Lab Project,
Final report

Group 5
Magnus Krane, Erik Sørensen, Piyush Bajpayee

{eriksore, magnkr, piyushb}@stud.ntnu.no
TTM4135 Information Security
March 11, 2013

Abstract

This lab report is a term assignment in the course TTM4135 at NTNU.

1 Introduction

The web could be argued to be one of the most important infrastructures in the world.

2 Experimental procedure

During the experiment we followed [1] point by point, therefore only questions asked in the text (other than Q-questions) and points were we changed procedure will be mentioned in this part.

Part 1.4 The purpose of the 'echo' and 'touch' commands were to create a "database" for the certificates.
In the *caconf.cnf* file the default-md and policy-match variables were changed. default-md were set to SHA1, which is more secure than md5 [3], and more matches were set in policy-match. These steps were done to make the certificate more secure.

3 Results

4 Discussion

4.1 Q1

Comment on security related issues regarding the cryptographic algorithms used to generate and sign your groups web server certicate (key length, algorithm, etc.).

For the web server certificate, the signature algorithm were chosen as SHA1 with RSA encryption, with a public-key length of 2048 bits. At the current time, RSA laboratories (creators of the RSA algorithm) recommends a 2048 bits key size for extremely valuable keys, and 1024 bits key size for corporate use [2]. Even 768 bits key might be regarded as sufficient. The use of 2048 bits key size in this case could therefore be regarded as 'over-the-top'. Ultimately the choice of key length is a trade-off between de-/encryption speed and security.
SHA1 (160-bit hash) were chosen over Md5 (128-bit hash) as message digest due to known security problems with Md5 [3]

4.2 Q2

The detached PGP signature of the source code should be verified using GnuPG.

Explain what you have achieved through each of these verications. What is the name of the person signing the Apache release?

Through these step we have identified that a "Jim Jagielski" has signed the file, but we can still not trust the file as it is not certified with a trusted signature. A check against the pgpkeys.mit.edu keyserver also shows this. In conclusion, the received public key can't be trusted. To validate the authenticity of a key (and therefore also, the integrity of the file) [4] recomends a face-to-face validation with the signee.

4.3 Q3

What are the access permissions to your web servers conguration files, server certicate and the corresponding private key? Comment on possible attacks to your web server due to inappropriate file permissions.

The following access permissions have been set.

- Configuration files: 600, -rw- — —
- Server certificate: 400, -r– — —
- Private key: 400, -r– — —

These permissions has been set as restrictive as possible. Especially for the certicate and key files, their persmissions have been set to read only. This is to make sure their integrity is kept. All files have been given read/write access for owner only, i.e. you would need to be user root/gr05 to read these files. The files is also stored in folders not accesible from the Internet.

Possible attacks could be to change certificate requirements in the configuration file to gain access, or if the private key were to be accessed the whole certificate access control would be rendered useless.

4.4 Q4

Web servers offering weak cryptography are subject to several attacks. What kind of attacks are feasible? How did you congure your server to prevent such attacks?

4.5 Q5

What kind of malicious attacks is your web application (PHP) vulnerable to? Describe them briey, and point out what countermeasures you have developed in your code to prevent such attacks.

Conclusion

References

[1] ITEM, *Part 3 - Design and implementation of a resource allocation service*. NTNU, 2013. http://www.item.ntnu.no/fag/ttm4120/current/lab/lab3spread.pdf, retrieved Feb 27th 2013.

[2] RSA Laboratories, *How large a key should be used in the RSA cryptosystem?* http://www.rsa.com/rsalabs/node.asp?id=2218, retrieved Mar 8th 2013.

[3] IETF.org, *Updated Security Considerations for the MD5 Message-Digest and the HMAC-MD5 Algorithms* http://tools.ietf.org/html/rfc6151, downloaded Mar 9th 2013.

[4] Apache.org, *Verifying Apache HTTP Server Releases* http://httpd.apache.org/dev/verification.html, retrieved Mar 10th 2013.

[5] William Stallings, *Cryptography and Network Security - Principle and Practice*, fifth edition, Prentice Hall, 2011.

List of Figures