System Description and Risk Analysis

Loris Reiff Miro Haller Raphael Eikenberg
Robertas Maleckas

November 21, 2019

Contents

1 System Characterization

This section explains how our system works, what features it provides and how data is passed within the network.

1.1 System Overview

The goal of the here introduced system is to provide certificate authority (CA) capabilities for iMovies. Employees can request and revoke digital certificates through the system. The obtained certificates will be used for secure e-mail communication between employees. The system is designed with maintainability, security, reliability and backwards-compatibility in mind.

An overview of our system can be seen in figure 1. The core of this infrastructure consists of a triangle of three servers: web server, database server, and core CA. The web server manages authentication as well as authorization of the users. The database stores user information, which can be updated over the web server, as well as public certificates and encrypted private keys of users. The core CA is responsible for issuing and revoking certificates as well as generating the Certificate Revocation List (CRL). It provides an API to the web server and updates sensitive data in the database (passwords, certificates and encrypted private keys). This triangle is mirrored completely redundant with another three identically configured servers. Our load balancer distributes traffic between those two infrastructures, while monitoring their state and switches to the healthy one in case of failures.

Employees can login from the internet over a secure HTTPS connection only, using their password or certificate. They are provided with a web interface, where they can edit their personal information and request a new certificate and revoke old ones. The web servers also serve the web interface for administrators who may only use certificate authentication.

We use host-based firewalls on every host (symbolised by the brown circles in figure 1). To reduce the already nontrivial resource footprint of our virtual machines, we refrained from using dedicated firewall machines to create a demilitarized zone for the load balancer, web servers and configuration server. However, the behaviour of our system is very similar to the one of a DMZ. While network separation using different VirtualBox networks imitates physical separation, *nftable* rules on the servers decide which traffic to accept.

As for VirtualBox, we use two different networks – namely a public network and a private network, which are visualized by the blue and green cloud in figure 1 respectively. All internal servers can be maintained over the configuration server, which is the only server besides the load balancer, that is accessible from the public internet.

For monitoring purposes all events are logged on the machines and additionally sent to an append only log server.

Lastly, backups of the database, the log server and the configuration server are pushed to the backup server regularly. The core CA servers immediately backup the private keys upon issuing a new client certificate.

1.2 System Functionality

In the following, the functional aspects of the system will be presented.

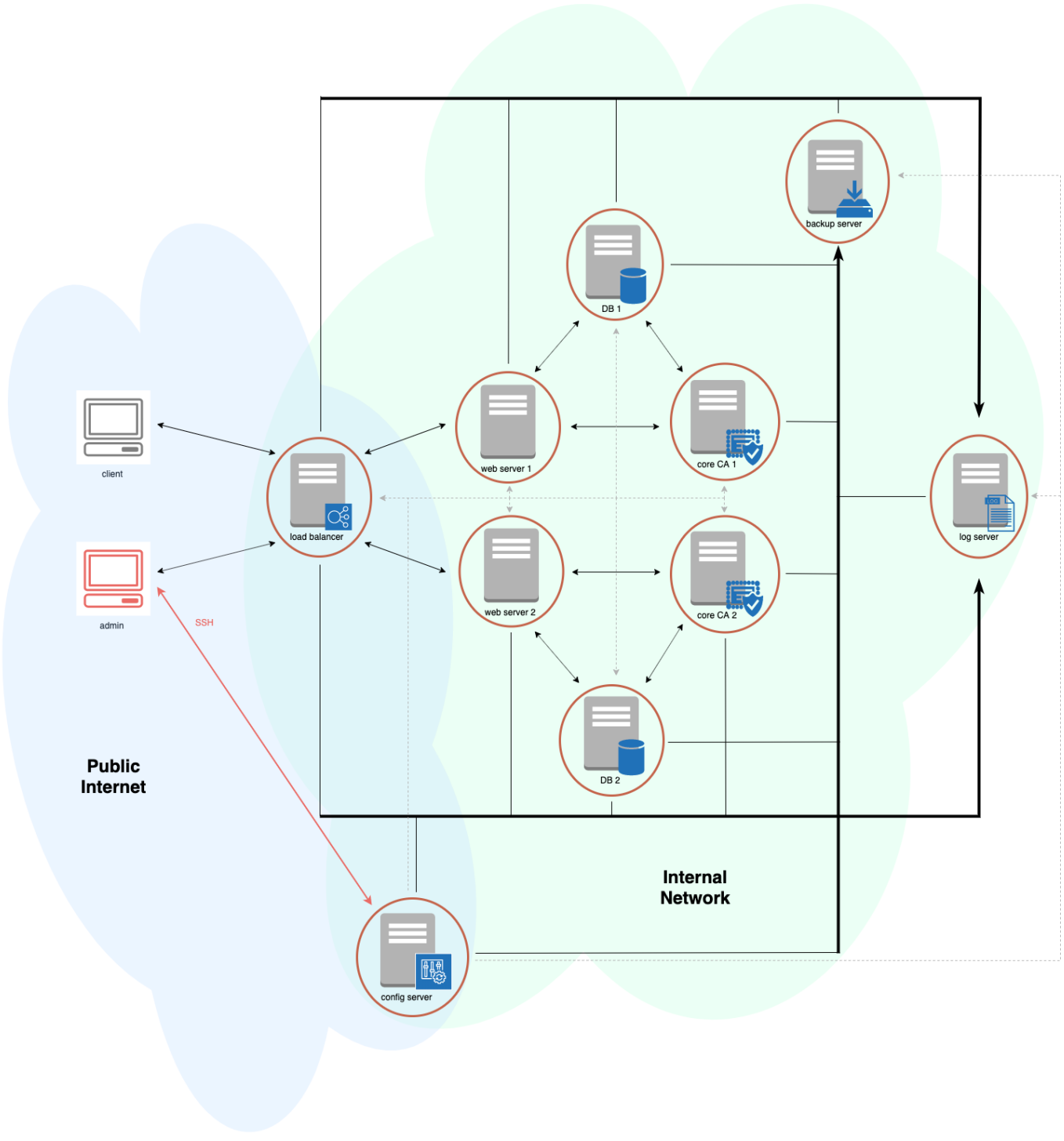

Figure 1: Overview of the network layout.

1.2.1 User Authentication

To change account information or manage certificates, a user must login to access the web interface. To authenticate, the user can either use their username and password or standard TLS client certificate authentication.

1.2.2 Certificate Issuing

Users may obtain a certificate through the web interface. They have to enter their password and a new passphrase to do so. The latter will be used to encrypt the archive containing the generated private key and the certificate. The certificate can be used to communicate via e-mail securely as well as authenticate with the web interface. Users can only hold one valid certificate at the time; hence, requesting a new certificate will be revoke the old certificate (if held) automatically.

1.2.3 Certificate Revocation

Once a private key is lost or leaked, a user should revoke their keys to guarantee the integrity of the system. This can be done by logging in to the web interface and using the revocation form provided or implicitly by requesting a replacement certificate (they will be offered an archive including the newly generated private key).

The web server maintains the Certificate Revocation List (CRL) under the publicly accessible URL `http://imovies.ch/crl/revoked.crl`. All certificates issued by the Core CA have the CRL Distribution Point (CDP) set to this address, which will be resolved to check the validity of a given certificate.

It is common practice to serve the CRL over HTTP, because some clients refuse to download it over HTTPS since this could lead to endless certificate verification loops. We do not consider this a security risk as the CRL is signed by the root CA and it contains the timestamps of the current and next updates to prevent MITM replay attacks. Traffic to all other HTTP routes besides the CRL is redirected to HTTPS.

1.2.4 Profile Information

Every user has certain profile information bound to their account, including their name and the e-mail address. The e-mail address is attached to the certificate, so other users can tell with whom they communicate with.

The profile information can be updated by the user via a web interface. The user cannot change their email address to the email address of another user. If the information is changed successfully, the previous certificate of the user is revoked, and a new certificate may be requested.

In our system, the user must remember the passphrase used to encrypt the private key in order to be able to import it into their browser or email client. This de-centralizes the protection of the private key and certificate archives stored in the Core CA since the passphrase is not recorded, preventing a single point of failure in case of a certificate database leakage.

1.2.5 CA Administration Interface

CA administrators have access to an additional administration panel, provided they authorize themselves with a client certificate. Username-and-password authentication is disabled for CA administrators since this would allow for a "loophole" of authenticating using a password to obtain a new certificate, effectively bypassing this requirement. CA administrators can obtain new certificates by visiting the system administrators. The administration panel allows them to see the following information:

- number of issued certificates
- number of revoked certificates
- current serial number

1.2.6 Backup

Backing up configurations and private keys is crucial in case something goes wrong. While by themselves the backups are not important to the functioning of the systems, they are needed to recover the system in case of data loss.

All relevant data is archived daily at 3:00 UTC and saved to an append-only off-site storage. Specifically, the backups include the configuration of the system, database dumps, log files and private keys for all certificates ever issued. Additionally, if a new certificate is issued, a backup of the private key will be created immediately. The backup process is implemented in a push-based approach using Borg [1] a well tested open source solution. Data in backups can be restored partially or in full through a manual process initiated by a system administrator.

1.2.7 System Administration and Maintenance

The system is accessible for administration over the Internet. To minimize exposure, the internal hosts can only be accessed over the configuration server. Each system administrator has a user on this server, where their public key is stored as authorised key for SSH access. They can then change to the 'ansible' user, which has the key material to access all internal servers via SSH.

1.3 Security Design

In this section, we will elaborate on the security of the system, including considerations regarding access control, key and session management as well as data integrity.

1.3.1 Authentication and Access Control

Upon initiating a TLS connection to the CA web interface, the user may provide their client certificate for authentication. The basic certificate validation performed by the server includes temporal validity, Extended Key Usage (EKU) and Certificate Revocation List (CRL) checks. Furthermore, the server validates the certificate's issuer specifically to guard against impersonation attacks using a rogue CA that the web server machine may trust by default.

As an alternative to client certificate authentication, the web interface supports username and password authentication. The user's credentials are sent to the server over a secure (TLS) connection, and upon successful authentication a short-lived session token is issued. Subsequent authenticated requests with this token in the HTTP headers cause similar data freshness checks to ensure the identity claims contained in the token are always fully in-sync with the database.

In our system, we enforce a minimum password length of 8, and a minimum passphrase length of 12. Passwords are used for login, while passphrases are used to secure private certificates in transit. Furthermore, we disallow the use of common passwords by checking against a password list prior to password change. Note that this password policy is only enforced for new passwords on a password change request, while the original passwords given in the legacy database remain unchanged (and may not meet the policy).

1.3.2 Session Management

Certificate based session management largely depends on the user's browser, responsible for (optionally) including the client certificate with authenticated requests. We observe that two widely popular browsers - Google Chrome and Mozilla Firefox - prompt the user to select a client certificate to be used (if available) and may remember the choice for subsequent requests until the browser is restarted. From the server's perspective, a valid (non-expired and non-revoked) employee's certificate is always accepted for authentication.

Following successful username and password authentication, the session token is stored in a browser cookie with the *HttpOnly* [3] attribute and a short lifetime (under a day). This follows the principle of minimum exposure by reducing the window of opportunity and risk of session hijacking attacks (e.g. through XSS).

1.3.3 Data Integrity

As outlined in the sections above, the system performs extensive integrity checking at every step of its function to ensure there is no de-synchronization between the database, any active CA web interface sessions and valid certificates. Any change of the database immediately causes any cookie-based sessions to be refreshed; the web interface reacts to changes instantaneously, however it is the responsibility of the client system (e.g. e-mail client) to ensure the window of opportunity for certificate misuse due to any propagation delays such as CRL caching (if present) is sufficiently small.

1.3.4 Additional Defense Measures

Each host is equipped with a host-based firewall. The firewall rules follow the principle of secure, fail-safe defaults. Any connection that is non explicitly allowed is dropped immediately. We install restrictive rules based on the needs of our service.

The web server interface is furthermore hardened against several specific attacks. Standard framework-provided mechanisms guard all non-idempotent API endpoints against Cross-Site Request Forgery (XSRF) attacks by including nonce values in the corresponding data entry forms. Sensitive (e.g. sign-in) or computationally expensive (e.g. certificate issuance incl. keypair generation)

operations are rate-limited to reduce the risk of unauthorized access using brute-force or credential stuffing attacks as well as Denial-Of-Service (DOS) using an excessive number of requests. Separate, dedicated data binding models are used for any requests involving user data (e.g. sign-in, personal information update), limiting the surface area for mass assignment ("overposting") attacks.

1.3.5 Maintenance

The configuration of the complete infrastructure is automated using Vagrant[1] and Ansible[2] in order to reduce the chance of human errors. Additionally, this improves scalability, maintainability and enables a faster recovery in case a server has to be setup from scratch. Vagrant is an open-source software for building virtual environments, including "real" production servers running on a type-1 hypervisor. Additionally, it improves the development process by enabling the creation of an accurate local mirror of the production infrastructure using virtual machines. Ansible is the state-of-the-art for configuration management in industry. It automatically configures and provisions the servers, depending on their role (database, web, backup or logging server), following the principle of least privileges.

For manual interaction, system administrators can remotely access all servers using the configuration server as a jump host. This design decision prioritizes simplicity and usability at the cost of a single point of failure. Although this makes the configuration server a critical asset in our infrastructure, it enables us to minimize the exposure to one host, i.e., we only have to provide controlled access to the SSH port of the jump host. All the other servers are not directly accessible from the public Internet via SSH due to strict firewall rules.

Physical access to the servers is supervised by iMovies' security guards. The server room is cleaned regularly and defective parts are replaced timely by the internal system administrators.

1.3.6 Certificate Issuing

Clients have to re-enter their password to issue new certificates. This implements the principle of minimal trust: In case the web server is compromised, it cannot issue new certificates for all users. The reason is that it cannot issue new certificates without knowing the password. Until a user logs in using password instead of certificate authentication, the web server has only access to hashed passwords. Additionally, it cannot change the passwords in the database because this runs over an API call of one of the core CAs, where the old password must be supplied.

1.3.7 Core CA Key Management

Keys are generated using the widely used command line tool OpenSSL as well as the standard *System.Security.Cryptography* library of ASP.NET Core 3.0, which both provide secure randomness. The private key used to create the self signed root CA certificate is stored offline to prevent leakage. This certificate is installed in the root of trust of all servers of the internal architecture as well

[1] `https://www.vagrantup.com/`
[2] `https://www.ansible.com/`

as of all users. Multiple intermediate CA-enabled certificates are signed with the offline root CA key: Every core CA server generates two key pairs, one for signing internal TLS certificates and one for signing external user certificates, and obtains an intermediate certificate for each of them. This follows the common cryptographic principle of using different keys for different purposes (which could be seen as an instance of the compartmentalization principle) and prevents that a certificate can be abused in another context. Using intermediate certificates has the advantage that if one gets compromised or expires, it can be revoked and a system administrator can create a new one and sign it with the offline root key simply by running a dedicated Ansible playbook. Neither internal servers nor client devices have to be updated, since by transitivity of trust, they automatically trust the new intermediate certificate. In case the root CA certificate must be revoked, we would have to update the root of trust of both. New CRLs (for the root CA and the internal intermediate CAs) are generated and distributed automatically by Ansible.

The key pairs for user certificates are generated on the core CA. The private key is asymmetrically encrypted with a special public key for backups before it is sent to the backup server. The corresponding backup private key is stored offline so neither the core CA nor the backup server can read the user's private key, implementing the principle of minimal trust. However, system administrators can recover it with the offline backup private key as required.

Additionally, the core CA stores the archive, containing the issued certificate and the user's private key, encrypted with a user provided passphrase in the database. Because this passphrase is not stored persistently, the core CA cannot decrypt the archive. However, user can still download the encrypted archive at a later point in time (e.g. because they may want to install it on multiple devices). It is the user's responsibility to remember the passphrase and if they do not, a new certificate should be requested. If necessary, encrypted data could still be recovered manually by a system administrator using the backup of the user's private key (backed up before passphrase encryption).

1.3.8 Availability

The web, database and core CA servers, which are essential for the functionality of our system, are implemented with geographic redundancy. Moreover, load balancing mechanisms are deployed so servers can split the work fairly. They also monitor the health of all servers and switch to the functional instance if there is a problem with the other one. This achieves single failure tolerance. However, if for example the core CA of one and the web server of the other fail, then our service is unavailable until manual actions are taken, although a server of each type would be functional. We made this design decision, instead of connecting every server to all others, for the sake of simplicity. The other solution would require internal load balancers, which would consume more resources (we are limited in the memory usage of our VMs) and provide more attack surface.

1.3.9 Application Privileges

To decrease the chance of privilege escalation, the privileges of applications are chosen in accordance to the principle of least privilege. That means, only as much privileges shall be given to said programs as needed for correct opera-

tion. Applications are assigned their own user, and only paths required for the application to run are made available to that user.

1.3.10 Logging

In order to easily troubleshoot issues and to investigate on-going but also past attacks, we store the logs on a centralized log server. The logs of the individual servers are sent to the log server over a secured channel. It ensures that the logs are append only and an attacker who has compromised another server of the system is unable to alter the logs and cover their tracks. Besides sending logs to the log server, logs are kept locally in case of transmission failure.

To implement this, we use the state-of-the-art logging framework rsyslog. It allows us to authenticate the logs using certificates provided by our core CA.

Logs will also be backed up daily in case data loss on the log server occurs.

In advanced setups, monitoring could be deployed to check resource usage and access on the hosts. Since this introduces new complexity and maintenance effort, we refrained from adding monitoring capabilities to our system.

1.3.11 Data in Transit

To prevent internal man-in-the-middle attacks and lower the chances of espionage, data sent within the network is always authenticated and encrypted. For instance, the communication between web server and database as well as the data exchange between web server and CA core servers are end-to-end encrypted with TLS. A passive attacker will therefore not be able to decrypt the traffic. Generally, certificate based authentication is used, with one exception: The communication with the client authentication to the database is done using a password, otherwise client TLS certificates are used. The communication for the backups happen over a secure, restricted SSH connection provided by borg [1].

1.3.12 Data at Rest

Data that is currently not in transit is an attractive target for attackers, since usually larger amounts are saved in one place. In our system, we generally try to save only a small amount of sensitive data, and aim to enforce best practice where applicable. This is in accordance with the principle of minimum exposure. However, we opted for usability & maintainability in the sense that we do not encrypt the whole disk, as it would hinder remote administration substantially. Instead, we introduce encryption & cryptographic primitives on sensitive data at rest, if applicable.

The CA core servers generate user privates keys and certificates. These are only stored encrypted in the database, using a passphrase chosen by the user.

Passwords are stored SHA1-hashed in the database. This is bad practice, however, due to legacy reasons, we could not migrate to a more secure hash function yet. To ensure a smooth transition, this could be done incrementally by implementing support for different password storage modes (e.g. SHA-1 and bcrypt) and replacing the legacy passwords by securely hashed (with salt) ones upon password change to avoid forcing everyone to change at once. The process could be expedited by contacting the users and urging them to change their passwords in a timely manner (with a deadline after which they would need to contact system administrators to regain access).

Data on the backup server is encrypted using symmetric authenticated encryption, except for private keys, which are additionally encrypted asymmetrically as described in section 1.3.7. The keys for the AEAD are not stored on the backup server, but on the individual servers that are backed up. Also only the backup user has access to the key, this follows the principle of least privilege.

Additionally, we designed our system with compartmentalization in mind. For instance, no (long-term) user data and certificated data is stored on a web server, the load balancer or on the config server. On the log server the user data is limited to the user id.

Access to the database tables is restricted with the principle of least privileges in mind. And only users from the right host are allowed to access the data provided they authenticated themselves correctly using a password.[3]

To decrease the probability of physical data loss, a RAID 6 with at least 8 disks is used on the backup server to ensure stable data storage. On the other services we use RAID 5 with 5 disks.

1.3.13 Hiring Process

Background checks are performed on candidates for positions with access to sensitive information, such as security guards or system administrators. Their criminal record and financial situation is regularly checked.

1.4 Components

Our system requires communication among different applications running on various machines.

1.4.1 Platforms

Web Servers End-user[4] servers host the web interface, handling authentication and mediating the information exchange between users and the "backend" infrastructure (database and core CA).

Load Balancing Server(s) Dedicated proxy servers that distribute user sessions among different Web Server machines in a round-robin fashion, distributing the load and providing improved availability in case of partial system failures.

Database Servers The legacy database is hosted on non-public-facing machines separate from the core CA and web server, thus reducing its attack surface.

Core CA Servers A HTTPS API is provided to web servers offering the following functions: Issue new certificates, revoke existing certificates, re-download an encrypted archive (containing the users private key and certificate), generate a new CRL and change the user password. They access the database servers to authenticate users or update their passwords. Additionally, they populate the public certificate's table with new certificates and update the private certificate table with the last issued, encrypted archive.

[3] see ansible config: `~/roles/init_db/tasks/create_users.yml`

[4] not directly exposed to the public web, located behind dedicated load balancing boxes.

Configuration Server At the heart of the system a configuration server is used to configure all other machines using SSH access with a dedicated user. It can be accessed over SSH by remote administrators.

Log Server Logs of all servers are centrally stored on a log server. In case a server is unreachable, the logs will allow system administrators to trace down the problem without having to physically access the machine. Hence, high availability of the log server is necessary to maintain a continuous record of events of all servers.

Backup Server The backup server stores important information of the system for a certain amount of time. The config server, log server, database servers and ca servers push their backups to this platform over a secure SSH connection.

1.4.2 Applications

Web Interface An ASP.NET Core MVC web interface for users and CA admins running on Kestrel behind an nginx reverse proxy, hosted on the web server machines.

HAProxy Proxy service with load balancing capabilities running on dedicated load balancer machines.

Ansible A configuration management, and application-deployment tool used to configure, maintain, scale and deploy our system. A authenticated administrator can use ansible to deploy configurations to an arbitrary server of the system.

CA Service An internal application responsible for generating and managing private keys and certificates.

MySQL/MariaDB (with Galera) A Database backend running on the database servers. It communicates with other authenticated MariaDB processes on other servers over a secure TLS connection in order to provide redundancy.

nftables nftables is set up on each machine to restrict traffic to not explicitly allowed ports. The rules follow the principle of secure, fail-safe defaults.

SSH SSH is used to access the machines and is installed on all platforms. However, access is limited as specified in the security design.

rsyslog rsyslog is used to manage our logs, which are all sent to the log server which stores the logs in a centralized manner.

Borg Backup Borg Backup is installed on the Core CA servers, the log server, the Ansible server and database servers. It is also installed on the backup server, where it acts as backend for the clients.

1.4.3 Data Records

User Information User information is stored in the database on the database servers as well as on the backup server. It includes their user id, email address, first and last name.

Logs The logs of each service is stored locally in `/var/log/syslog` as well as on the append only log server. An attacker shouldn't be able to modify the logs.

Configuration files The configuration files are generated with the ansible files. They are thus available on the config server and the generated versions of the necessary configuration files is on the respective servers.

User Certificates & Private Keys Employee private keys are stored with the certificates in an encrypted PKCS#12 archive in the database, using a user-chosen passphrase (required) and are available to be re-downloaded from the web interface later. Upon key generation, the keys are also archived on the backup servers, encrypted under the backup keys.

SSH Private Keys The config server has a SSH private key that allows access to all the other servers. Additionally, administrators have a SSH private key to access the config server. Furthermore, every server that has data that is backed up has a SSH private key that is specifically for this use case. Those private keys authorize the servers to perform a backup on the backup server.

Server Certificate & Private Keys Each server has a private key and a corresponding signed certificate that is used for secure communication between the components of our system.

1.5 Backdoors

As required by the assignment, two backdoors are incorporated into our system. Both of them allow a remote attacker to compromise the system.

1.5.1 Target-Initiated Remote Code Execution

The Core CA (CertServer) binaries are patched with malicious code that polls a hard-coded attacker-controlled HTTP endpoint every 30 minutes. If the endpoint's response body satisfies certain conditions (length > 3; could include more sophisticated checks), the response is dumped into a temporary file and executed as a bash script, effectively allowing (scheduled) remote code execution.

For obfuscation, the backdoor code is disguised under a fake update check which serves no real purpose beyond triggering similar traffic and log entries by polling a hard-coded build publishing endpoint and comparing the latest build version against the installed one.

Using this mechanism, the attacker may run arbitrary commands under the Core CA service user's privileges, which include access to the intermediate CA certificate private keys that can be used to issue end-user client certificates and thus impersonate arbitrary users. They are also able to steal existing (passphrase-protected) certificates from the database, manipulate CRL and certificate DB records, and more, effectively taking over all critical Core CA functions.

1.5.2 Off by Slash – Deadly Path Traversal

We setup nginx such that a path traversal vulnerability is introduced. The vulnerability was presented at Black Hat USA 2018 by Orange Tsai [4] and

appeared first at HCTF 2016. The path traversal vulnerability arises from a missing trailing slash on an "alias" directive's argument in nginx and enables the attacker to read the contents of the immediate parent directory (i.e. one level up).

In the case of our app, the alias "/static" is mapped to the "wwwroot" directory inside the application root. Consequently, the backdoor allows an attacker to read all files in the application root directory (one level up from "wwwroot"), e.g. `https://imovies.ch/static../appsettings.json`.

Malicious code hidden in our web application binaries registers a file system event listener to detect read accesses to ".json" files in the application root directory. The backdoor code tracks such events over time, and upon registering 10 reads over a 5-minute sliding window, a web shell (also embedded in the application binaries) is triggered, allowing arbitrary code execution under the application service account by issuing a GET request to the publicly available endpoint: `https://imovies.ch/Account/Login?log=[URLencodedcommand]&logName=/bin/bash`. The command execution resets the counter, requiring a new sequence on reads on the JSON file to re-activate the web shell again.

From here on the attacker can compromise the purpose of the system. The intruder can for instance obtain the unsalted, weakly hashed passwords by querying the database server and crack them offline. They can then impersonate the web server by accessing its TLS client certificate used to authenticate with the Core CA, which allows the attacker to obtain certificates for any users whose passwords were cracked as well as lock the real account owners out of the system by changing their account details.

2 Risk Analysis And Security Measures

In this section we evaluate the risks of our system and discuss the measures taken to minimize these risks from having impact.

2.1 Assets

In this section, we will point out what assets the system consists of. This evaluation only considers assets of the company as well as employees and staff of iMovies.

2.1.1 Physical Assets

The servers in Switzerland are located in the basement of iMovies' headquarter in Zurich, while those in Iceland are located in a dedicated room in the office in Keflavík.

Core Infrastructure The core functionality is provided by a web server, core CA server and database server. This infrastructure is completely redundant, with one instance in Switzerland and the other in Iceland. As for the state space, it might be fully functional (all servers operating as expected), partially functional (only one site runs as expected) or not functional (both sites have not functional servers). Additionally, each server can be compromised or not.

Supportive Infrastructure The core infrastructure is supported by a load balancer (CH), log server (IS), backup server (IS) and the configuration server (CH). Those servers are not replicated. Their state space is a tuple of two Boolean values declaring whether or not a server is functional and whether or not it is compromised.

2.1.2 Logical Assets

User data That is user id, e-mail address, password, first and last name. This data must be treated confidentially, even though some information is not critical. Access should be restricted to principals that need the data to fulfill their function. The users data's state space is described as set of people who have access to the data.

Private Keys in Database The private key access should be limited to the owner of the corresponding certificate.

Public Certificates in Database The public certificates can be considered public knowledge. However, only legitimate principals should have write access, i.e. those who uphold the correct mapping of users issued the certificate with the corresponding certificate.

Logs The logs must be authentic and ideally confidential as they might reveal meta data which can be used to facilitate other attacks. Furthermore, the meta data reveals privacy relevant aspects about users. The state space is the set of people who have read from the log and principals who have written to the log.

Backups Backups are important to recover from data loss. The backed up servers should have read access to their backups if they have read access to this data anyway. This allows them to easily recover. The private keys of users shouldn't be accessible on the other hand. The people with access to the backup server shouldn't be able to access the backed up files.

Configuration Files Configuration files should only be modified by system administrators. Therefore the configuration files have to be authentic. The state space associated with the configuration files consists of all people who have modified the file at what time.

Certificates Certificates should be considered public as they have to be shared in order to communicate securely by e-mail with employees of iMovies. The state space is a boolean value that indicates if it is safe to use the certificate. The certificate is safe to use if it has not been revoked and if the private key has not been compromised.

Private Keys for User Certificates A private key is given to the user together with a certificate when they issue it. It has to be treated confidentially. The state space associated with the private keys consists of all people who have accessed the private key.

Private Keys for Intermediate CAs Private keys for intermediate CAs have to be confidential as they can sign new certificates either for users or for internal TLS communication. Their state space is the set of people who have accessed them.

Private Keys for Root CA The private key for the root CA is confidential as it can sign new intermediate CA certificates. The state space is the set of people who have accessed it.

Private Keys for Backup The private key for the backup is confidential as it can be used to decrypt user's private keys. The state space is the set of people who have accessed it.

SSH Private Keys SSH private keys have to be confidential as they allow administrators to login into other systems. The state space of SSH private keys is the set of people who have accessed them.

CRL The root CRL contains revoked intermediate CA certificates, the internal intermediate CA CRL contains revoked TLS certificates for the internal infrastructure and the external one contains revoked client certificates. They have to be valid and authentic to prevent the use of revoked certificates.

Internet Connectivity Internet connectivity is important to guarantee users the access to the CRL and system availability for certificate issuance and revocation.

User Session An active user session is effectively a temporary credential to act on the user's behalf and should be considered confidential, with the state space being people with access to it.

iMovies Software The integrity of iMovies in-house developed software has to be guaranteed.

Domain Name The domain name must stay in possession of iMovies, as the system could be compromised otherwise.

2.1.3 People

Non-Technical Employees The employees of iMovies are highly important to the company, simply due to the fact that their work is the main reason for operating secure communication on the network overall. All people that work for iMovies observe issues with the system and report these to the system staff. As a consequence, without them the correct functioning of the system cannot be validated.

System administrators People in charge of the functioning of the system play a special role in that they know a lot about the system structure, the interfaces between systems and the protocols and technologies used in the network. Enabling system administrators to accomplishing their tasks is essential for operating a secure network. Some system administrators have access to confidential information or may have the knowledge to acquire it. Therefore, this group of employees has great value to the company.

CA administrators The most critical group of employees are CA administrators, who definitely have access to confidential information.

2.1.4 Intangible Goods

Reputation To the company, the customers are everything. If the public thought bad of iMovies, there are likely less customers to expect. This means, from iMovies' point of view, the reputation is of great value.

2.2 Threat Sources

For iMovies, we could identify the following threat sources.

Nature The servers are located in Keflavík, IS and Zurich, CH. Zurich has especially low seismic activity [2]. However, one still has to account for earthquakes, as it is by no means impossible for them to occur. Hence, a second location in Iceland was chosen. Also other natural factors like lightnings may interfere with the operation of the servers or network.

Employees Employees include the ordinary employees of iMovies who use the system, the security staff, the concierge, the cleaner as well as the administrators of the system. They might get paid by national agencies in order to leak information or install malware. It is also possible that they are blackmailed by organized criminals. They could also be frustrated by their employer and want to damage the reputation of the company.

Visitors Visitors of iMovies could attack the system by various means. As has been observed in the past, a third party could try to gain access to iMovies' offices, where they actively enable remote attackers to access the network. Even without access, by-passers can spread inconspicuous USB sticks in the area, hoping for iMovies employees to plug them into their workstation.

Independent Contractors External services have access to both the offices and the data centers, e.g. for cleaning. They could be involved into an attack, which is why their access is critical to the security of the system.

Ex-Employees Unsatisfied ex-employees could try to damage the company using their knowledge about the system. Even non-technical ex-employees have had access to the system already and can use their understanding of the inner workings for their advantage.

Governmental Agencies This mainly includes the executive of governments about which iMovies is reporting. Governments could be offended by iMovies investigative reporting. Additionally, they could intend to prosecute potential informants.

Organized Crime Organized criminals may want to *silence* an informant that revealed precious information about the inner workings of the crime organization to iMovies.

Script Kiddies Script kiddies do not specifically choose their target. Usually, their methods are not highly sophisticated, and the attacked systems run unpatched software. If the exposed parts of our system are not kept up-to-date, they could become the victim of script kiddies, too.

Skilled Hacker A skilled hacker might be hired by organized criminals or nation state adversaries and attack iMovies. Skilled hackers do not solely rely on public exploits and can either come up with their own exploits, or buy exploits when supported with enough money to do so.

Malware Directed malware (e.g. from a state actor) or undirected malware (e.g. from a script kiddy) must be taken into account. In the past years, especially the spread ransomware has increased. Our system must hence protect against self-spreading software.

Competitors are not considered a threat, as they do not gain a big advantage by knowing about what iMovies is going to report on. Moreover, the investigative reporting and the movie industry are not known for aggressive competitive behavior. *Terrorists* are also not regarded as a threat source, as they would not cause visible damage to the public by interfering with the CA system of iMovies. If they wanted to stop a publication of a investigative report, they will behave similar to organized crime.

2.3 Risks Definitions

In the following, we define likelihood, impact and risk level.

Likelihood	
Likelihood	Description
High	The threat source has power and resources which greatly exceed those of the defense mechanisms and is motivated to exploit this advantage.
Medium	The threat source has power and resources which are comparable to those of the defense mechanisms and is partially interested in exploiting this advantage.
Low	The threat source has power and resources which are generally inferior to those of the defense mechanisms or it has little interest to exploit a possible advantage.

Impact	
Impact	Description
High	A great decrease in values of assets, tangible or intangible, living or non-living.
Medium	A substantial decrease in values of assets, tangible or intangible, living or non-living.
Low	A minor decrease in values of assets, tangible or intangible, living or non-living.

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 section, we list all potential threats and the corresponding countermeasures. We estimate the risk based on the information about the threat, the threat sources and the corresponding countermeasure.

2.4.1 *Evaluation Asset Core Infrastructure*

No.	Threat	Countermeasure(s)	L	I	Risk
1	Nature: Earthquake, major natural disaster destroys server	The core infrastructure (web server, database and core CA) is completely redundant (instances in CH and IS)	*Medium*	*Low*	*Low*
2	Nature: Hardware failure, defective component	Regular backups, geographically redundant infrastructure, an administrator is on call 24/7	*Medium*	*Low*	*Low*
3	Nature: Pollution, dust decreases lifespan of components	Clean server room every three month by specialized cleaning crew	*Medium*	*Low*	*Low*
4	Governmental Agency, Independent Contractors, System Administrator: Malicious hardware is added	Cleaning staff is monitored by iMovies security staff, background checks are performed on system administrators before hiring them	*Medium*	*High*	*Medium*

2.4.2 *Evaluation Asset Supportive Infrastructure*

No.	Threat	Countermeasure(s)	L	I	Risk
5	Nature: Earthquake, major natural disaster or hardware failure destroys configuration server	Fast automatic reconfiguration of new Ansible server with Vagrant, Ansible playbooks with server configurations stored in remote git repository	*Medium*	*Low*	*Low*
6	Nature: Earthquake, major natural disaster or hardware failure destroys log or backup server	Server locations have low seismic activity	*Medium*	*Medium*	*Medium*

2.4.3 *Evaluation Asset User Data*

No.	Threat	Countermeasure(s)	L	I	Risk
7	Script Kiddy: a script kiddy hacks a database or backup server and leaks user data	None of the mentioned servers is directly accessible from the internet, we keep our software updated	*Low*	*Medium*	*Low*
8	Governmental Agency, Organized Crime, Skilled Hacker: penetrate a database server or backup server and obtains user data	Not directly accessible servers, firewalls	*High*	*High*	*High*
9	Governmental Agency: gains physical access to a database or backup server	Strict access control to servers, and background check of people with access	*Medium*	*High*	*Medium*

2.4.4 *Evaluation Asset Private Keys Database Table*

No.	Threat	Countermeasure(s)	L	I	Risk
10	Employees, Independent Contractors: physical access or abuse of access rights to dump the database table	Private keys are encrypted with a user provided passphrase, which is not stored persistently	*Medium*	*Low*	*Low*
11	Skilled Hacker: compromise database server by exploiting configuration or software issues to leak data	This server is not accessible from the public network (firewall rules), the private keys are encrypted	*Low*	*Low*	*Low*

2.4.5 *Evaluation Asset Public Keys Database Table*

No.	Threat	Countermeasure(s)	L	I	Risk
12	Skilled Hacker: replace users public key with his own for impersonation attacks	Only the certificate servers, which are exclusively accessible from the internal network, have write access to the database; they provide an API to edit this table that requires knowledge of the user's password to use	*Low*	*Medium*	*Low*
13	Employees, Independent Contractors: physical access or abuse of access rights to modify public keys to impersonate a coworker	Users will notice, because they will not be able to decrypt emails anymore nor use their private key for login	*Medium*	*Medium*	*Medium*

2.4.6 *Evaluation Asset Logs*

No.	Threat	Countermeasure(s)	L	I	Risk
14	Nature: Logs are lost due to natural disaster or hardware failure	Regular backups of logs	*Medium*	*Low*	*Low*
15	Governmental Agency, Skilled Hacker: Gain access to logs by compromising log server	Log information is restricted to necessary, non-sensitive information, log server is not exposed to the public network	*Low*	*Low*	*Low*
16	Skilled Hacker: Tamper with logs to cover tracks of malicious activities	Logs are append only (internal servers cannot modify existing entries), logs are backed up regularly	*Medium*	*Medium*	*Medium*

2.4.7 *Evaluation Asset Backups*

No.	Threat	Countermeasure(s)	L	I	Risk
17	Employees: Misconfigured or disabled backup routine	Regular manual backup evaluation	*Low*	*Medium*	*Low*
18	Employees: Physically access backup server and copy the hard disk	User private keys are protected by asymmetric encryption where the decryption key is kept offline, logs, databases and configuration files are encrypted symmetrically	*Medium*	*Medium*	*Medium*
19	Skilled Hacker, Malware: Encrypt backups and demand ransom	Backup server is only accessible from the internal network, configuration files, logs, and the database are also stored on the running hosts, users private keys can still be retrieved by users from the passphrase encrypted database table entry	*Low*	*Medium*	*Low*

2.4.8 *Evaluation Asset Configuration Files*

No.	Threat	Countermeasure(s)	L	I	Risk
20	Employees: intentional or accidental mistake in Ansible configuration files	Automated configuration can be tested on accurate local mirror of the instance which is created using Vagrant	*Medium*	*Medium*	*Medium*
21	Skilled Hacker, Malware: encrypt configuration files, demand ransom	Specially hardened server, firewall, configuration files are backed up and stored in a remote git repository	*Medium*	*Low*	*Low*

2.4.9 *Evaluation Asset Certificates*

No.	Threat	Countermeasure(s)	L	I	Risk
22	Employees: the owner of the certificate loses access to the corresponding private key	A replacement certificate is issued, restoring a secure communications channel	*Medium*	*Low*	*Low*
23	Ex-Employees: certificate holders following their departure from iMovies misuse their certificates prior to revocation	Certificates are revoked in a timely manner upon a holder's departure from the company	*Low*	*Medium*	*Low*
24	Skilled Hacker, Government Agency: a motivated individual obtains an employee's login credentials and issues a new certificate to impersonate them	Only one certificate per employee; users will notice (login fails, email decryption not possible)	*High*	*Medium*	*Medium*

2.4.10 *Evaluation Asset Private Keys (for Certificates)*

No.	Threat	Countermeasure(s)	L	I	Risk
25	Employees: either through negligence or malice an employee allows an unauthorised party access to their private key	Password-Based Encryption (PBE) is enforced upon certificate issuance to encrypt the archive which contains the user's private key, certificates can be revoked	*Medium*	*Medium*	*Medium*
26	Skilled Hackers: a capable and motivated party steals an employee's private key through a targeted hacking attack	Private keys are encrypted using a passphrase upon certificate issuance	*Low*	*Medium*	*Low*
27	Governmental Agencies: a well-funded state actor brute-forces the private key using great computational resources or standard-library CSPRNG weaknesses unknown to the public	Large keys are used in combination with limited certificate validity periods	*Medium*	*High*	*Medium*

2.4.11 *Evaluation Asset Private Keys for Intermediate CAs*

No.	Threat	Countermeasure(s)	L	I	Risk
28	Employees: access the core CA and sign bogus user or internal TLS certificates for impersonation	Only system administrators have access to the core CA, their actions are logged and background checks are performed before hiring them	*Medium*	*High*	*Medium*
29	Organized Crime, Governmental Agency, Skilled Hacker: Compromise core CA and steal private key to impersonate and decrypt emails of arbitrary informants	The core CA is only accessible from the internal network, it provides a small attack surface by using a small well defined API, only few system administrators have access to it, intermediate certificates can be revoked easily without changing the client's root of trust	*Low*	*High*	*Low*

2.4.12 *Evaluation Asset Private keys for Root CA*

No.	Threat	Countermeasure(s)	L	I	Risk
30	Employees: Use the root CA's private key to sign intermediate CA, user or internal TLS certificates for impersonation	The root CA's private key is stored offline in a well protected safe, to which only few system administrators have access	*Low*	*High*	*Low*
31	Organized Crime, Governmental Agency, Skilled Hacker: Compromise core CA and steal private key to impersonate and decrypt emails of arbitrary informants	The root CA's private key is stored offline	*Low*	*High*	*Low*

2.4.13 *Evaluation Asset Private Keys for Backup*

No.	Threat	Countermeasure(s)	L	I	Risk
32	Employees: use the private key to decrypt the backup of user's private keys to eavesdrop or impersonate them	The backup private key is stored offline in a well protected safe, only few system administrators have access to it	*Medium*	*High*	*Medium*
33	Organized Crime, Governmental Agency: Compromise the backup key to recover user private keys and eavesdrop on the communication of arbitrary informants	The backup private key is stored offline in a well protected safe	*Low*	*High*	*Low*
34	Malware, Skilled Hacker: Encrypt backup private key and request ransom	Key is stored offline, users can still download their private key from the database, where it is stored encrypted with a passphrase	*Low*	*High*	*Low*

2.4.14 *Evaluation Asset SSH Private Keys*

No.	Threat	Countermeasure(s)	L	I	Risk
35	Employees: either through negligence or malice a system administrator allows an unauthorised party access to their private key	Activities are logged and each administrator has a personal key, administrators are trained	*Medium*	*High*	*Medium*
36	Governmental Agencies, Skilled Hacker: perform targeted attack on system administrators to gain access to the internal system and plant a backdoor	To access other servers, additionally to the SSH key, they also have to know the sudo password of this particular administrator's account on the jump host as well as the passphrase for the Ansible user's SSH key	*Medium*	*High*	*Medium*
37	Governmental Agencies, Skilled Hacker: SSH keys for the internal servers are compromised by directly attacking the configuration server	The server is hardened and maintained with special care	*High*	*High*	*High*

2.4.15 *Evaluation Asset CRL*

No.	Threat	Countermeasure(s)	L	I	Risk
38	Nature: Loss of CRL distribution point due to hardware failure or a defective component	Redundant architecture; CRL is hosted on both, geographically separated, web servers	*Medium*	*Low*	*Low*
39	Employees, Skilled Hacker: Remove or garble CRL for DoS by SSL authentication failure	The CRL is re-fetched automatically every ten minutes	*Medium*	*Medium*	*Medium*

2.4.16 *Evaluation Asset Internet Connectivity*

No.	Threat	Countermeasure(s)	L	I	Risk
40	Nature: Submarine cable gets damaged by tectonic plate shift	Redundant infrastructure in CH and IS, both locations are close to multiple submarine cables	*Low*	*Medium*	*Low*
41	Employee: Configuration mistake leads to network outage	Redundant infrastructure with automatic load balancing seamlessly switches to the other instance	*Medium*	*Low*	*Low*
42	Organized Crime, Skilled Hacker: DoS attack to throttle network connection	Redundant infrastructure with load balancing is able to handle more traffic than a single one	*High*	*High*	*High*

2.4.17 *Evaluation Asset User Session*

No.	Threat	Countermeasure(s)	L	I	Risk
43	Governmental Agencies, Organized Crime, Skilled Hacker: access different user accounts, revoke keys, use social engineering	Session timeouts, requesting user password when issuing new certificates	*Low*	*Low*	*Low*

2.4.18 *Evaluation Asset iMovies Software*

No.	Threat	Countermeasure(s)	L	I	Risk
44	Nature: destroying of all digital copies of the software	Redundant storage	*Low*	*Medium*	*Low*
45	Employees: introducing bugs or leaking copies	Automated tests, regular manual checks, strict NDA upon hiring	*Low*	*Medium*	*Low*
46	Visitors: picking up hardware with digital copies	Security scan at entrances and exits	*Low*	*Low*	*Low*
47	Ex-Employees: leaking source code	Extended period of validity of NDA	*Low*	*Low*	*Low*
48	Governmental agencies: introducing backdoors	Hardened network, compartmentalization of services	*Low*	*Medium*	*Low*

2.4.19 *Evaluation Asset Domain Name*

No.	Threat	Countermeasure(s)	L	I	Risk
49	Employees: forgetting to renew the domain registration	Automated renewal	*Low*	*Medium*	*Low*
50	Ex-Employees: trying to transfer ownership	Passwords change when personnel changes	*Low*	*Medium*	*Low*

2.4.20 *Evaluation Asset Non-technical Employees*

No.	Threat	Countermeasure(s)	L	I	Risk
51	Nature: harm to employee due to environmental disaster, loss of manpower	-	*Low*	*Low*	*Low*
52	Governmental Agencies, Organized Crime: kidnapping or controlling of employee, loss of manpower	Office and server room are controlled by security guards	*Low*	*Low*	*Low*

2.4.21 *Evaluation Asset System Administrators*

No.	Threat	Countermeasure(s)	L	I	Risk
53	Nature: harm to system administrator due to environmental disaster, loss of knowledge	Good documentation, automated configuration scripts instead of not repeatable actions	*Low*	*Low*	*Low*
54	Governmental Agencies, Organized Crime: kidnapping or controlling of system administrators, full compromise of the system	-	*Low*	*High*	*Low*

2.4.22 *Evaluation Asset CA Administrators*

No.	Threat	Countermeasure(s)	L	I	Risk
55	Nature: Harm to CA administrator due to environmental disaster, loss of manpower	Encrypted data can be recovered with the backup of the private key	*Low*	*Low*	*Low*
56	Governmental Agencies, Organized Crime: Kidnapping or controlling of CA administrators, leak of internal information	-	*Low*	*Low*	*Low*

2.4.23 *Evaluation Asset Reputation*

No.	Threat	Countermeasure(s)	L	I	Risk
57	Script Kiddy: penetrates a critical component	Up-to-date software, skilled employees	*Low*	*Medium*	*Low*
58	Skilled Hacker: leaks user data	Hardened network, restrictive data access	*Low*	*Medium*	*Low*

2.4.24 Risk Acceptance

For all medium and high risks we propose additional countermeasures that could be implemented to further reduce the risks.

No. of threat	Proposed additional countermeasure including expected impact
4	Renting a rack space in a data centre would ensure professional access control and certified security standards.
6	Add geographically redundant log and backup server
8	Install intrusion detection system. Migrate database to store secure, salted password hashes that make it computationally expensive to recover the passwords.
9	Renting a rack space in a data centre would ensure professional access control and certified security standards making it harder for a governmental agency to break in.
13	Notify users over an independent path (SMS or unencrypted email) about changes of critical data (password, public key).
16	Pull logs instead of pushing them to prevent that servers can choose to not send their logs and further reduce the attack surface of the log server. Alternatively, implement monitoring to immediately raise an alarm if a host does not send its logs.
18	Prevent that data can be recovered from the cloned disk (e.g. by recovering the symmetric key and decrypting), either by encrypting all data with asymmetrically or enabling disk encryption.
20	Monitor changes to the configuration files, enforce reviews of substantial changes.
24	Migrate legacy database to use more secure, salted password hashes. Use 2FA to make it harder to steal user credentials.
25	Train employees regularly both theoretically and with phishing attack simulations to raise their awareness.
27	Invest in external audits of third-party products to reduce the risk of unknown backdoors
28	Implement policy where the presence of multiple system administrators is required to access confidential key material.
32	Enforce a procedure that multiple system administrators are required to access the backup private key and strictly monitor its usage.
35	Implement monitoring to automatically detect suspicious behaviour of administrators.
36	Raise administrator's awareness to phishing attacks. Implement policy to change SSH keys regularly. Restrict the software running on devices of system administrators.
37	Implement a VPN to allow administrators to access the internal infrastructure without exposing the configuration server.
39	Implement monitoring to check the CRL's availability.
42	Add an additional load balancer to remove the single point of failure that it currently is. Or outsource load balancing to a CDN (since it is not security critical) and benefit from their DoS protection.

References

[1] https://borgbackup.readthedocs.io/en/stable/

[2] seismo.ethz.ch/de/knowledge/seismic-hazard-switzerland/

[3] https://tools.ietf.org/html/rfc6265#section-4.1.2.6

[4] https://i.blackhat.com/us-18/Wed-August-8/us-18-Orange-Tsai-Breaking-Parser-Logic-Take-Your-Path-Normalization-Off-And-Pop-0days-Out-2.pdf