System Description and Risk Analysis

Bähler Alessio   Enz Andreas   Niederberger Matthias

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

The aim of this system is to provide the customer company with an in-house certificate authority (CA). This CA provides employees with digital certificates on demand, which are used to secure email communication. The System consists of three machines in a company network and external client machines that connect over the Internet. Inside the company network we have Machine 1 housing the Core CA functionality and the legacy MySQL database. This means that the main signing key and certificate revocation list (CLR) are stored and maintained on Machine 1. Machine 2 contains the web server with a firewall to shield it and the company network because any traffic from the Internet will have to cross the web server machine anyway. Finally Machine 3 is used for the physical separation of the backup service, with backup daemons connecting to the other two machines.

1.2 System Functionality

1.2.1 Certificate Issuing Process

- TODO

1.2.2 Certificate Revocation Process

- TODO

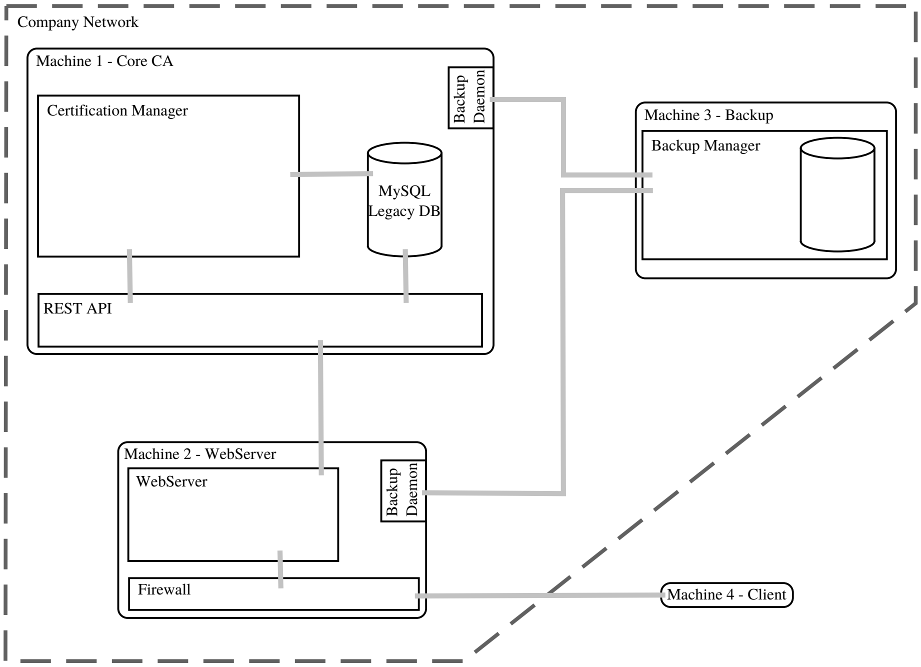

Figure 1: System Architecture of the company network including an external client machine.

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

Describe the system's security design, including access control, key and session management, and security of data at rest and in transit.

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. 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. TODO

TODO: hardening (1 second wait against brute force, 3306 localhost only accessible)
A short description of the components in Figure 1.

- **Certification Manager**: Manages certificate state (creation, revocation, deletion, ...). Interfaces with the Web Server over the REST API and directly with the legacy MySQL database. It has two main subcomponents:
  - Certification Store: A directory where keys and certificates are stored.
  - Certification Generator: Built with OpenSSL
- **MySQL DB**: As provided. Interfaces with the web server over a REST API and with the Certification Manager.
- **REST API**: Interface between Core CA machine and WebServer machine.
- **Web Server**: Accepts web traffic filtered through a firewall. Does authorization by checking legacy database and can request certificate state changes from the Certification Manager.
- **Firewall**: Filters traffic.
- **Backup Manager**: Periodically stores specified data in the backup database. Interfaces with Core CA and Web Server machine.
  - Backup Daemon: Sends data to backup machine

Describe the implemented backdoors.

1.5 Backdoors

1.5.1 Easy Backdoor

TODO: Andi

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

- Web Server: 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 Schlpfer, Springer, 2011