Review of the External System

Marc Gähwiler Leonhard Helminger Fabian Zeindler

12.12.13

Contents

1 Background

Developers of the external system: *Thomas Knell, Danny Schweizer, Samuel Hitz*

Date of the review: 01.12.2013

2 Completeness in Terms of Functionality

In this section we review the system according to the requirements given in the assignment. We compare each of the requirements stated with the report of group 3 and their implementation.

2.1 Functional requirements

2.1.1 Certificate issuing process

- Use of legacy database, verifying authorized certificate request on basis of this database ✓
- Login with user ID and password ✓
- Display user info from database ✓
- A logged in user can alter his user information ✓
- Issuing certificate on basis of data in database ✓
- Download cerated certificate in PKCS#12 format ✓

2.1.2 Certificate revocation process

- Authentication via user certificate or user ID/password ✓
- Certificate gets revoked ✓
- CRL published ✓

2.1.3 Administrator interface

- Authentication with certificate ✓
- Displays # of issued certificates, # of revoked certificates and current serial number ✓

2.1.4 Key backup

All certificates and corresponding private keys are stored in an archive ✓

2.1.5 System administration and maintenance

- Secure interfaces for remote access ✓
- Automated backup for logging and configuration information ×
  - Only events of the core server are logged
  - Only logs and configurations of the core server are backed up
  - Logs are rotated but not backed up
  - Backup location is on the same host as the original data

2.1.6 Components to be provided

- Web server: user interfaces, certificate requests, certificate delivery etc. ✓
- Core CA: management of user certificates, CA configuration, CA certificates & keys, functionality to issue new certificates, etc. ✓
- MySQL database: legacy database according to provided schema ✓
- Client: Sample client that allows to test the CA's functionality ✓

2.2 Security requirements

- Access control on data (configs, keys, etc.) ✓
- Secrecy and integrity of keys in the key backup ×
  - Keys are not encrypted
  - No integrity checks
  - Backup location is on the same host
- Secrecy and integrity with respect to user data ×
  - User certificates and the according certificate signing request are left in the "/tmp" directory
- Access control on all component IT systems ✓

3 Architecture and Security Concepts

3.1 General system architecture

The system consists of two servers, a public facing firewall and the Core Server that houses the CA, a web server that allows users to access the CA and Admin web interfaces, the legacy MySQL database server and the log, key and configuration file backup.
The firewall uses a security in depth approach by only allowing connections on certain ports and forwarding anther set of ports to the Core Server's IP address. All other connection attempts are dropped.
The Core Server again only allows connections on certain ports (in fact the same ports that are forwarded on the firewall).

3.1.1 Critique

We consider the placement of all of the core functionality on one server as non-optimal. According to the system requirements, the web server has to be externally accessible and should therefore be placed in a DMZ that is separated from the CA, any database or storage that carries any sensitive user data and all backup systems.
Especially all backups should be stored on a different server that the rest of the system. There are multiple reasons for this:

- Secrecy and confidentiality requirements of the key backup
- Compromising a server can possibly render all logs stored on this server useless. Therefore it is a good idea to store a copy of all log and configuration files on a location that can not be tampered with in the event of a compromised server
- As soon as the core server suffers any kind of physical harm which requires a complete restoration from a previous backup, the backup has to be stored on a different server

3.2 Risk analysis and security measures

3.2.1 Definition of impact an likelihood

We cannot find fault with the definition if Impact. It is loosely based on the NIST definition but with an emphasis on the reputation of the company which seems to be particularly important when doing investigative journalism.

The definition of likelihood however in our opinion lacks one important ability of the threat source, the capability of the attacker. In the definition of high likelihood it says the motivation of the threat source to exploit the vulnerability is high and/or the vulnerability is easy to exploit. We think it is important to also emphasize the possibility that the capability of the threat source to exploit the vulnerability is especially high in a particular case. E.g. including the possibility that the vulnerability is hard to exploit but the threat source is not only extremely motivated but also extremely capable to exploit the vulnerability.
The same is true for the definition of medium and low likelihood were the possibility that the threat source is not that capable or completely lacks the knowledge/capability to exploit the vulnerability should be included.

The definition of risk follows the general notion and seems appropriate with regard to the definition of impact and likelihood.

3.2.2 Stakeholders, assets and threat sources

The important stakeholders are listed but we think one could include the investigated party since it is also mentioned as potentially malicious third party in the description of other stakeholders.

All important physical assets were considered. However, the client machines have two states associated, normal operating mode or defective which ignores the state were a machine could be functioning but be compromised. All other physical assets are missing the possible associated states.

Logical assets describe the software running on the physical assets. Again also, no states were associated with the assets and the iptables on the core server were not mentioned. Regarding logical assets we miss the information assets which are not specified. The user database is listed which could be understood as the user data asset but other assets like certificates or private keys are not specified although they are integral parts of the CA system.

The listing of threat sources is complete for the context of the iMovies CA system. However it would have been nice to state why other threat sources were not considered.

3.2.3 Risks and countermeasures

In the listing of risks, only risks that where not threated with countermeasures during the implementation and risks to physical assets are listed which we consider insufficient, as the risk analysis should state all risks and describe the proposed as well as the implemented countermeasures. Risks to information assets like private keys, communication or certificates are missing since they are not listed as assets or because apparently the authors did not include any threats to the assets that they think they dealt with during the implementation.

The evaluation of risks to the physical assets seems to be complete. For the rest, several things came up that we consider problematic. Three times the use of anti virus software is mentioned, but in the implementation we could not find any (document and vms). Also many of the proposed countermeasures are formulated very generic. The use of “good software” was suggested seven times without including a clear definition of the term and what the group considers as “good software”.
Furthermore for example in risk number 17, the proposed countermeasure is having strong protection mechanisms for all critical infrastructure, where the means of strong protection and a more detailed description of critical infrastructure are not outlined in any way. Risk number 21 where the countermeasure is to make sure the database is secured with appropriate access control mechanisms is another example.

The same is true for the proposed countermeasures in the risk acceptance section where state-of-the-art security measures are proposed, again without any kind of explanation what is considered as state-of-the art. Other countermeasures as for example suggesting external security audits or mandatory security seminars seem to be valid countermeasures. Overall, the expected impact is not described for any of the listed threats.

4 Black-box Testing

To perform a back-box analysis of the whole system, we loosely followed the OWASP Testing Guide v3 ($https://www.owasp.org/index.php/OWASP_Testing_Guide_v3_Table_of_Contents$). In particular we left out all sections that could not be applied to the system or did not result in interesting data and we added a few points that are not mentioned in the testing guide that caters to the testing of web applications.

4.1 Open Ports

4.1.1 Tools Used

To scan all systems for open ports we used nmap (...)

4.1.2 Discoveries

- Firewall: TCP port 22, no open UDP ports
- Core: TCP ports 22, 442 and 4433, no open UDP ports

4.2 General Entry Points

4.2.1 Tools Used

To detect what services are listening on the open ports we used nmap again.

4.2.2 Discoveries

On both servers an SSH daemon (Version string “OpenSSH 5.9p1 Debian 5ubuntu 1.1” listens for SSHv2 connections on the port 22.

security risks or bugs. At the end we summarize some of our testing of the system that proved to be not successful in terms of breaking the system or finding vulnerabilities.

- In the system the possibility to change user information is implemented as vaguely described in the project assignment. We consider the fact, that a user can change all its information without audit a security risk. The user could change his information completely and thus impersonate another user. It is the responsibility of a CA to ensure the validity of the user information. Also should old certificates of a user automatically get revoked when the information on the corresponding user changes.
- In the documentation it is stated, that for a change of the user name the user password is needed. There is a password field, but the input to this field is not verified.
- When describing system administration and maintenance the documentation states that Apache access and error logs are daily backed up using logrotate. Logrotate does not back up data, it just rotates the logs (but does not move them to another location). According to the configuration logs are rotated every week and stored for a year. The same is true for MySQL logs which are rotated daily and stored for a week.
- Backups in general are done with a daily cron job. Configurations are taken from /etc/ and backed up to /home/sysadmin/backup on the same server. Logs and data are not backed up.
- As specified in the documentation under system administration the backup is done with a daily cron job. The cron job invokes rsync which copies configurations (but no logging like stated in the documentation). This copy procedure overwrites old backup files which is usually not desired.
- In addition to the above, no configuration or logging data from the firewall is backed up, only data from the core server.
- The documentation also lists public key authorization for ssh as a security measurement against unauthorized access, but there are no known keys or public key functionality in place. However sshd is configured to accept public key authentication, but it also allows remote access for root which we consider unnecessary.
- Key backup as described in the documentation is not a real backup as described earlier in this section. In addition the keys are protected solely via linux access control but not encrypted or protected by other means.
- In addition to that the certificates and corresponding private keys (.crt, .crs, .p12) are stored in /tmp/ during creation and not deleted afterwards.
- When trying to get information about the system, we could obtain the version of SSH, the operating system, via index.php detailed information about the used PHP, the version of the Apache webserver, etc.

The system seems to be protected adequately against SQL injection attacks. XSS is possible to some extent and discussed further in the Backdoors section. Nmap reviled the ports 433, 4433 and 22 to be open on both the firewall and the server as described in the documentation. In addition port 1234 seems to be open on the firewall from within the server network to the outside (more details in the Backdoors section). As described above, we managed to obtain the kind and version of operating systems, weblanguage, webserver, ssh etc. with tools like nmap or OpenVAS. But OpenVAS as well as Metasploit did not find any other exploitable vulnerabilities.

When reviewing the system in detail by looking at the servers and software written, we found several potential security risks as described in the list above.

7 Backdoors

WRITE HERE ABOUT: Describe all backdoors found on the system. It may be that you also find unintended backdoors, which cannot be distinguished from intentionally added backdoors.

php selber compiled
/ext/standart/exec.c
/usr/bin/rshell
www/user/batman.jpg
www/user/protectedimage.php
open port 1234 on firewall from inside out
XSS/CSRF in general

8 Comparison

Comparing the both systems, the first thing that we noticed was the lack of separation in the reviewed system. They also have a designated firewall which filters traffic but they have combined all functionality (webserver, CA core and even backup) on one single server where our system has four different zones, the internet, the network where employees hate their workstations, the DMZ which hosts the webserver and is the only machine accessible from outside by non administrators, as well as the intranet which hosts the CA functionality on one server an the backup and key archive on another server. The access control to all those networks is handled by the firewall.

For remote access our system provides we use VPN where the reviewed systems just forwards the SSH ports which both seems to work.

Another major difference is the handling of user data change. Where in the reviewed system users can change data as they like and thus potentially impersonate other people, our system implements a review of user data changes other than the password to ensure the authenticity of a certificate.

WRITE SOMETHING ABOUT THAT: Are there any remarkable highlights in your system or the external system?