System Description and Risk Analysis

Bähler Alessio[1], Enz Andreas[1], and Niederberger Matthias[1]

Department of Computer Science, ETH Zurich
{*abaehler, aenz, niederbm*}*@student.ethz.ch*

November 22, 2017

Page limit: 30 pages.

Contents

Recall the following guidelines when writing your reports:

- *Adhere to the given templates.*
- *Refer to the security principles in the book for justification.*
- *Use clear terminology:*
  - *secure = confidential + authentic. Be clear about which properties you are writing.*
  - *Are pairwise distinct: certificate, private key, public key, archive to of certificate with private key. Please avoid mixing these up.*
- *Refer to the source document of your risk definitions if appropriate.*
- *For the risk evaluation, formulate the threats in active, not passive, voice: who (threat source) does what (threat action)?*
- *Use a spell checker before hand-in!*

1 System Characterization

1.1 System Overview

iMovies is a company producing independent movies with a focus on investigative reporting. This requires that information within the company and with informants is handled confidentially. Therefore email communication should be secure. The system described in this report implements a certificate authority (CA), that allows employees to download digital certificates created by iMovies. Those can then be used to secure their mail correspondence.

The CA System is reachable from the internet so employees can access it from anywhere and certificates can be managed through a web interface. A network firewall serves as a first layer of defense for the iMovies company networks. The company networks are further divided into the DMZ and the internal network. The DMZ contains only the web server that hosts the CA web application. In the internal network we have a server creating and managing certificates (for short: core_ca), a dedicated database server and a backup machine. See Fig. (?? REF OVERVIEW).

Web traffic is handled on the web server, which in turn gets user data and certificates from the database and core_ca server through a REST API. The backup machine periodically pulls a backup from from firewall, web server, database and core_ca server. All traffic on the network is encrypted.

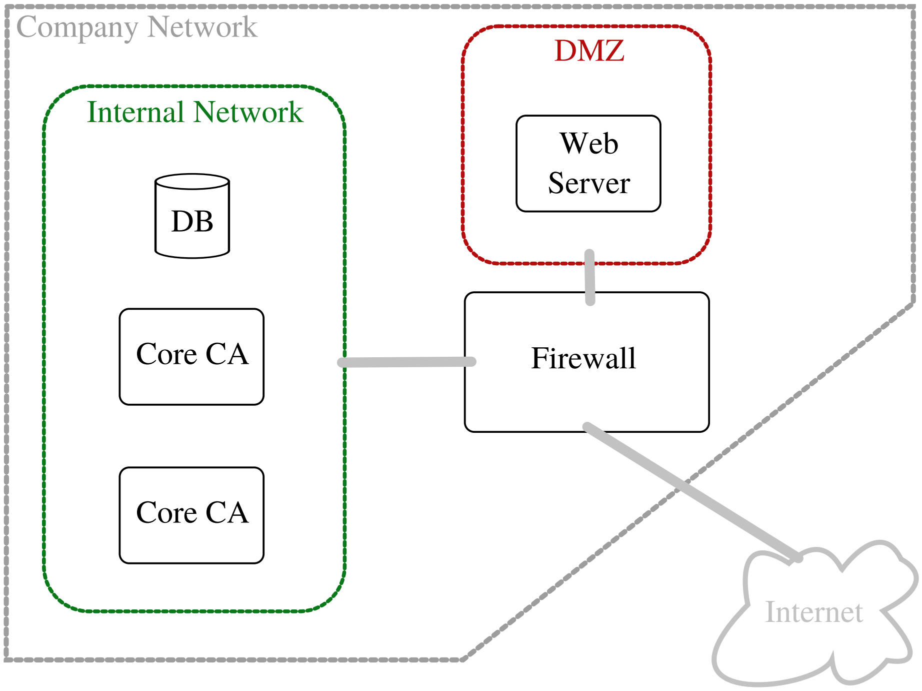

Figure 1: System Architecture of the company network including an external client machine.

1.2 System Functionality

1.2.1 Certificate Issuing Process

- TODO

1.2.2 Certificate Revocation Process

- TODO

1.2.3 CA Administration Interface

Allows CA admins to see:

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

1.2.4 Key Backup

- TODO

1.2.5 System Administration and Maintenance

- TODO

1.3 Security Design

We refer to the following security principles [2]:

1. Simplicity
2. Open Design
3. Compartmentalization
4. Minimum Exposure
5. Least Privilege
6. Minimum Trust and Maximum Trustworthiness
7. Secure, Fail-Safe Defaults
8. Complete Mediation
9. No Single Point of Failure
10. Traceability
11. Generating Secrets
12. Usability

and to the project's security requirements:

a. Access control with regard to the CA functionality and data
b. Secrecy and integrity with respect to the private keys in the key backup
c. Secrecy and integrity with respect to user data
d. Access control on all components

1.3.1 General

- Every process in the different machines runs with only the privileges that are needed to accomplish its task, according to *5. Least Privilege*.

1.3.2 Database

- The MySQL database is accessible only with username:password authentication from localhost, in accord with *5. Least Privilege*, *8. Complete Mediation* and *d. Access control on all components*

- The REST API is reachable only over HTTPS, in accord with *1. Simplicity* and *2. Open Design* since no custom protocol is used, and ideally with client side verification to ensure that only the Webserver can send requests (*Complete Mediation*).

1.3.3 Core CA

- Keys are generated using RSA and are 2048-bit long (*11. Generating Secrets*).
- Thereafter they are deleted as soon as the password protected PKCS#12 file is generated (*4. Minimum Exposure*).
- For backup purposes a copy of the key is encrypted using the public key

1.4 Components

1.4.1 Core Certificate Authority (CA)

The Core CA server runs in the iMovies internal network at IP address 192.168.50.31 and exposes a SparkJava REST API on port 8100, which accepts HTTPS connections only from the Webserver IP address 192.168.51.14 and uses a certificate signed with the CA root key. It offers calls to issue and revoke certificates, as well as to get information about the state of the CA.

The SparkJava application runs under user *coreca* and uses *openssl* commands to manage the CA state. Any data received and sent from the application is in Json format.

The following table shows the available REST calls.

#	**Method and Url**	**Parameters**	**Return**
1	POST /certificates/new/userId	password	pkcs12
2	DELETE /certificates/userId/one	serialNumber	certificateRevocationList
3	DELETE /certificates/userId/all	-	certificateRevocationList
4	GET /ca/issued	-	issued
5	GET /ca/revoked	-	revoked
6	GET /ca/serial_number	-	serialNumber

Description:

1. Creates a new private key and corresponding certificate signed with the CA root key for *userId.* Both are then stored in a PKCS#12 file that can be opened with *password.* The generated private key is encrypted and saved so that it can be backed up, then all other generated data is deleted and the bytes of the PKCS#12 file are returned in *pkcs12*
2. Revokes the certificate with *serialNumber* for *userId* and generates a new certificate revocation list, whose bytes are returned in *certificateRevocationList*
3. Revokes all certificates for *userId* and generates a new certificate revocation list, whose bytes are returned in *certificateRevocationList*
4. Returns the number of issued certificates in *issued*
5. Returns the number of revoked certificates in *revoked*
6. Returns the current serial number in *serialNumber*

TODO: hardening ()

1.4.2 Database

The Database server runs in the iMovies internal network at IP address 192.168.50.33 and exposes a SparkJava REST API on port 8100, which accepts HTTPS connections only from the Webserver IP address 192.168.51.14 and uses a certificate signed with the CA root key. It offers calls to handle user data.
The SparkJava application runs under user *database* and interacts directly with a local MySQL database, which contains only the legacy *users* table. The database is reachable on port 3306, but only from localhost. Any data received and sent from the application is in Json format.
The following table shows the available REST calls.

#	**Method and Url**	**Parameters**	**Return**
1	GET /users/userId	-	lastname, firstname, emailAddress
2	POST /users/userId	lastname, firstname, emailAddress	-
3	POST /users/verify/userId	userPasswordHash	correctCredentials

Description:

1. Returns *lastname*, *firstname* and *emailAddress* attributes for *userId* from the database

2. Changes *userId* attributes in the database to the given *lastname*, *firstname* and *emailAddress*

3. Changes *userId* attributes in the database to the given *lastname*, *firstname* and *emailAddress*

TODO: hardening (1 second wait against brute force,

1.4.3 Backup Machine

The backup machine pulls files from other machines using rsync 3.0.9 in archive mode over an ssh connection. We do full (non-incremental) backups at scheduled intervals of important system logs, applications logs, application configuration and data. Backed up machines are:

- web server
- firewall
- core ca
- database

Not only the last backup is stored, we keep old backups. But to reduce the amount of data stored a cleanup process deletes backups after they reach a certain age. Files can be restored using rsync and reversing source and target of the backup command. While the data in transit is encrypted through the use of ssh, backups on the machine are not encrypted. The machine can only be accessed physically and over ssh with the private key of the sysadmin.

Scheduling of pulling and cleaning the backups is done with cron. There are two main backup frequencies. The first one is a daily pull of seldom changing, less important files. The second one pulls every 20 minutes. Jobs are staggered so that they don't start at the same time. To be able to automate this process over ssh a passwordless private key is needed for the backup user and all machines listed above need to authorize the corresponding public key. The files and folders that need to be pulled have to be listed in configuration files on the backup machine. In general, the pull approach allows central administration of the whole backup process on a single machine.

1.4.4 Network Firewall Machine

- pfsense shell interface - webinterface for pfsense - interfaces: inet, dmz, internal, webinterface

This machine separates the iMovies company networks from the internet and serves as a first line of defense. Furthermore it serves as a router, mainly for incoming webtraffic and ssh connections. We use pfsense 2.4.1 installed on FreeBSD 64-bit. Administration of pfSense is mostly done over a web interface, which is only reachable from the internal network. But remote administration of the web interface is possible by first creating an ssh tunnel to the internal network interface and then starting a web session over this tunnel (using for example Firefox with a SOCKS proxy)

As seen in Fig. (?? REF OVERVIEW) the Firewall has three network interfaces connecting to the internet, DMZ and internal network. In the following we describe the routing (NAT) and firewall rules set up on each interface. All rules we set up do explicitly allow certain traffic, because the pfSense default is to reject everything.

We use static IPs on all machines and network interfaces (see Figure ??? REF TOPO). For the sake of readability we will use the following names for the IP addresses:

WAN port routing table

The only IP exposed to the internet is that of the WAN interface. This means traffic has to be routed to the correct machine using NAT. The only traffic from the internet we we want to allow is https traffic to the web server and ssh traffic to every machine for remote administration. Table 2 shows the routing rules for TCP traffic depending on destination port.

pfSense automatically creates firewall rules to allow NATed traffic. The only rule we add is to allow ICMP traffic to the WAN interface from any host, so that the ping command can be used to check if the WAN interface is reachable.

DMZ firewall rules

Name	IP	Description
WAN	192.168.70.10	Firewall interface connecting to the internet
DMZ	192.168.51.51	Firewall interface connecting to the DMZ
INTERN	192.168.50.50	Firewall interface connecting to the internal network
WS	192.168.51.14	Web server
DB	192.168.50.33	Database server
BK	192.168.50.32	Backup machine
CA	192.168.50.31	Core CA server

Table 1: Names of IPs used in further explanations. See Figure ??? REF OVERVIEW for a graphical representation

Dest. Port	NAT IP	NAT Port
443	WS	8100
5050	INTERN	22
5031	CA	22
5032	BK	22
5033	DB	22
5114	WS	22

Table 2: NAT port routing at the WAN interface

Only connections from the web server to the https ports of database and core_ca are allowed. Also enabling ICMP to be able to ping any host on the company network.

Protocol	Src. IP	Dest. IP	Dest. Port	Action
TCP	WS	DB	8100	Pass
TCP	WS	CA	8100	Pass
ICMP	*	*	*	Pass

Table 3: Firewall rules at the DMZ interface

INTERN firewall rules
Describe the implemented backdoors.

1.5 Backdoors

1.5.1 Easy Backdoor

We put the easy backdoor on the Backup machine. In certain intervals a port is opened for a short time that is directly bound to a shell. Once the attacker has access to the Backup machine he can ssh as root into all other machines from which backup data is pulled. This is possible because the backup process has root access over ssh to all machines that need to be backed up, since it needs to pull system logs readable only by root.

We did this with netcat listening on port 9844 for 10 seconds: ncat -l 9844 -i 10 -v -e /bin/bash The interval scheduling is done with a few cronjobs in the crontab of user backup. To obscure those crontab entries, all their output to stdout and stderr is sent to /dev/null. Also the crontab calls a bash script instead of the netcat command, so that is looks a bit less suspicious. Finally that script is hidden with with a not so easy to find name ". ", which makes it more suspicous in the crontab entry but harder to find in general. The port opens in a pattern every few minutes. The pattern repeats every 10 minutes and during those opens at minute 0, 1, 3, 5, 6, 8.

The connection is persistent if an attacker connects curing those 10 seconds when the port is open.

1.5.2 Hard Backdoor

The hard backdoor is a two-stage process that allows any attacker to execute bash commands with root privileges on the Webserver, Core CA and Database machines. The first phase consists in a hidden webpage on the Webserver and a hidden REST call on both Core CA and Database that, when a given state is reached, allows the execution of any command given by the attacker. Since these commands will be executed with the rights of the unprivileged user running the processes, the second phase consists in using a specially crafted executable that is hidden in the target machine filesystem to obtain passwordless sudo privileges. The attacker can then execute any command through the hidden webpage/REST call and receive its output.
Here a more detailed explanation of the two phases:

- **Phase 1**: TODO
- **Phase 2**: the executable file */usr/lib/systemd/system-agent* has *setuid bit* set and when executed with option *-a* will modify */etc/sudoers* by adding a line that gives the unprivileged user on the machine the right to execute any command without password. If it is executed with option *-z* the original will be written in */etc/sudoers* and any other case will result in no action being performed. Since there aren't many files with *setuid bit* set, the file is placed in a legitimate and pre-existent operating system's directory, is given a misleading name and has creation date set before semester begin to make more difficult its discovery.

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

1.6 Additional Material

You may have additional sections according to your needs.

1.6.1 Login credentials

Machines user accounts		
Machine	**User**	**Password**
Backup	TODO	TODO
Core CA	iadmin	TODO
Core CA	coreca	TODO
Database	iadmin	TODO
Database	database	TODO
Firewall	TODO	TODO
Webserver CA	TODO	TODO

MySQL Database users	
User	**Password**
root	reallySecurePwd1!
dbuser	securePwd17!

iMovies users	
Username	**Password**
db	D15Licz6
fu	KramBamBuli
ms	MidbSvlJ
a3	Astrid

2 Risk Analysis and Security Measures

2.1 Assets

Physical Assets

- Server Machines: physical machine hosting the Web Server Application. Must be available and enable secure and tamper resistant communications with the clients.
- Core CA: physical machine hosting the CA application and the legacy database.
- Backup: physical machine hosting the backup data.
- Internet Connectivity: Modem and lines connecting the WebServer to the Internet.
- Internal Network: LAN via physical lines and a switching modem.

Logical Assets

- Software
  - Web Server Application
  - Core CA Application

  - Legacy MySQL database/application/driver?
  - REST API
  - Backup Daemon
  - Backup Manager
  - Firewall

- Information
  - Certificates
  - Keys
  - User data
  - Configuration files
  - Logs

Persons

- System Administrator: maintains the system by applying software updates, controlling system logs to search malicious behaviours that could lead to security issues and ensuring that the machines hosting the systems components are working properly. He therefore has access to sensitive data, in the form of a remote connection well as physical access to all components.
- CA Administrators: are able to verify the current state of the CA.
- Users: Employees and Informants that both use the system to obtain certificates which allow them to communicate securely with the WebServer.
- Management

Intangible Goods

- Company Reputation
- Confidentiality of informant identities.

2.2 Threat Sources

- Nature: Floods, lightning strikes, earthquakes can damage the physical infrastructure.
- Users: Employees (includes also cleaning personnel etc.) and informants can act maliciously or be careless/poorly trained.
- Competitors: may be interested in obtaining confidential information to gain an advantage, blackmail or cause harm by publishing it. May resort to Skilled Hackers to achieve their goals.

- ”Victims”: subjects of investigative reports that were publicly exposed and may want to get revenge by causing any kind of damage. May resort to Skilled Hackers to achieve their goals.
- Organized Crime: can directly or indirectly be ”Victim”, could be interested in blackmailing the Company to gain money or just to obtain important information that can be sold on the black market/used for other illegal activities.
- Malware: may be non-directional or self-spreading and have different goals, e.g. Ransomware, Trojans.
- Expert Hackers: A skilled hacker has expert knowledge for some systems. He can write his own code and may use unknown or unpublished vulnerabilities (from book). May itself be a ”Victim” or act for monetary interests.
- Script Kiddies: This type of adversary has basic computer knowledge and uses mainly known vulnerabilities for which exploits are available on the Internet. However, he might write scripts to automate tasks or use tools to automatically create malware. His main motivations are challenge, glory and destruction (from book).
- Organizatorial Deficiencies: lack in employee training, poor/non-existing/non-enforced security measures, such as unsanitized user input, can weaken the overall security of the system.
- Hardware Failures

2.3 Risks Definitions

Definition of Likelihood, Impact and Risk level using the following three tables from [2].

Likelihood	Description
High	The threat source is highly motivated and sufficiently capable of exploiting a given vulnerability in order to change the asset’s state. The controls to prevent the vulnerability from being exploited are ineffective.
Medium	The threat source is motivated and capable of exploiting a given vulnerability in order to change the asset’s state, but controls are in place that may impede a successful exploit of the vulnerability.
Low	The threat source lacks motivation or capabilities to exploit a given vulnerability in order to change the asset’s state. Another possibility that results in a low likelihood is the case where controls are in place that prevent (or at least significantly impede) the vulnerability from being exercised.

Impact	
Impact	Description
High	The event (1) may result in a highly costly loss of major tangible assets or resources; (2) may significantly violate, harm, or impede an organization's mission, reputation, or interest; or (3) may result in human death or serious injury.
Medium	The event (1) may result in a costly loss of tangible assets or resources; (2) may violate, harm, or impede an organization's mission, reputation, or interest, or (3) may result in human injury.
Low	The event (1) may result in a loss of some tangible assets or resources or (2) may noticeably affect an organization's mission, reputation, or inter- est.

Risk Level			
Likelihood	**Impact**		
	Low	Med	High
High	Low	Med	High
Med	Low	Med	Med
Low	Low	Low	Low

2.4 Risk Evaluation

Potential threats and countermeasures with the inferred risk.

2.4.1 *Evaluation Web Server*

No.	Threat	Countermeasure(s)	L	I	Risk
1	Expert Hackers: mount MitM attack to spy on and tamper with communications between Clients and WebServer. This allows the hackers to learn in particular a user's password and private keys.	HTTPs connection with Server side authentication	*Med*	*High*	*Med*
2	Victim: resorts to Script Kiddies to launch DDoS attack on WebServer and cause damage, disruptions, maybe even ask money to stop	Simple DDoS protection like SYN cookies against syn flood	*High*	*High*	*High*

2.4.2 *Evaluation Core CA*

No.	Threat	Countermeasure(s)	L	I	Risk
1	Hardware Failures: cause damages to the hard drives and privite CA key and certificate can't be recovered.		*Low*	*Med*	*Low*

2.4.3 *Evaluation Backup*

No.	Threat	Countermeasure(s)	L	I	Risk
1	User: exploits physical access to Backup Machine and obtains backup data.	Physical protection of System Components, Disk Encryption	*Low*	*Med*	*Low*

2.4.4 *Evaluation System Administrator*

No.	Threat	Countermeasure(s)	L	I	Risk
1	Expert Hacker: steals System Administrator credentials	Enforce Strong Passwords, Increase security sensibilization/awareness	*Med*	*High*	*Med*
2	Organizatorial Deficiencies: illness or injury impede its work and the System is left unattended in case of problems/attacks	Good Documentation and making sure that not only one person knows the system	*High*	*Med*	*Med*

References

[1] Computer Security: Principles and Practice. William Stallings and Laurie Brown, Prentice Hall, 2008

[2] Applied Information Security: A Hands-on Approach, David Basin, Patrick Schaller and Michael Schläpfer, Springer, 2011