TTM4135 Project

Norwegian University of Science and Technology

Department of Telematics

Pavel Arteev
Marius Münch
Bjørn Tungesvik

1 Introduction

2 Results

2.1 Part 1: Certificate authority

As result of part 1, we generated personal certificates for the server. Then established the group CA in the certificate hierarchy of NTNU and generated the certificate for the apache web server.

Q1. Comment on security related issues regarding the cryptographic algorithms used to generate and sign your groups web server certificate (key length, algorithm, etc.).

The SSL certificate has a 2048-bit length private key. This is the same as the recommended minimum key length advised by NIST for asymmetric encryption. This key length is expected to be sufficient until 2030. [1]This implies a key length that is sufficient for securing the certificates in this assignments.

Unfortunately, we used the default message digest algorithm, MD5, to sign our certificates. MD5 offers a short hash value making it vulnerable to collisions. Previous research has shown that is possible to produce collisions with relatively ease. A group of researchers described how on could create a pair of files having the same MD5 hash. Later a group of researchers shown, with using this technique, how one could generate a fraudulent SSL certificate derived from a valid one. They showed that the certificate will be accepted as valid. Due these advances, the MD5 is considered cryptographic broken and unsuitable for further use. We need to emphasize that no wild attacks has been reported using this technique, but it should still be avoided. [4, 3]

Further analysis of the web page also reveals that we support more than we should. The site will accept weak cipher suits. Note that during the SSL handshake, a cipher suite that is appropriate for both client and server is chosen. In our case we allow the client to choose an cipher suite with encryption lower than 128 bits, which is consider in secure. Normally modern browsers don't support weak ciphers, but on our site hackers could force a lower encryption session. However, we should emphasize that the risk is minimal since the browsers won't support weak ciphers. [2]

The fix is relatively issue though. We just modified the configuration files to support higher cipher suits.

2.2 Part 2: Access control and Apache

Q2. Explain what you have achieved through each of these verifications. What is the name of the person signing the Apache release?

Q3. What are the access permissions to your web server's configuration files, server certificate and the corresponding private key? Comment on possible attacks to your web server due to inappropriate file permissions.

Q4. Web servers offering weak cryptography are subject to several attacks. What kind of attacks are feasible? How did you configure your server to prevent such attacks?

2.3 Part 3: Writing PHP application

Q5. What kind of malicious attacks is your web application (PHP) vulnerable to? Describe them brifly, and point out what countermeasures you have developed in your code to prevent such attacks.

2.4 Part 4: Setting up a subversion repository

Completing part 4, the web server had a fully operational SVN repository with authentication mechanisms

Q6. Describe the security measures you have undertaken to secure your repository, and how did that affect the security of your Web Application (Better? Worse?).

Using the apache servers built-in authentication methods, the repository is only accessible to users that can provide valid usernames and passwords. The repository is also only accessible over an https connection, which protects the integrity of the communication.

3 Discussion

The lab work introduced many tools and techniques to secure the web server. In this section we will discuss the most critical ones in order to achieve desired security.

3.1 Certificates and SSL

The resulting web server relies heavily on the SSL protocol and the x.509 standard in order to provide access control and authentication. As seen, in order for SSL certificates to provide sufficient security, on most make sure that we use sufficient key length when generating keys. It is also important to make sure that recommended algorithms are used during the generation part.

3.2 SVN repository

4 Conclusion

In future work we should be a little more aware of the algorithms we are using. Even though we are using state of the art tools in order to achieve wanted protection, the tools also may support legacy encryptions standards. We need to make sure this tools are properly configured to make sure that we provide the security intended. The problem with the MD5 algorithm is an example of security weaknesses that should be avoided. Next time we should use a stronger digest algorithm like sha-512 for instance.

References

[1] Nist recommendation. Online, accessed 16 - march - 2013.

[2] Networking4All. Cipher suit. Online, accessed 16 - march - 2013.

[3] Networking4All. Weaknesses in ssl certificates with a md5 signature algorithm. Online, accessed 16 - march - 2013.

[4] Wikipedia. Md5, 2013. Online, accessed 16 - march - 2013.