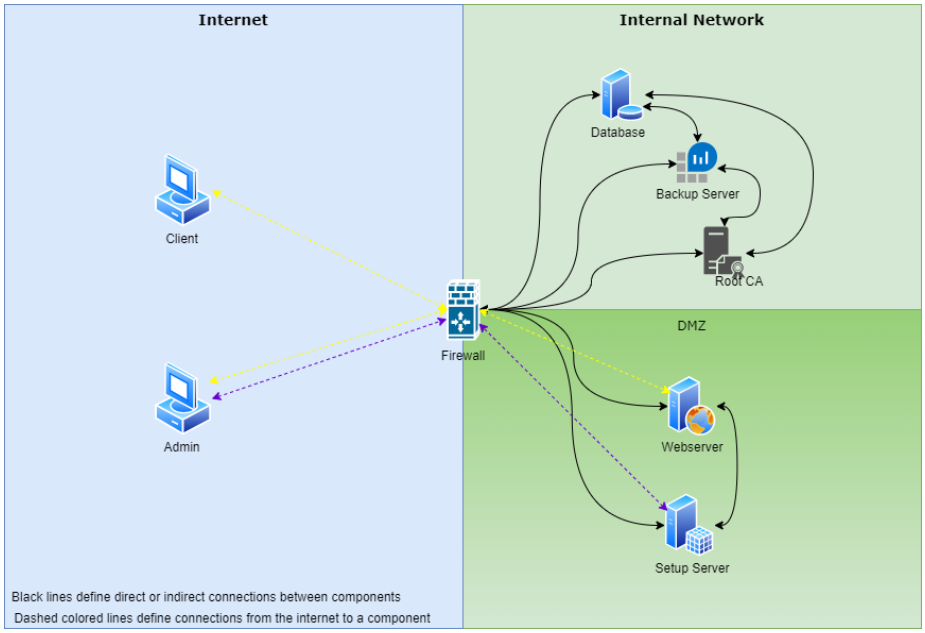

Figure 1: System Overview

1.2 System Functionality

In the following we list the core functionalities our system offers to both employees and informants of the iMovie company, from now on referred to as users.

1.2.1 User Authentication with Passwords

1. The client enters his user ID and password into the login web form.

2. The Webserver then requests the hashed password of the respective user from the Database Server and verifies it matches the entered password.

3. Only if authentication was successful, the client will be redirected to his corresponding user profile.

1.2.2 User Authentication with Certificates

1. The client accesses the webpage dedicated to the login with certificate through the browser.

2. The browser then uses the mutual authentication feature from TLS to both authenticate the client and the Webserver.

3. If the mutual authentication fails the Webserver will directly cancel the connection. However, if the TLS handshake was successful, this means that the provided client certificate was issued by the iMovies CA and is not present in the revocation list. In this case the webserver checks if the email from the client certificate matches any user email, and if so redirects the client to his profile.

1.2.3 Certificate Issuing

To simplify the management of user certificates on the Core CA server and improve general scalability, we allow each user to possess only one valid certificate. This does not conflict with the system requirements, as the users only need a single valid certificate to encrypt their email correspondence. If a new certificate is issued, while the client still has an old one, the old certificate is first revoked according to Section 1.2.4.

1. The user logs in via a web form according to Section 1.2.1 or Section 1.2.2

2. The user gets redirected to his or her profile, where he can update and review his personal Information. Any changes to his profile are stored on the database server through a HTTPS POST request.

3. To issue a certificate, the web server sends an HTTPS request to the Core CA containing the possibly updated user information and a passphrase chosen by the user.

4. The Core CA first revokes the old user certificate if there is one. Then the Core CA responds to the webserver with a PKCS #12 archive encrypted with the passphrase, containing the issuing CA's complete trust certificate chain, a newly issued user certificate and the corresponding user private key.

5. The user can now download the PKCS #12 archive from the webserver and install it in his mail account.

1.2.4 Certificate Revocation

1. The user logs in via a web form according to Section 1.2.1 or Section 1.2.2
2. To revoke a certificate, the webserver sends an HTTPS request to the Core CA containing the user ID of the certificate in question. The Core CA subsequently updates its certificate revocation list and publishes it to the webserver.
3. The webserver will adopt the updated certificate revocation list in cronjobs that are scheduled for once every 10 minutes. Any future authentication attempts with the revoked certificate will be detected by the webserver after the period of 10 minutes and result in a failed login.

1.2.5 CA Administration

CA administrators can only log in with their digital certificate, as described in Section 1.2.2. However, after successful login they get redirected to a dedicated profile, that shows not only the administrators personal information, but also the following CA's current state:

- Number of issued certificates
- Number of revoked certificates
- Current serial number

1.2.6 Backups

In every 4 hours, keys, certificates and mysql dumps with their respective logs are copied to a backup server. Moreover, syslog and auth.log files are also backed up to increase traceability. These procedure enables our system to recover from possible incidents in a short amount of time.

1.2.7 System Administration and Maintenance

The system administration must be performed using the Setup Server accessible from Internet via SSH. The SSH keys for this connection must be kept confidential and only accessible by company administrators which we could assume trustful. An administrator can enter the 'ansible' user in this server to be able to SSH to any other components in the system, as all components have been configured to only allow SSH accesses given the key stored in this user. With this configuration we aim to minimize the exposure of the internal components of the system, and allow administrators for a centralized way of managing the different components. The Setup Server additionally receives the name 'ansibleadmin' as Red Hat's ansible engine [1] has been configured to use the key material in the 'ansible' user to perform automatic configuration of the different

[1] https://www.ansible.com/overview/how-ansible-works

components of the system, and could be used to perform further maintenance by creating, modifying and executing ansible playbooks that target a single or a group of system's machines.

1.3 Security Design

In this section we further describe and analyze the system's security design by taking the 12 principles in Applied Security Book [1] into account, including access control, key and session management, and security of data at rest and in transit.

1.3.1 Authentication

iMovies enables their users to use two authentication methods. The privileged users have both password and certificate authentication enabled and prepared whilst, regular users have only password authentication initially. Upon first authentication to the system, regular users can also request a certificate to enable certificate authentication method.

In order to prevent brute force attacks, users that perform five consecutive authentication failures are blocked. In addition, the users are enforced to follow a password policy when they create a password or change it or set a passphrase for their accounts. For the password policy, OWASP Authentication Cheat Sheets[2] is consulted. Although it is up to the users to select their passwords, following rules are enforced to provide acceptable entropy keeping the *maximizing secrets* principle in mind:

- Passwords cannot be shorter than 8 characters.
- Passwords must contain at least one upper case and lower case letters.
- Passwords should contain at least one number.

1.3.2 Security of Data in Transit

To provide secure transmission of data, we only allow HTTPS connections to our web pages and we enforce HTTP Strict Transport Security in order to prevent session downgrade attacks. Moreover, all communication inside the network and between services are encrypted using SSH (backup, administration) or HTTPS (CA). All HTTPS connections enforce strong ciphersuites and are mutually authenticated, with the exception of the HTTPS connection from the client to the webserver, where client authentication is optional for *usability*.

1.3.3 Access Control

In all iMovies servers, both Role Based Access Control (RBAC) and Discretionary Access Control (DAC) is applied with *Least Privilege* principle.

For the external users, a Role Based Access Controls are associated with each user session and access to the resources are decided in the back end.

All components of the system are configured to run with the least privilege required to fulfil the tasks as *Least Privilege* principle suggest.

1.3.4 Data Integrity

Following the *Traceability* principle, we need to protect the logs with respect to integrity and completeness to avoid any possibility for an attacker to cover his tracks and delete or change logged events. Furthermore, respecting the principle of *Secure and Fail-safe Defaults* need to carefully protect the backups to ensure System Integrity in case a recovery is necessary, thus, the backup machine should only be accessible by system administrators and the rest of the core system should periodically send its latest information to this machine.

1.3.5 Network security

Following the *Compartmentalization* and *Minimum Exposure* principle, network is divided into segments by utilizing the iptables utility. Internal network is isolated from the traffic from the internet and a DMZ is created to place the webserver. This way, the internet facing assets are limited to a minimum.

To protect the availability of the system, additional rules are defined in firewall.

- Packets with bogus TCP flags are dropped.
- Spoofed packets are dropped.
- Connections per source IP address and new connections per second are limited.
- Protection against port scanning is applied.
- Brute forcing protection in network level in addition to application level is put in place.

1.3.6 Web application security

Following the *Open Design* principle and protecting against historical website attacks, web server and client are combined with Django REST framework to provide an open design with RESTful API and secured website.

Our web application can provide the following protections from Django framework.

- Cross site scripting (XSS) protection: Using Django templates at web application side can protect from XSS.
- Clickjacking protection: The web server is setup with dajngo.X-Frame-OptionssMiddleware to block the resource from loading in a frame no matter which site made the request.

- Cross site request forgery (CSRF) protection: Django has built-in protection against most types of CSRF attacks. CSRF protection works by checking for a secret in each POST request. This ensures that a malicious user cannot "replay" a form POST to your website and have another logged in user unwittingly submit that form.
- SQL injection protection: The querysets are protected from SQL injection since their queries are constructed using Django's query parameterization.
- Strict-Transport-Security (HSTS): The web server support HSTS, which tells the browser to load the site over HTTPS only.
- User-upload content protection: The web server sets up the limits of the file format to be uploaded (HTML or XML will be excluded).
- Host header validation: Only allowed web client can be accepted by the web server.
- Session security : Session data is stored at database table and webserver will use database-backed session engine to avoid session was stolen. This data is in form of JSON web tokens.

1.3.7 CA key management

The Core CA consists of a root CA, which is only used to generate two intermediate CAs (ICA). One ICA signs the certificates for the HTTPS communication between the system components and the other signs the actual certificates that are issued for the clients. Using an ICA instead of the root CA for the signing of the certificates ensures, that we can simply replace the affected ICA in case their private keys are compromised or lost.

The private key of the root CA and the private key of the ICA, responsible for HTTPS communication, are both only required when issuing a new Intermediate Certificate Authority (ICA) or adding a new host in the internal network. Thus we can store these keys offline to minimize *Exposure*. The private key of the ICA, used for the client certificates on the other hand, needs to be stored at the Core CA component, as it is required for each certificate issuance. For *simplicity* and *usability*, we store this ICA's private key in plaintext in the Core CA filesystem, as it is used for every certificate issuance, and revocation.

When issuing a new user certificate the core CA generates a 2048 bits long RSA user private key and the corresponding certificate signed by the ICA. Usually it is bad practice for the CA to store private keys of their clients. Instead the clients should themselves generate the private key on their own machine and only send the Certificate Signing Request (CSR) to the CA. However, to follow the *usability* principle the system needs to provide the functionality of restoring lost user private keys. The encryption of the user private keys works as follows, such that the user private keys can only be recovered by a System Admin in possession of the root CA private key (stored offline):

1. First we generate a 128 bit symmetric encryption key (SymKey) explicitly to encrypt this users private key.

2. Then we store the SymKey encrypted by the root CA public key using the root CA public key.

This method ensures both secrecy of the user private keys and efficient encryption because slow asymmetric cryptography is only used to encrypt the short SymKey instead of the long user private key. In most cases, when a user looses his private key, he can just issue a new certificate. However, if this is not enough, he can contact a iMovies system administrator, that is in possession of the root CA private key, to recover the lost user key.

1.3.8 Logging

Logging in iMovies systems is important to ensure *Traceability*. Essential events from running applications and servers are logged in order to make it easier to trace back when an abnormal event occurs. Traceability is also a prerequisite for Accountability[1]. This makes logging even more critical since it enables the company to link the actions to subjects especially in crisis times.
Considering *Single Point of Failure* principle, all logs are sent to the backup server in an encrypted channel (SSH) to prevent a situation in which logs are not accessible. In order to keep our system as simple as possible as *Simplicity* principle suggests, it is decided not to keep and maintain a dedicated log server. All the logs are sent and backed up at Backup Server every 4 hours starting at 00:00 UTC.

1.3.9 Maintenance

All components and their configurations can be maintained by system administrators either by connecting to them over ssh through the Setup Server or by adjusting and rerunning the ansible playbooks. This gives the administrators the possibility to easily managing the entire system from one location adhering to the *Simplicity* and *Usability* principles.

The software and libraries on the components need to be regularly patched by the system administrators, to make sure revealed bugs or vulnerabilities can not harm the system.

1.4 Components

In this section we list all the system components and their interfaces, subdivided into platforms, applications and data records.

1.4.1 Platforms

Web Server: Web Server is setup by Django REST Framework in python. It follows security protections from Django.

Core CA: The Core CA is a python module that handles certificate generation, storage and revocation for the system. To make use of its services, the webserver component issues HTTPS requests to the Core CAs Rest API.

Database: Database server is running MySQL. This server is dedicated only to the MySQL database in order to prevent *Single Point of Failures* and placed inside the internal network for *Minimum Exposure*. Access to the service is secured with password authentication and monitored with firewall access logs. Access to the service is possible from localhost and from Setup Server for maintenance. The Webserver is also granted read and write access for functionality.

Backup: To perform backups, every server's key is stored as authorized key in backup server. Servers store files mentioned in 1.2.6 to the backup server with rsync over SSH as a cron job. As the amount of data stored is small, full backups are performed every four hours.

Firewalls: A firewall has been placed between the internal network and the internet in order to block undesired connection attempts from the internet to the internal network. This firewall works as NAT that forwards packets between the network and segmenting the networks as DMZ and internal network.

Each host inside the DMZ and internal network is also configured with a host based firewall in order to prevent single point of failures and add an additional layer of protection.

Setup Server: The Setup Server is the component in charge of the configuration of the other internal components in the system through SSH. This server is accessible only by System administrators using SSH.

1.4.2 Applications

SSH: SSH is used and available on all the components of the system, however, its access is limited for security purposes as described previously. Moreover, IP based block for 1 hour is applied for more than 5 consecutive failed login attempts in each server that allows SSH connection.

Red Hat Ansible: We borrow ansible description from its documentation page: *"Ansible is an IT automation tool. It can configure systems, deploy software, and orchestrate more advanced IT tasks such as continuous deployments or zero downtime rolling updates."* [2]. It has been used for initial configuration and deployment of the system's machines and can be further used by system administrators for maintenance and configuration purposes.

[2]https://docs.ansible.com/ansible/latest/index.html

Webserver: The webserver provides the background functionality that serves the frontend interface with information and resources. It is setup as a Django REST api, that is connects through the gunicorn Web Server Gateway Interface (WSGI) to the Nginx webserver.

CA: Any cryptographic functions required in the logic of the Core CA are implemented in the pyOpenSSL library, a python wrapper around the OpenSSL library. The Rest API is implemented using the Flask framework and is served through the gunicorn Web Server Gateway Interface (WSGI) to the Nginx webserver.

SQL: The system uses mySQL [3] as DBMS in order to work with the legacy users table, providing the webserver with an interface to read and write this table's information in the default mysql port.

iptables: iptables [4] is utilized for both as network and host based firewalls. Since it is capable of being configured to forward packets, this software is also used as NAT to forward packets from the internal network to the internet and filtering the incoming packets from the internet.

Web interface: The web interface uses a MVVM Javascript framework (Vue.js) for a correct visualization and interaction of the information from a front-end point of view. It is server and communicates with the end-points exposed by the Django Webserver over HTTPS to give the user and CA administrators the desired functionality in a secure way.

1.4.3 Data Records

User Data User Data is stored in the database, which is a running legacy system and can not be replaced. The database stores the user ID, name, e-mail address and the hashed password of each user.

Certificates All the Client certificates are stored in plaintext in the filesystem of the Core CA.

Private Keys All the private keys are stored encrypted in the filesystem of the Core CA. The encryption mechanism used for that is explained in detail in Section 1.3.7. However, the private key of the intermediate certificate authority, that issues the user certificates, is stored in filesystem of the Core CA in plaintext,for *simplicity* and *usability*, as it is frequently used.

Logs The Logs of all system components are stored encrypted integrity protected and append-only at the Backup Server.

[3] https://www.mysql.com
[4] https://linux.die.net/man/8/iptables

2 Risk Analysis and Security Measures

In this section we define the assets, threat sources and risk definitions that will be later used to define and explain the risk evaluation of our system.

2.1 Assets

In this section we briefly describe the different assets of the system, including physical, logical, personal and intangible ones.

2.1.1 Physical Assets

System Infrastructure : Physical machines, one for each component in the system, are located within the company building in Switzerland, in a restricted area under lock. Each of the components can be individually compromised, which will cause the system to be available, partially-available or unavailable depending on the affected component importance and degree of operation. This point includes the internal network that accounts for the interconnection of the different components of the system.

Internet Connectivity: The internet connectivity is provided by a SP and is crucial for the correct operation of the system as it exposes the system to the exterior. This asset can be fully available, partially-available or unavailable depending on the speed of the connection, including a lower bound where the speed is so low that the system is declared as unavailable.

2.1.2 Logical Assets

Information

- **User Data:** Since the user data contains names and emails of informants, any leakage of its content could lead to their revelation. Furthermore, the user data contains password hashes, which could be used for password probing attacks when exposed. Thus it is of great importance to protect the user data with respect to confidentiality.
- **User Private Keys:** Users private keys are stored in the file system of the Core CA, and are used to secure their email communication. If a users private key gets compromised this allows an attacker to impersonate them or eavesdrop on their related email communication.
- **User Certificates:** User certificates are also stored in the file system of the Core CA, they can be used together with the user private key to log in to the corresponding user profile on the web application. Certificates are considered public and thus do not have to be protected with respect to confidentiality, however they are integrity protected by design through the signature of the certificate authority.

- **CA Private Keys:** We have two different kinds of CA private keys, namely the private key of the root CA and the private keys of the intermediate CAs. Any revelation of these keys allows an attacker to sign certificates in the name of iMovies and then impersonate any client or eavesdrop on the entire iMovies email correspondence.
- **SSH Private Keys:** The SSH private key of the Setup Server can be used to access and modify configurations in all components of the system.
- **Backups & Logs:** Backups are important when a recovery is needed. Also, there are valuable information in the logs that are stored in backup servers when an analysis is needed in case of a system anomaly.

Software

- **iMovies Web Application:** The web application REST api serving the frontend is implemented using the Django Rest Framework [5].
- **MySQL Database:** The databse is running MySQL [6], an open source database service.
- **Firewall Software:** For both network and host based firewalls, iptables [7] is configured.
- **CA module:** The CA is implemented using the pyOpenSSL [8] library which is a python wrapper around the widely established OpenSSL library. On top of the CA we have a REST api implemented using FLASK [9].
- **Webclient:** The Webclient is implemented using Vue.js [10], a JavaScript MVVM framework used to build user interfaces and Single Page Applications. The different pages access the exposed end-points by the Web Application to display information to the user and allow him to interact with it. This piece of software should perform the first validation to users' input to avoid possible exploits.

2.1.3 Persons

CA Administrators CA Administrators are highly important for the company as they have access to confidential information and the ability to manage the core components of the system. Modifications of the Certificate Revocation List, creation of false certificates or ex-filtration of users certificates are some of the potential capabilities of this kind of person, thus, making the a critical person in the system.

[5] https://www.django-rest-framework.org/
[6] https://www.mysql.com
[7] https://linux.die.net/man/8/iptables
[8] https://pypi.org/project/pyOpenSSL/
[9] https://flask.palletsprojects.com/en/1.1.x/
[10] https://vuejs.org

System Administrators System Administrators similarly to CA Administrators are critical to the company as have access to all the components in the system and are capable to modify the system or have means to access confidential information. Additionally, this kind of person know in detail the intrinsic of the system and its work is extremely important to keep the system up-to-date and protected against the newest known cyber-threats.

Normal employees These employees are the users of our system and will be the ones to potentially provide feedback about it, including possible improvements or disruptions of the system. Normal employees thus have an important role as the end users and would seek that the system fulfill all its needs in the easiest and fastest way possible.

External employees These kind of employees could include cleaning or maintenance of the iMovies work-space and are not direct users of the system, however they are potential threats to the correct operation of the system and could potentially be able to access confidential data.

2.1.4 Intangible Goods

Reputation and Confidence Reputation is extremely important from iMovies point of view, as it is considered a key asset for the company. Investigative reports require of informants that most probably want to remain anonymous, thus, any data leakage could lead to the disruption of more reports production. Additionally, the users of iMovies could stop using its services if its information is compromised, the trustfulness of the reports is in question or the service does not work smoothly.

2.2 Threat Sources

For iMovies we identified the following threat sources:

Nature: The system machines are located in Lausanne, Switzerland. This country has a low risk of natural disasters, for example strong hearth-quakes are highly rare. However, there still a non-negligible risk of flood, given its location near a lake and a riven, or electric storms that could cause disruptions in the systems correct functioning and availability.

Employees: This threat source includes normal iMovies employees using the system, administrators of the system, other staff working in the company including security, maintenance and cleaning. Employees in this group could be harmful by negligence or by action. For example, no proper training of system administrators is potentially risky. On the other hand, a government or organization being target of an investigative report by iMovies could pay an employee to actively harm the system. Additionally, discontent employees could also try to disrupt the system as a form of revenge.

Script Kiddies: Usually this kind of threat source doesn't make specific targets and uses mainly know vulnerabilities who could be found in the internet for its exploits. Not keeping a good maintenance could cause that our system becomes vulnerable to this kind of threat sources.

Skilled Hacker: This kind of threat source can usually design its own exploits which may include zero-day vulnerabilities for which no mitigation has been found. Similarly to "Employees", governments or organisations being target of an investigative report by iMovies may try to hire an "Skilled Hacker" to disrupt the system operation or leak confidential data in order to stop/disrupt the investigation.

Malware: This kind of threat source includes direct and indirect malware. The first type could be planted by a government agency to find information about a certain report. Both the direct and indirect type could be really harmful for our systems, an the possible damage of self-spreading malicious programs should be limited. A possible example of this, is this month's Ransomware attack to Software AG. [11].

Organized Crime or Governmental Agency: As described previously a criminal organization or governmental agency being investigated by iMovies could hire an skilled hacker or try to pay an organization's employee to disrupt the aforementioned investigation or try to discover the whist-blowers.

Competitors: A competitor agency may try to hire an skilled hacker to obtain confidential information of an investigation, to try to disrupt the early publication of an exclusive or to discover potential informers for its own economic benefit.

Hacktivism: A terrorist group could be incentivised to disrupt the system in case some investigative report targeted a sensitive affair, thus acting in retaliation. We could consider as an example, terrorism related to religious affairs a trigger for this kind of group to target iMovies after some religious related investigation.

2.3 Risk Definitions

In this section we define the likelihood and impact of a possible event and construct a risk level table following those definitions. This will be later used for the risk evaluation.

[11] https://www.infosecurity-magazine.com/news/software-ag-datastealing/

Likelihood	
Likelihood	Description
High	The threat source is highly motivated and has enough capabilities to exploit system's vulnerabilities to disrupt its supposed behaviour. Defenses are ineffective for this kind of source as the source exceeds the resources of the mitigation.
Medium	The threat source is motivated and has enough capabilities to exploit system's vulnerabilities to disrupt its supposed behaviour. Defenses may be effective against this kind of source and match its resources.
Low	The threat source is not motivated or lacks of enough capabilities to exploit system's vulnerabilities to disrupt its supposed behaviour. Defenses are effective against this kind of source as the mitigation exceeds the resources of the threat source.

Impact	
Impact	Description
High	The event may cause a highly cost loss of tangible assets, may seriously damage intangible assets or cause a serious damage in person assets.
Medium	The event may cause a moderate cost loss of tangible assets, may damage intangible assets or may cause a damage in person assets.
Low	The event may cause a minor cost loss of tangible assets or may slightly but noticeably damage intangible assets.

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

2.4 Risk Evaluation

In this sections we list the system's potential threats and the corresponding countermeasure(s) to palliate them.

2.4.1 *Evaluation Asset System Infrastructure*

No.	Threat	Countermeasure(s)	L	I	Risk
1	Nature: Hardware failure	Periodic backups and maintenance team available for emergencies 24/7	*Low*	*Medium*	*Low*
2	Nature: Major natural disaster	Periodic backups and fast recovery using Vagrant configuration	*Low*	*High*	*Low*
3	Employees and Governmental Agency: Physical disruption	Background check in the hiring process and security monitoring of the activities in restricted areas	*Medium*	*Medium*	*Medium*

2.4.2 *Evaluation Asset Internet Connectivity*

No.	Threat	Countermeasure(s)	L	I	Risk
4	Nature: Hardware failure	The contract bounds the ISP to compromise to provide connectivity 24/7	*Low*	*High*	*Low*
5	Nature: Major natural disaster	Include a wireless connection in case the main wired fiber option connection is disrupted	*Low*	*High*	*Low*

2.4.3 *Evaluation Asset User Data*

No.	Threat	Countermeasure(s)	L	I	Risk
6	Script Kiddie: Gain access to the database or log server	Keep the systems up-to-date and avoid direct connection to the machines through the internet	*Low*	*Medium*	*Low*
7	Malware: Ransomware, encrypt the database or log server data	Keep the systems up-to-date, avoid direct connection to the internet and train system administration to detect and avoid phishing and social hacking techniques	*Medium*	*High*	*Medium*
8	Governmental Agency, Hacktivism, Organized crime and Skilled hackers: Gain access to the database or the log server	Avoid direct connection to the machines through the internet and protect the Setup server access	*High*	*High*	*High*
9	Governmental Agency: Gain physical access to the database or the log machines	Maintain the systems under security monitoring in a restricted area	*Medium*	*High*	*Medium*

2.4.4 *Evaluation Asset User Private Keys*

No.	Threat	Countermeasure(s)	L	I	Risk
10	Employees: by action or omission allow a third party to gain access to its private keys which can be used to impersonate the user or decrypt its mails	Employees are trained and certificates can be revoked	*Medium*	*Medium*	*Medium*
11	Skilled hacker: target an employee to obtain its private key to obtain its confidential emails	Certificates can be revoked	*Low*	*Medium*	*Low*
12	Governmental Agency: brute-force the private key using large computational capacity	Limited time certificates validity and long and provably pseudo random key generation	*Low*	*High*	*Low*

2.4.5 *Evaluation Asset User Certificate*

No.	Threat	Countermeasure(s)	L	I	Risk
13	Employees: by omission allow a third party to gain access to its certificate which can be used to issue or revoke certificates in this employee's name or lose the certificate	Employees are trained and certificates can be revoked and reissued	*Medium*	*Low*	*Low*
14	Employees: usage of a certificate by an employee departing the company while the revocation is not active	Immediate revocation of the certificate upon departure	*Low*	*Medium*	*Low*
15	Skilled Hackers, Governmental Agency: Obtain an employee's credential or certificate to impersonate them	Only one active certificate per user, any new issued or revoked certificate is notified to the employee which can determine that an unauthorized access has happened	*High*	*Medium*	*Medium*

2.4.6 *Evaluation Asset CA Private Keys*

No.	Threat	Countermeasure(s)	L	I	Risk
16	Employees: Get access to the core CA or to the root private key to sign intermediate CA, or user certificates	The access to the core CA is limited to an small set of administrators and every access and action is logged, in the case of the root private key the set is even smaller and the access is physical	*Low*	*High*	*Low*
17	Organized Crime, Hacktivism, Governmental Agency and Skilled Hackers: Gain access to the core CA and compromise private keys in order to decipher company emails related to a certain investigation	The core CA is not directly accessible through the internet, the number of administrators with access is controlled and their actions logged, administrators pass a background check during hiring, root private key requires physical access and intermediate CA's revocation is possible without compromising user's confidence	*Medium*	*High*	*Medium*

2.4.7 *Evaluation Asset SSH Private Keys*

No.	Threat	Countermeasure(s)	L	I	Risk
18	Employees: an administrator by action or omission allow a third party to gain access to its private key	Administrators are trained and their actions are logged	*Medium*	*High*	*Medium*
19	Organized Crime, Hacktivism, Governmental Agency and Skilled Hackers: Compromise SSH keys by attacking the configuration server or targeting an administrator	Administrators pass a background check on hiring, the configuration server is up-to-date and possible inclusion of an IDS to alert in case of "abnormal" behavior	*High*	*High*	*High*

2.4.8 *Evaluation Asset Backups*

No.	Threat	Countermeasure(s)	L	I	Risk
20	Employees: Physical or logical access to the backup server to retrieve backup data or modify backup policies	Data encryption, periodic evaluation of backup policies, backup server limited physical (authorized personal) and logical (administrators) access	*Medium*	*Medium*	*Medium*
21	Skilled Hacker, Governmental Agency, Malware: Ransomware, encrypt the backup server data	Backup server not directly accessible through the internet, data backed up is also accessible in the running machines i.e. MySql Database	*Low*	*Medium*	*Low*

2.4.9 *Evaluation Asset Logs*

No.	Threat	Countermeasure(s)	L	I	Risk
22	Nature: Logs are lost due hardware failure	Regular logs backup	*Low*	*Low*	*Low*
23	Skilled Hacker, Governmental Agency: gain access and/or tamper the logs by compromising the log servers to retrieve information or cover an attack	Regular logs backup, non-sensitive information or appropriately protected, append only logs and log server not directly accessible through the internet	*Medium*	*Medium*	*Medium*

2.4.10 *Evaluation Asset iMovies Web Application*

No.	Threat	Countermeasure(s)	L	I	Risk
24	Script Kiddies: Try to gain access using well known vulnerabilities	Periodic Security Audits and Code Reviews	*Low*	*Medium*	*Low*
25	Skilled Hacker: Gain query or command injection with 0days	Periodic Security Audits and Code Reviews	*Low*	*High*	*Medium*
26	System Administrator: Introduce secret backdoors	Regular external audits and other system admins check for the rules.	*Low*	*High*	*Low*
27	Competitors, Governmental agency, Hacktivists, Crime organizations: Hire skilled hackers to hack the system	Periodic Security Audits and Code Reviews	*Medium*	*High*	*Medium*

2.4.11 *Evaluation Asset mySQL Database*

No.	Threat	Countermeasure(s)	L	I	Risk
28	Skilled Hacker,Governmental agency : Gain access to the database an steal information	The database is not directly accessible through the internet, keep the system up-to-date	*Low*	*Medium*	*Medium*
29	System Administrator: access the database and leak information	actions are logged, background checks during hiring process	*Low*	*High*	*Low*

2.4.12 *Evaluation Asset Firewall Software*

No.	Threat	Countermeasure(s)	L	I	Risk
30	Skilled Hacker: Bypass the firewall with 0 days or known vulnerabilities	Keep the firewall up to date and harden the firewall rules. Check for anomalies in access logs.	*Medium*	*Medium*	*Medium*
31	System Administrator: Introduce secret backdoors	Regular external audits and other system admins check for the rules.	*Low*	*High*	*Low*
32	Criminal Organizations: Bypass the firewall with 0 days or known vulnerabilities	Keep the firewall up to date and harden the firewall rules. Check for anomalies in access logs.	*Medium*	*Medium*	*Medium*

2.4.13 *Evaluation Asset CA module*

No.	Threat	Countermeasure(s)	L	I	Risk
33	Skilled Hacker,Governmental agency : Gain access to the CA core and obtain certificates to impersonate employees or decrypt confidential information	The CA module is not directly accessible through the internet, keep the system up-to-date	*Low*	*Medium*	*Medium*
34	System Administrator: access the CA core and leak information or modify/revoke/issue certificates	actions are logged, background checks during hiring process	*Low*	*High*	*Low*

2.4.14 *Evaluation Asset Webclient*

No.	Threat	Countermeasure(s)	L	I	Risk
35	Skilled Hacker, Script Kiddies: Bypass the input validation	Input validation is enforced in the backend	*High*	*Low*	*Low*

2.4.15 *Evaluation Asset Normal Employees*

No.	Threat	Countermeasure(s)	L	I	Risk
36	Nature: Harm a normal employee due to natural disaster or malfunction of electrical devices, low of manpower	Obey all the regulations related and perform simulacrums	*Low*	*Low*	*Low*
37	Gorvernmental Agency, Crime organization, Hacktivism: Extortion, kidnap or harm a normal employee, loss of manpower	iMovies facilities are controled by a security team	*Low*	*Low*	*Low*

2.4.16 *Evaluation Asset External Employees*

No.	Threat	Countermeasure(s)	L	I	Risk
38	Nature: Harm an external employee due to natural disaster or malfunction of electrical devices, low of manpower	Obey all the regulations related and perform simulacrums	*Low*	*Low*	*Low*

2.4.17 *Evaluation Asset CA administrator*

No.	Threat	Countermeasure(s)	L	I	Risk
39	Nature: Harm a CA administrator due to natural disaster or malfunction of electrical devices, lost of manpower	Obey all the regulations related and perform simulacrums	*Low*	*Low*	*Low*
40	Governmental Agency, Crime organization, Hacktivism: Extortion, kidnap or harm a CA administrator, leak of internal information	iMovies facilities are controlled by a security team	*Low*	*Low*	*Low*

2.4.18 *Evaluation Asset System administrator*

No.	Threat	Countermeasure(s)	L	I	Risk
41	Nature: Harm a System administrator due to natural disaster or malfunction of electrical devices, lost of knowledge	Maintain up-to-date documentation and maintain configuration scripts with the latest modifications, obey all the regulations related and perform simulacrums	*Low*	*Low*	*Low*
42	Governmental Agency, Crime organization, Hacktivism: Extortion, kidnap or harm a System administrator, complete compromise of the system	iMovies facilities are controlled by a security team	*Low*	*Low*	*Low*

2.4.19 *Evaluation Asset Reputation*

No.	Threat	Countermeasure(s)	L	I	Risk
43	Script Kiddies, Skilled Hackers: Compromise any component in the system, including obtaining user data	Maintain the system up-to-date, maintain a fast response team to mitigate the damage in case of attack	*Low*	*High*	*Low*
44	Skilled hacker, Governmental Agency: Obtain confidential information about an investigation	Network hardened, restrictive data access	*Low*	*High*	*Low*

2.4.20 Risk Acceptance

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

No. of threat	Proposed additional countermeasure including expected impact
3	Move the system infrastructure to a data center that can enforce higher security standards
7	Duplicate the system in different locations under different access control behind a load balancer to be able to revert the data and continue the operation
8	Use an Intrusion Detection System and migrate the database to store passwords in a secure way (i.e. salted)
9	Move the system infrastructure to a data center that can enforce higher security standards
10	Train employees regularly using social engineering attacks to raise their awareness
15	Migrate the database to store passwords securely and enable 2FA so stolen credentials can not be directly used.
17	Make the access to confidential key material subject to the presence of at least two administrators
18	Use an automatic system to check for "abnormal" behaviour and raise an alert in that case
19	Raise the awareness of the administrators through training and simulacrums, SSH keys must be periodically updated, move the Configuration Server inside the internal network and make that it is online accessible inside iMovies facilities or through a VPN
20	Enable disk encryption to avoid that the cloned data can be leaked
23	Monitor the log updates to check for "abnormal" behaviour i.e. non-consistent logs, not enough logs
25	Invest in third-party security audits and features to mitigate as much as possible possible vulnerabilities
27	Invest in third-party security audits and features to mitigate as much as possible possible unknown vulnerabilities, including maintaining hardware up-to-date (against hardware attacks)
28	Invest in third-party security audits and features to mitigate as much as possible possible vulnerabilities
30	Invest in third-party security audits and features to mitigate as much as possible possible vulnerabilities
32	Invest in third-party security audits and features to mitigate as much as possible possible unknown vulnerabilities
33	Invest in third-party security audits and features to mitigate as much as possible possible unknown vulnerabilities

References

[1] Michael Schläpfer David Basin, Patrick Schaller. Applied information security: A hands-on approach, 2011.

[2] OWASP. Authentication Cheat Sheet. `https://cheatsheetseries.owasp.org/cheatsheets/Authentication_Cheat_Sheet.html`.