System Description and Risk Analysis

Team 9: Kamila Babayeva Marin Cornuot Niels Lachat
Lin Zhang

January 3, 2024

Contents

1 System Characterization

1.1 System Overview

The company iMovies specializes in independent film production and primarily focuses on investigative reporting. It is crucial that all information exchanged within the company, as well as with informants, is managed with the utmost confidentiality. iMovies would like to take its first steps towards PKI-based services. Based on the project requirements, a Certificate Authority (CA) system is designed to manage and provide digital certificates to company employees, ensuring secure email communication and authentication within the organization's network. It also facilitates certificate revocation and administrative monitoring, all while integrating with a legacy database system.

An overall network diagram of the system with all the components is presented in Figure 1. Applying the principle of compartmentalization, the web server is placed in a DMZ (Demilitarized Zone), acting as a separate network segment from the internal, trusted network. The legacy MySQL database with user information, CA server for certificate management, and the backup server ensuring data continuity as well as logging are within the system's internal boundaries. External client systems, remote administration and maintenance access by the system admin and CA admin are allowed to connect with the intranet services from the internet. The functionalities of each component are elaborated as follows:

- Web Server: handles user interfaces, personal information/password update requests, certificate requests, certificate delivery, revocation requests.
- CA Server: manages user certificate, CA configuration, CA certificate and keys, functionality to issue new or revoke existing certificates.
- MySQL Database: stores the user information and the users' latest certificate.
- Backup Server: backs up keys, certificates, databases, configurations, and logs.
- Router/Firewall: is a simple machine which acts as a NAT for the internal network. It is used to separate the web server in the DMZ from the machines in the internal network. It also enforces some rules to protect the machines from various types of attacks.
- Client System: is a representative machine used by iMovies employees, system admin, or CA admin to access the CA services from the internet.

1.2 System Functionality

1.2.1 User authentication to the web server

There are two ways in which the user can authenticate to the web server:

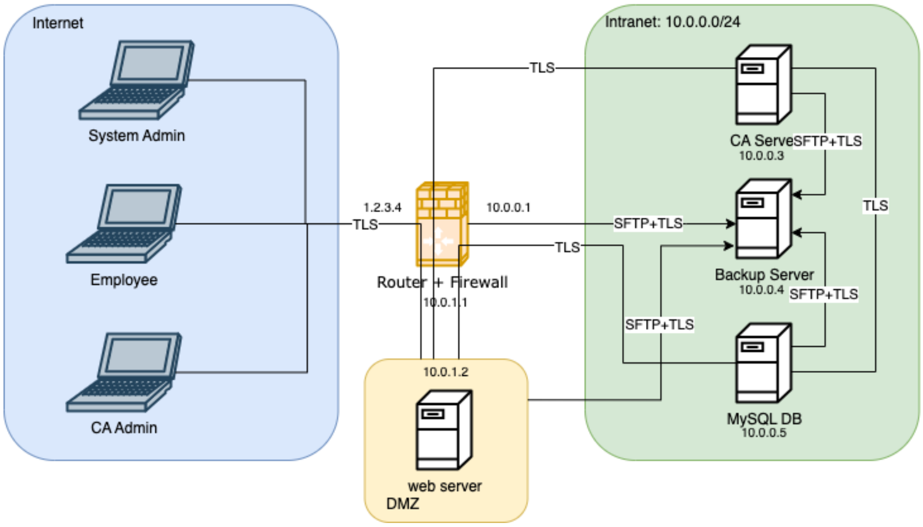

Figure 1: Network diagram of the system with all the components.

1. Password authentication: The user sends their username and password to the web server. The web server computes SHA256(password), requests the entry corresponding to username in the MySQL DB and compares the hashes. If they match, the user is authenticated.

2. Certificate authentication: upon establishment of the TLS connection to the web server, the client certificate is checked (must be signed by Certificate Authority and not be in the revoked list). If the authentication fails, the client is redirected to the login page where he can try to authenticate again with one of the two methods. If too many login attempts failed, he will be blocked for a given amount of time.

1.2.2 Certificate Issuing Process

Applying the principle of Simplicity, a user can only have one key-pair and one certificate valid at a time. It makes the certificate authentication system easier to analyze and review, and it is thus easier to establish the system's trustworthiness. The process of issuing a certificate is elaborated as follows:

1. The user authenticates to the web server.

2. The user can see and modify their information as well as view their currently valid certificate (if it has not been revoked) and download it. The user cannot view or download their key pair which were generated in previous sessions. If a user modifies their personal information, the modified information is sent to the MySQL DB and a new certificate and key pair can be issued.

3. To issue a new certificate: a request is sent to the CA server by the web server with the user's information. The CA server replies with the signed certificate and the generated key pair. The CA server also sends the freshly generated certificate, public key and it corresponding private key encrypted by the master backup public key to the backup server. Before generating this new certificate, all previously emitted certificate for the user are revoked.
4. The user can then download the freshly generated key pair and certificate in a PKCS12 file.
5. If a user loses their device, they have the possibility to connect using their username/password and issue a new certificate. This will automatically cause the old certificate to be revoked.

1.2.3 Certificate revocation process

1. The user authenticates to the web server.
2. The user clicks on the "Revoke Certificates" button. The web server then sends the uid of the user to the CA server for which the certificate should be revoked. Because the users can only have one valid certificate at a given time, there is no ambiguity about which certificate to revoke.
3. The CA server adds the certificate to the certificate revocation list (CRL), this list can then be downloaded by the user.
4. The user gets a confirmation that his certificate has been revoked and will not be able to connect to the revoked certificate anymore.

1.2.4 CA Administrator interface

1. The CA administrator connects via HTTPS to the web server and authenticates using TLS client authentication when accessing the CA Administrator Interface.
2. The interface shows the following information: number of issued certificates, number of revoked certificates, current serial number.
3. In case of the CA Administrator client certificate loss, system administrator would verify CA Administrator authenticity in a reliable alternative approach, e.g. video call, and issue a new certificate and distribute it physically to CA Administrator.

1.2.5 System administration and maintenance

Remote administration by the system administrator is enabled via SSH. The system administrator has an SSH private key and uses it to authenticate to all the machines of the intranet and the DMZ. The private key is encrypted on

the system administrator machine with a password which needs to be entered before new ssh connections.
The SSH communications are done by SSH tunneling through different ports of the firewall since it is the only machine accessible from the internet.
When a system administrator needs to utilize root privileges on remote machines, they are required to enter their unique password for each machine. This practice adheres to the security principle of maximizing the entropy of secrets, ensuring that each password is distinct and complex. By combining this with the use of a private key, it establishes a multi-factor authentication process. This approach effectively implements the security principle of avoiding a single point of failure, as a private key and two passwords are necessary for sudo access on remote machines, enhancing overall system security.

1.2.6 Logging

To enable traceability of the system, events and actions are logged in the corresponding host machines and backed up to the backup server. All the logs are sent to the backup server automatically in real time and the configuration files are backed up daily using a cron job on each host machine. The integrity of the logging in transit is implemented by secure communication (TLS) to prevent log alteration and log deletion. The following log categories are backed up in the backup server to attain the security principle of complete mediation:

- system-wide logging in user space: such as logins and other user activities related to processes or services, on all Intranet (e.g. Apache2 access and error logs).
- system-wide logging in kernel space: related to errors, warning or information entries that the system kernel writes, on web server, CA server, backup server, MySQL server, router+firewall instance.
- program specific log files: MySQL database, configuration files.

1.2.7 Backup

Client agent backup approach is adopted for backing up files in the below list from the client agents on the intranet. The reasons of choosing a 'push' method instead of a 'pull' method are (1) to reduce the risk of a single point of failure by granting the backup user with ssh accesses to all client agents; (2) to minimize the workload on the backup server, where the client agents process data and sends it to the backup server; and (3) to allow client agents to choose the backup files with its corresponding backup trigger mechanism.

- Key & certificate backup: all the key pairs issued by the CA server are stored in the backup server, encrypted with the public master backup key. Once a new certificate is issued, it is directly backed up to the backup server. The entirety of the backup server is itself backed up, once a week, by the system administrator on external, offline drives to provide redundancy of the backup. These backup drives are stored in a physically secure vault.

- MySQL database backup: a logical backup of the *imovies* database is scheduled and sent with a daily cron job on MySQL server to the backup server. Older versions of the logical database backups would be rotated and migrated in an external, offline drives until the lawful retention period.

- Logs backup: all logs in Section 1.2.6 are real-time backed up in the backup server.

- System configuration: code as infrastructure and system re-configuration file backups are stored in the backup server by a daily cron job. Only former day configuration files are stored in the backup server. The former copies of the configuration files are migrated offline until compliance retention period.

To adhere to the security principle of least privilege and avoid confusion with the Debian system's reserved backup user and the directories owned by the root user, a separate, unprivileged user named *backupusr* is created. This user is specifically designated for client agent backups, which are carried out by setting up data dumps through SFTP.

1.3 Security Design

The security design provides a comprehensive framework that encompasses various critical components to ensure the integrity, confidentiality, and availability of data and resources. This design is articulated through the following components safeguarding the system against a wide range of threats and vulnerabilities, thereby maintaining a robust security posture.

Cryptographic keys

- Employee: key pair and a certificate issued by the CA server. Employee private key is stored on the employee's machine, and is also in the Backup server, encrypted with the master public backup key.

- Web server: key pair and certificate issued by the CA server used for TLS communications with all other intranet machines to process employee requests. Access to the private key is restricted. The webserver also has a keypair and certificate to serve the website frontend over HTTPS, which is also issued by the CA. The webserver also has an ssh key pair to do backups over sftp to the backup server. It also has a key pair and certificate for TLS client authentication to the MySQL server.

- System Admin: key pair for SSH key-based authentication. Password is required for the root privilege access in the machines. Passwords should be stored using a password manager on the system admin machine. The private key is stored encrypted on the system admin machine.

- CA Admin: key pair and a certificate issued by the CA server used for TLS client authentication to the CA admin interface (via HTTPS) hosted on the web server. It is password protected and stored on the CA-admin machine. (on client-gui)
- System Backup: Master backup private key stored offline (in physically secure location, accessible only by system admin in case of emergency), public key used to encrypt the pkcs12 certificate and private key bundles sent to backup by CA-server.
- MySQL server: key pair and a certificate issued by the CA server used for TLS server authentication during the establishment of MySQL connections. The MySQL server also has an ssh key pair to do backups over sftp.
- Backup server: key pair and a certificate issued by the CA server used for TLS server authentication during the establishment of logging connections.
- CA server: root private key stored and used in the CA root to issue new certificate. The CA server also has key pair and certificate for HTTPS server authentication to serve the CA admin web application requests. The CA server also has an ssh key pair to do backups over sftp. It also has a key pair and certificate for TLS client authentication to the MySQL server.

Session management On the HTTPS connections to the web server, upon successful login, users' connection are managed by the flask application through session cookies. Upon logout, the session fields are "poped" and the cookies can no longer be used to access the webserver services until re authentication (through uid/password or certificate).

The SSH sessions by the System administrator to all intranet machines and the SFTP sessions by backing up client agents to the backup server are configured in the $sshd_config$ file. For all SSH connections, password authentication is disabled and public key authentication is enabled with whitelisted allowed users. Once authenticated, SSH negotiates session keys through a process known as key exchange, ensuring that each session has its unique encryption, thus maintaining confidentiality and integrity. SSH also incorporates mechanisms like session timeouts and rekeying. The former can be set by updating the values for $ClientAliveInterval$, $ClientAliveCountMax$, and $TCPKeepAlive$ in the $sshd_config$ file, whereas the latter being a process where new keys are generated at regular intervals within a session, significantly reducing the risk of cryptographic attacks. Moreover, SSH's resistance to network-level attacks such as IP source routing and DNS spoofing makes it a formidable tool for secure communications.

DOS/bruteforce protection At the network layer, firewall uses fail2ban to monitor logs for suspicious activity and automatically blocking IP addresses that repeatedly exhibit malicious behavior, such as multiple failed login attempts. As an extra layer of security, at the application layer, the web server enforces a temporary block on users who performs repeated failed authentication attempts (max 10 per minute) by using Flask Limiter on the /login endpoint of the webserver.

Data in transit

- User - Webserver: data is securely transmitted over an HTTPS (TLS 1.3) connection, ensuring the authenticity, confidentiality and integrity of information.
- Webserver/CA server - MySQL Database: data is securely transmitted over the MySQL protocol using TLS 1.3, ensuring the authenticity, confidentiality and integrity of information through client and server authentication.
- Webserver - CA server: data is securely transmitted over an HTTPS (TLS 1.3) connection, ensuring the confidentiality and integrity of information.
- Webserver/CA server/MySQL - Backup Server: For logging service, data is securely transmitted over the BSD protocol using TLS 1.3, ensuring the confidentiality and integrity of information through server authentication. SSH is used for backup service (using SFTP) with public key authentication, ensuring the authenticity, confidentiality and integrity of data transmission.
- System administrator (from the internet) - intranet machines: SSH is used with public key authentication, ensuring data transmission's authenticity, confidentiality and integrity. Password authentication through SSH is disabled for enhanced security.

Data at rest

- MySQL database: The database is stored in plain text with SHA-256 hashed employee passwords. Only webserver and CA server database users have specific access to tables (ensuring the Least Privilege principle). Authentication for web server and CA server users require certificates and passwords (ensuring No Single Point of Failure through multi-factor authentication). The root database user is only enabled on localhost, accessible solely to the system admin after a successful SSH to MySQL server. Password authentication to the root database user is mandatory.
- Backup server: Logs, MySQL database backups, and configuration files are stored in plain text. Private employee keys are encrypted using a master backup private key, which is stored offline (ensuring Minimum Exposure). Access to this machine is restricted to the system admin via SSH.

Webserver hardening To reduce the attack surface, minimize trust and maximize trustworthiness, the following system hardening measures are implemented:

- Strict Transport Security HSTS is setup to protect the webserver against man-in-the-middle attacks by ensuring that web browsers communicate with the server over secure, encrypted connections (HTTPS) only. HTTP request are not accepted and automatically redirected to HTTPS on port 443.

- Regular expressions (regex rules) are employed to enforce constraints on user input for first and last names, permitting only English letters in lowercase or uppercase, numbers, and spaces, thereby mitigating the risk of code injection attacks. Additionally, email inputs undergo validation against regex rules to ensure coherence of user-provided data. Also the size of user input is limited in size.

- Utilization of prepared statements in SQL queries ensures that user inputs are treated as data rather than executable code, thereby mitigating the risk of SQL injection and cross-site scripting (XSS) attacks.

- Secure, fail safe defaults configuration.

- Each action is meticulously logged at a granular level, stratified by varying degrees of criticality (info, error, critical), and automatically transmitted to the logging server.

- Flask form intrinsically enforces protection against Cross-Site Request Forgery (CSRF), ensuring a robust mitigation of CSRF vulnerabilities within the framework.

- Flask sessions bolster security through the generation and management of cryptographically signed session cookies, mitigating the risks of session-based attacks and ensuring the integrity and confidentiality of user data within the web application.

- Utilises host-based firewall rules to add an extra layer of security beyond the primary firewall. The policy is set to deny both incoming and outgoing traffic by default, allowing only essential services through a carefully curated whitelist. Additionally, ping responses from the web server are disabled as a precautionary measure in case of a compromise, enhancing overall security.

- The web application is run by a low privileged user, webserver which has no sudo right. This mitigates potential hazards following an eventual successful security compromise.

CA-server hardening

- Validate inputs used to build distinguished names for certificates (prevents injection of illegal fields or data into signed certificates). This is already enforced by the webserver but avoids having a possible single point of failure.
- SSL authentication is enforced for communication between webserver and ca-server.
- The ca-server user has sudo access only to run the openssl command in order to sign certificates and emit new certificate revocation lists. This enforces the principle of Least Privilege.

1.4 Components

We will only use IPv4 for simplicity and the IP addresses are all configured as static.

1.4.1 MySQL server

OS: Debian 12.0 with no GUI
Intranet IP: 10.0.0.5
Open ports: 22 - ssh, 3306 - mysql
System users: *sysadmin* with root privileges, *backupusr*

Service MySQL MariaDB is used to run MySQL server on port 3306.

- Database: *imovies* database, which includes two tables: $users$ and $certificates$. The $users$ table contains the following fields: *uid*, *lastname*, $firstname$, *email*, and *password*. The $certificates$ table includes *uid* and the current valid $certificate$ if any.
- Database users and privileges:
  - root - System admin - the access is only from the localhost after the system administrator ssh to the server. The remote access for a root user is disabled. Root password is required for the access.
  - webserver - Web server - read/write access to the table $users$; read access to the table $certificates$. X509 certificate and password is required for the access.
  - caserver - CA server - read/write access to the table $certificate$. X509 certificate and password is required for the access.

Service SSH SSH listens on a default port 22. Password authentication is disabled, only public key authentication is allowed. Public key of allowed user *sysadmin* is stored in the corresponding home directory.

Service Backup Cron job initiates a daily logical backup of *imovies* database and configuration of MySQL server to the backup server using SFTP (dest ip: 10.0.0.4, dest port: 22).

Service Logging System-wide and program-specific logs are streamed live to the backup server using the syslog-ng service (dest ip: 10.0.0.4, dest port: 6514). The connection uses TLS, server authentication is performed.

1.4.2 Backup server

OS: Debian 12.0 with no GUI
Intranet IP: 10.0.0.4
Open ports: 22 - ssh, 6514 - syslog-ng
System users: *sysadmin* with root privileges, *backupusr*, *router*, *caserver*, *webserver*, *mysql*

Service SSH SSH listens on a default port 22. Password authentication is disabled, only public key authentication is allowed. Public keys of allowed users *sysadmin*, *router*, *caserver*, *webserver* and *mysql* are stored in the corresponding home directories.

Service Logging The syslog-ng service is configured to listen on port 6514, collecting streamed logs from the intranet machines. The connection uses TLS. A daily cron job triggers a script responsible for log rotation, logs are removed on a bi-weekly basis.

1.4.3 Web server

OS: Debian 12.0 with no GUI
DMZ IP: 10.0.1.2
Open ports: 22 - ssh, 80/443 - Apache2/Flask
System users: *sysadmin* with root privileges, *webserver*, *backupusr*

Service Apache2 The server listens on port 443 for HTTPS connection to the iMovies web interface. HSTS used, all connection attempts to port 80 (HTTP) are automatically redirected to port 443. A low privileged user is running the web application. No sensitive data is stored on the webserver machine. Authentication, data modification, and certificate issuance/revocation, all involve queries to the CA server and MySQL server. Sensitive information, such as certificates, are temporarily stored in secure files, promptly deleted upon client download. All intranet communications are encrypted and authenticated using TLS.

Service SSH SSH listens on a default port 22. Password authentication is disabled, only public key authentication is allowed. Public key of allowed user *sysadmin* is stored in the corresponding home directory.

Service Logging System-wide and program-specific logs are streamed live to the backup server using the syslog-ng service (dest ip: 10.0.0.4, dest port: 6514). The connection uses TLS, server authentication is performed.

Service Backup Cron job initiates a daily backup of configuration files and web app source code to the backup server using SFTP (dest ip: 10.0.0.4, dest port: 22). The low priviledged user "backupusr" is responsible for the collection and transmission of backup files.

1.4.4 CA server

OS: Debian 12.0 with no GUI
Intranet IP: 10.0.0.3
Open ports: 22 - ssh, 443 - Apache2 + Flask
System users: *sysadmin* with root privileges, $ca - server$ on which the Flask server code is running. This user has access sudo access only to run openssl to sign certificates upon request, revoke certificates, and emit new certificate revocation lists.

Service Apache2 The Apache server handles incoming requests and forwards them to the Flask application using WSGI.

Service SSH SSH listens on a default port 22. Password authentication is disabled, only public key authentication is allowed. Public key of allowed user *sysadmin* is stored in the corresponding home directories.

Service Logging System-wide and program-specific logs are streamed live to the backup server using the syslog-ng service (dest ip: 10.0.0.4, dest port: 6514). The connection uses TLS, server authentication is performed.

Service Backup Cron job initiates a daily logical backup of server configurations to the backup server using SFTP (dest ip: 10.0.0.4, dest port: 22).

1.4.5 Router/Firewall

OS: Debian 12.0 with no GUI
External IP: 1.2.3.4
DMZ IP: 10.0.1.1
Intranet IP: 10.0.0.1
Open ports: 22 - ssh, 443 - https, 2002, 2003, 2004 and 2005 - ssh for internal machines
System users: *sysadmin* with root privileges, *backupusr*

iptables for port forwarding Requests on tcp ports 443, 2002, 2003, 2004 and 2005 are forwarded to the corresponding machine using a prerouting rule to change the destination. The destinations are respectively: 10.0.1.2:443 (webserver, https), 10.0.1.2:22 (webserver, ssh), 10.0.0.3:22 (ca-server, ssh), 10.0.0.4:22 (backup-server, ssh), 10.0.0.5:22 (mysql-server, ssh)

iptables for firewall Requests which don't match the forwarding rules are dropped by default during forwarding. This enforces a secure fail-safe defaults approach by implementing a whitelist. Any unanticipated traffic will be dropped, which means that even if something is terribly misconfigured on the intranet (e.g. running telnet server), adversaries from the internet will not be able to even reach the machine on that port.

Other rules enforce separation between the DMZ and the intranet (in accordance with the principle of Compartmentalization). These rules include only:

- Allowing connection establishment from webserver to ca-server on port 443 to perform ca requests.
- Allowing connection establishment from webserver to backup server on port 22 for backup purposes.
- Allowing connection establishment from webserver to backup server on port 6514 for logging purposes.
- Allowing connection establishment from webserver to mysql server on port 3306 for authentication and information retrieval purposes.
- Allowing all established connections to send packets back to the initiator.

All other traffic going through the firewall is dropped (Fail-safe defaults).

Service SSH SSH listens on a default port 22. Password authentication is disabled, only public key authentication is allowed. Public keys of allowed users *sysadmin* are stored in the corresponding home directories.

1.4.6 User System

OS: Lubuntu 20.04 LTS with GUI
External IP: 1.2.3.42/24
System users: *sysadmin*, *vagrant*

Web browser This machine falls outside the project's scope. We provide a client GUI VM for connecting to the web server and accessing the iMovies service. Additionally, it enables the system admin user to connect via SSH to the firewall and machines within the intranet/DMZ for administrative tasks.

1.5 Backdoors

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

1.5.1 Easy Backdoor

This backdoor involves concealing SSH credentials for an internal machine within the metadata of the logo image on our website's login page as presented in Figure 2. Our proposed exploitation process includes the following steps:

1. First, by looking at the source code of the /login page one can find a hidden comment "Look further than you can see". And indeed, the "imoviies" logo should be examined to discover some interesting stuff.

2. Find hidden data in the logo image: Examine the image headers, accessible through local image viewers or online tools (e.g. Steganography decoder) to read hidden information. The hidden data is encoded in base64. It is also possible to use *exiftool* on Linux to extract the USERCOMMENT field which is contained in the metadata of the image.

3. Decoding the base64 comment reveals some credentials for a debug account on a machine with it's password.

4. The attacker can now do a quick port scan to discover some open ports on the firewall which are used for SSH into the intranet machine. Onyl one of these machines accept password for SSH (while all others only accept private/public key). The final SSH command is as follows: `ssh -p 2004 debug@1.2.3.4` , password: `Congratulations!Y0uF0undTh3Ea5y8ackd0or:+1:`. Once you're connected inside the LAN, the fun begins and quite a few things are possible, this is at the discretion of the attacker.

Figure 2: iMovies logo image on the web application

1.5.2 Advanced Backdoor

The secret password for the Flask application running on the web server is set to a weak, commonly used string:

```
app = Flask(__name__)
app.secret_key = ”secret”
```

This very weak password can easily be brute forced by an attacker to attack session cookies. Flask uses its secret key to sign session cookies which are stored on the client side. This key must remain secret to make sure that the session cookies can't be tampered with.
Discovering this secret key allows a user to forge session cookies which contains critical information like the user-id, personal information such as email and name. Manipulating those cookies could allow an attacker to impersonate someone else, but could also be used to bypass some restrictions like requesting a new certificate more often than once a minute. (since last certificate request is also stored in session)
Upon seeing a legitimate cookie signature, the web server grants access to the corresponding user if the session has not be revoked. It is thus critical to keep Flask's key secret and to generate a random string sequence as password.

In our case, this could allow the attacker to impersonate any user, except for the ca-admin which is protected by a certificate authentication only, not by session ids. This could allow the attacker to revoke or generate other users certificates and modify there personal information. However, password modification for other users remains improbable, given that passwords are not stored in sessions and need to be provided upon modification. CSRF attacks might also be possible since tokens are protected through Flask's secret key.

In the blackbox review, it is quite hard to find this vulnerability even though the key used is very basic, it should be much easier to find upon inspection of the source code in the whitebox review. There's many information on how this type of backdoor can exploited to do bad stuff on the internet, like here.

2 Risk Analysis

2.1 Assets

2.1.1 Physical assets

- Physical safe: Holds the master private key necessary for decrypting the private keys and certificates of employees from the backup server in the event of loss. The key safe is entirely offline and requires the System Admin or CA admin to physically open it. The key also grants access to the backups of the machines and database. All CIA (Confidentiality, Integrity, Availability) properties are crucial and applicable here. The key protects sensitive information, needs to be quickly available in case of loss, and must not be modified. Desirable state space includes system admin and CA admin.

- Physical Backup Storage: Regularly transfers backup data and logs from the backup server to external physical storage. This practice helps maintain free space on the backup server and ensures the availability of data. Integrity is crucial to prevent data corruption. Desirable state space includes system admin.
- Firewall/router: This is the central access point linking our system to both the Internet and the Intranet. Availability is a crucial property to maintain the connection. Desirable state space includes system admin.
- Web server: Serving as the primary interface, the web server is the sole connection between users and our systems. Desired properties include integrity to serve correct data to users and availability to our system.
- Intranet machines: This network of essential services operates internally, not directly connected to the Internet. Communication from the Internet to the intranet is facilitated through the firewall and DMZ. Availability is required for system accessibility. Confidentiality is crucial as these components produce and store sensitive information. Integrity is also important as the data is later served to users.
- Internet Connectivity: connection of the internal network to the rest of the internet, provided by an ISP. Availability is crucial for the system to work properly.

2.1.2 Logical assets: Software

- Firewall/Router: Runs on a Debian machine, managed by a system admin. The system admin keeps the firewall software up-to-date and manages the configuration/iptables whenever needed.
- Web server: Configured and regularly maintained by a system admin on a Debian machine. Utilizes Apache or Nginx for web content, runs a MySQL client for intranet connections, and employs SSH service for system admin access.
- Intranet devices:

  - MySQL server: Runs on a Debian machine, runs MySQL server and employs SSH service for system admin access.

  - Backup server: Runs on a Debian machine, collects logs from all the machines using SFTP and employs SSH service for system admin access.

  Confidentiality and integrity are ensured by encrypting logs with a master public key. The backup is stored offline providing availability in the event of backup server failure.

- CA server: Runs on Debian machine, hosts a Certificate Authority and employs SSH service for system admin access.

2.1.3 Logical assets: Information

- Employees data: Includes employees' personal information, credentials (username/password), and certificates.
- Employees private keys: Utilized to sign employees' messages, verifying their identity. Desirable state space includes employees (key owners) and the system admin.
- CA private key: Used to sign certificates. Desirable state space includes system admin (pending confirmation) and CA admin.
- Backup private key: Stored offline in a safe. Desirable state space includes system admin and CA admin.
- Private Keys for Intranet Communication: Since most of the services rely on TLS, the corresponding private keys should be kept confidential and accessible only for the designated servers.

2.1.4 Persons

- System Admin: responsible for maintaining the complete server infrastructure, including firewalls, web servers, databases, and more. System Admins have access to all critical data on the system.
- CA Admin: monitor the functionality of the CA server by checking the current state of requested/revoked certificates and serial numbers. CA Administrators have access only to the CA server.
- Employees: main users of our system. They have control only over their personal data and have the capability to request or revoke certificates based on their needs.

2.2 Threat Sources

- Nature: Fire, flooding, or earthquakes can be threat sources for the physical assets and persons of the company.
- Employees: Regular employees who hold some grudge against the company or who do not follow security best practices. The CA administrator or system administrator could be extorted or bought by actors wishing to impede some investigation of an iMovies employee.
- Script Kiddies: Since we have a web server exposed to the internet, script kiddies might want to shut it down for personal glory or chaos.

- Skilled Hackers: If iMovies is sufficiently well-known, skilled hackers might want to threaten some aspect of our system for fame or money (e.g., selling unreleased movies to the dark net).

- Competitors: Competing movie companies might want to have a competitive advantage over iMovies.

- Organized Crime: Criminals can be motivated to prevent the publication of movies investigating their organizations.

- Governmental Agencies: Similar to organized crime, these agencies might want to prevent the publication of investigations about state secrets or unpopular activities.

- Malware: Undirected malware can threaten our system, as it is connected to the Internet. The primary objective of undirected malware may involve controlling a large number of computers to establish a botnet.

2.3 Risks Definitions

Likelihood	
Likelihood	Description
High	The threat source is powerful and is highly motivated to attack our system using a vulnerability. Moreover, there are no effective counter-measures implemented to prevent exploitation of said vulnerability.
Medium	The threat source is motivated to attack our system, but there are effective counter-measures in place which make exploitation of vulnerabilities more difficult.
Low	The threat source lacks the capability to exploit a vulnerability, or there are very effective counter-measures in place which make the exploitation very difficult.

Impact	
Impact	Description
High	The event causes important and long-term (months to years) damage to the company or its employees, and will be very costly recover from.
Medium	The event causes damage to the company's assets, but the situation can be recovered from in at most a few weeks at a moderate cost to the company.
Low	The event causes some hindrance to the company's core function but can be recovered from in a matter of hours or days.

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

2.4 Risk Evaluation

2.4.1 *Physical Assets: Key safe*

No.	Threat	Countermeasure(s)	L	I	Risk
1	The key safe gets destroyed by a natural event.	Securely distributing and maintaining multiple safes in various locations.	*Low*	*High*	*Low*
2	The CA admin left the role without knowledge transfer, thus losing the combination of the safe.	The system administrator also knows the combination.	*Low*	*High*	*Low*

2.4.2 *Physical Assets: Backup storage*

No.	Threat	Countermeasure(s)	L	I	Risk
3	Physical access to a backup storage by employees or governmental agencies to steal/modify data.	Strong physical security, strong encryption algorithms to enforce confidentiality and integrity of data. Several copies of the backup storage.	*Low*	*Medium*	*Low*

2.4.3 *Physical Assets: Backup server*

No.	Threat	Countermeasure(s)	L	I	Risk
4	Physical access to steal/modify data on a backup server by employees or governmental agencies.	Strong physical security and access control, strong encryption algorithm to encrypt sensitive data with a encryption key stored in a separate secure device.	*Low*	*Medium*	*Low*

2.4.4 *Physical Assets: Internet access*

No.	Threat	Countermeasure(s)	L	I	Risk
5	A natural event breaks Internet connectivity to the intranet. This makes the system unavailable to employees and administrators.	No countermeasure but unlikely, the server would be located in a secured facility in switzerland where natural disasters are very rare.	*Low*	*Medium*	*Low*
6	A government agency blocks internet access to prevent one of iMovies investigations from being published.	Certificates can still be generated and transmitted out of band (e.g. by the CA admin, on a USB stick)	*Medium*	*Medium*	*Medium*

2.4.5 *Logical Assets: Software on the Web server*

No.	Threat	Countermeasure(s)	L	I	Risk
7	A Script Kiddie uses a known vulnerability to crash the web server.	Keep the software up to date so that known vulnerabilities cannot be exploited.	*Low*	*Medium*	*Low*
8	A criminal organization/-competitor hires a skilled hacker to find a zero-day vulnerability in the web server software which gives the hacker root access. From there the hacker tries to perform lateral movement in the intranet.	The firewall blocks any requests which are not in the accepted protocols. Security measures on individual intranet machines (e.g. strict access control, least privilege access) is implemented + efficient logging to detect and trace back these attacks	*High*	*High*	*High*
9	A competitors/skilled hackers/ Governmental Agencies attempt to brute-force employees' credentials on the login page.	Strong randomly generated password are enforced, delay mechanism for continuous login attempts is implemented, firewall rules temporarily blocks IPs that send suspicious or too many requests.	*Low*	*Medium*	*Low*
10	A Script Kiddie/Skilled hacker attempts to exploit vulnerabilities in a website's login page to retrieve sensitive data from a MySQL database.	Best practices like prepared statements and input validation, are implemented to prevent SQL injections.	*Low*	*Medium*	*Low*
11	A skilled hacker/organized crime may impersonate a website on a similar domain, tricking users into sending their credentials to the fraudulent site. Phishing attacks may be used to distribute the link to the impersonated website.	Educate employees about these attacks, encourage bookmarking the website, implement corporate mail filtering to prevent phishing attempts.	*High*	*Medium*	*Medium*

2.4.6 *Logical Assets: CA server software*

No.	Threat	Countermeasure(s)	L	I	Risk
12	An employee attempts to perform a DOS attack by continuously generating and revoking certificates.	Webserver enforces a delay mechanism that allows certificate generation or revocation once per a specified time window.	*Low*	*Medium*	*Low*

2.4.7 *Logical Assets: Backup server software*

No.	Threat	Countermeasure(s)	L	I	Risk
13	A Script Kiddie may attempt to execute numerous requests on the website, leading to a substantial increase in logs on both the web server and backup server.	Log rotation is implemented and log size limits is set.	*Low*	Low	*Low*

2.4.8 *Logical Assets: Software on Firewall/Router*

No.	Threat	Countermeasure(s)	L	I	Risk
14	Nature/System admins cause system reboots and loss of configurations.	Periodic backup of the configurations is implemented. This backup can later be used for automatic recovery.	*Low*	*Medium*	*Low*
15	Skilled hackers, Organized Crime bypass/modify firewall rules to gain access to internal devices.	Keep the firewall software up-to-date, log changes/access and access control mechanisms.	*Low*	*Medium*	*Low*
16	Skilled hackers, Organized Crime carry out DOS/DDOS attack to bring the system down or brute-force attack to gain unlawful accesses to any services behind the firewall.	Firewall blocks the IP address from which too many requests have been sent within a certain time window.	*High*	*Medium*	*Medium*

2.4.9 *Logical assets: Employees' data*

No.	Threat	Countermeasure(s)	L	I	Risk
17	A non-targeted ransomware encrypts the data of the backup server.	No direct connection from Internet to the backup machine, offline backups of the server are done, which allows us to quickly recover from this attack.	*Low*	*Medium*	*Low*
18	A skilled hacker/Organized Crime/Governmental Agencies gain access to mysql and/or backup machines.	No direct connection from the Internet to these machines, multi-factor authentication is implemented before one can steal data (private key for SSH, password for root privileges, password for mysql access).	*Low*	*High*	*Low*

2.4.10 *Logical assets: CA private key*

No.	Threat	Countermeasure(s)	L	I	Risk
19	Unauthorized access to the CA private key by a skilled hacker/Organized Crime/Governmental Agencies with malicious intent to create fraudulent certificates and access companies resources.	Access to the CA server and the private key is restricted to system and CA admins only, any access to these objects is logged.	*Low*	*High*	*Low*

2.4.11 *Persons: Employee*

No.	Threat	Countermeasure(s)	L	I	Risk
20	Phishing attack via email/SMS on employees by competitors/skilled hackers/Governmental Agencies to trick users to reveal their credentials to access the resources of the server.	Educate employees about phishing attacks, corporate mail filtering should be implemented.	*High*	*Medium*	*Medium*

2.4.12 Detailed Description of Selected Countermeasures

- Prepared SQL statements are used by the web server for the user name and password authentication to avoid SQL injection and XSS attacks.
- Network level DOS attack countermeasure is enforced by Firewall detecting intensive requests to the internal services and banning the IP address if too many requests from the IP address have sent within a certain time window. However, the countermeasure would be circumvented by DDOS attacks with higher request intensity from several different IPs.
- Application level DOS/brute-force attack countermeasure is done by the web application enforcing the frequency of requesting new certificates as well as sending requests to the internal services via the web application.
- Length and allowed characters of first name, last name and email are enforced from the web application to countermeasure SQL injection and XSS attacks.

2.4.13 Risk Acceptance

No. of threat	Proposed additional countermeasure including expected impact
6	If specific rules are deployed by the adversary to block some of our internet traffic, we could try to circumvent censorship by using the TOR network. The quality of service, speed and customer reach would be affected but it would allow to maintain some activity nonetheless.
8	Log monitoring tools can be setup to actively notify system admin about malicious/abnormal behaviour in the systems. With such countermeasure, the likelihood of the attack drops to Medium, the impact remains High. This countermeasure might only notify the system admin about possible malicious behaviour but does not prevent.
11	Two-factor authentication can be implemented with a time-based one-time password as the second factor, preventing attackers from successfully authenticating even if they acquire the initial credentials. Various detection tools for evil twin websites can be incorporated (e.g. Should I click?). With such countermeasures, the likelihood of the attack drops to Low and the impact remains Medium.
16	A content delivery network (CDN) combined with IPs blacklisting tactic can efficiently distribute the firewall load and blacklist IPs performing DDOS attack. With such countermeasures, the likelihood of the attack drops to Medium, because the attack can still be successful if the attacker is too determined. The expected impact remains Medium.
20	As in the threat number 11, two-factor authentication can be implemented with a time-based one-time password as the second factor. With such countermeasures, the likelihood of the attack drops to Low. The expected impact remains Medium.