Cipher Suite Rollback: A Misuse Pattern for the SSL/TLS Client/Server Authentication Handshake Protocol

ALI ALKAZIMI, Florida Atlantic University
EDUARDO B. FERNANDEZ, Florida Atlantic University

Transport Layer Security (TLS) is a cryptographic protocol that provides a secure channel between a client and a server. TLS is the successor to the Secure Sockets Layer (SSL) protocol. The secure connection prevents an attacker from eavesdropping an established client-server connection. It is used in most internet communications for enabling secure web browsing. The SSL/TLS security protocol is layered between the application protocol layer and the TCP/IP layer and includes as one of its sub-protocols the handshake protocol. We present here a misuse pattern for the SSL/TLS Handshake Protocol: the Cipher Suite Rollback, where the attacker intercepts the ”ClientHello” message, replaces the CipherSuite (a list of encryption algorithms), with a weak or NULL-Cipher, and passes the intercepted message to the server which will use now a weaker cipher, allowing the attacker to gain access to the exchanged data between the client and the server.

Categories and Subject Descriptors: D.2.11 **[Software Engineering]** Software Architectures–Patterns; D.5.1 D.4.6 **[Security and Protection]** Authentication

General Terms: Design

Keywords: TLS, TLS Handshake Protocol, misuse patterns, security patterns.

1. INTRODUCTION

The Transport Layer of the Internet protocol uses connection-oriented protocols that provide secure channels between clients and servers to protect their interactions. In this paper, we present a misuse pattern for the SSL/TLS Handshake protocol which is a sub protocol in the SSL/TLS layer (Figure 1). The SSL/TLS layer is divided into two layers: The Handshake layer which contains three sub protocols: Handshake Protocol, Change Cipher Specification Protocol, and the Alert Protocol. The second layer is the Record Layer which sends secure data to/from the application (Microsoft 2003). We concentrate here only on the Handshake Protocol which is used to negotiate the session parameters between the client and the server and we introduce a misuse pattern related to this protocol. Misuse patterns illustrate how a type of information misuse is performed, analyze the best available methods for stopping the attack, and describe how to trace the attack once it has happened

The Cipher Suite Rollback describes how this attack is performed from the viewpoint of the attacker. The Cipher Suite is a list of symmetric and asymmetric encryption algorithms that can be used by hosts to establish secure communication. In the attack, the attacker follows a man-in-the-middle approach to intercept the ”ClientHello” message, replaces its CipherSuite with a weak or NULL-Cipher, and passes the intercepted message to the server, which will use now a weaker cipher, allowing the attacker to gain access to the exchanged data between the client and the server.

It is important for system designers and network administrators to know how vulnerabilities in a system can be exploited .We assume that the readers are familiar with the POSA template and the network concepts that are used to describe these patterns (Buschmann et al. 1996, Schumacher et al. 2006).This pattern could be of value to network administrators, users, testers, and researchers.

Authors’ addresses: Ali Alkazimi , Dept. of Computer and Electrical Eng. and Computer Science, Florida Atlantic University, 777 Glades Rd., Boca Raton, FL33431, USA; email: aalkazi@fau.edu ; Eduardo B. Fernandez (corresponding author), Dept. of Computer and Electrical Eng. and Computer Science, Florida Atlantic University, 777 Glades Rd., Boca Raton, FL33431, USA; email: ed@cse.fau.edu

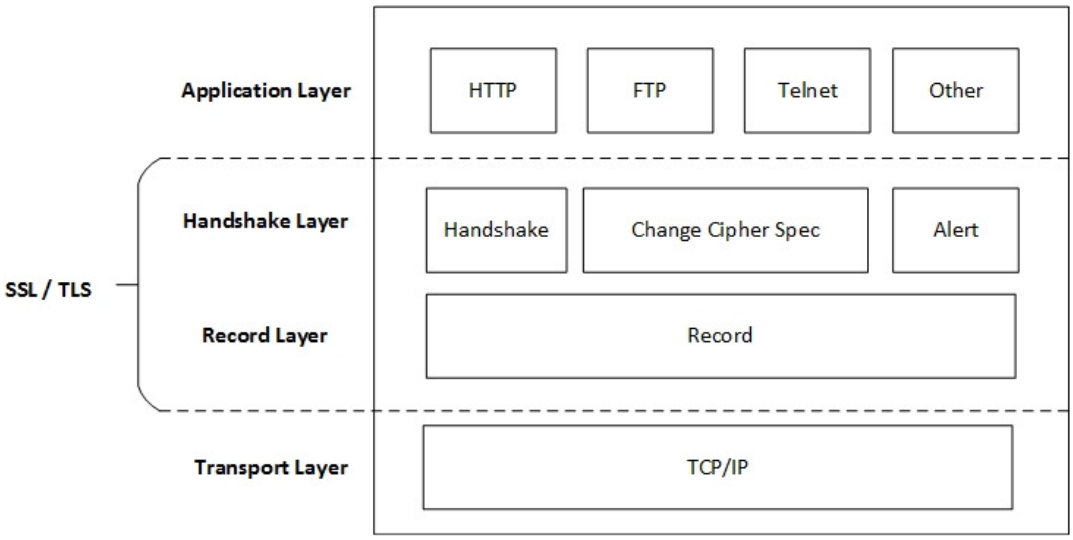

Fig.1. SSL/TLS Protocol Layers

2. CIPHER SUITE ROLLBACK

Intent

In the Cipher Suite Rollback the attacker has the goal of eavesdropping the communication between a client and a server. To achieve this goal, the attacker impersonates the client from the point of view of the server, and the server from the point of view of the client during the initial security negotiation in order to force that the communication between them uses weak security or no security at all. Specifically, the attacker intercepts the ”ClientHelloMessage”, replaces the CipherSuite with a weak or NULL-Cipher, and passes the intercepted message to the server. The server accepts the weak CipherSuite from the attacker and thinks that it is establishing a secure connection with the intended client, which lets the attacker eavesdrop on it. The attacker can now gain access to the exchanged data between the client and the server.

Context

This client/server authentication handshake can be used on TLS clients and servers that support Secure Sockets Layer (SSL) version 2.0 (IETF 2011). Also, it can be used on Internet browsers that are set to use the TLS 1.0 or TLS 1.1 by default instead of using the more secure TLS 1.2, which makes them vulnerable to the rollback attack. This problem affects SSL and TLS implementations previous to the 2011 specifications, when the use of SSL 2.0 was prohibited (IETF 2011),

The SSL/TLS Handshake protocol is an Authenticated Key Exchange (AKE) protocol for negotiating cryptographic information and algorithms which authenticate the identities of the parties involved in the key exchange (Meyer and Schwenk 2013, Kumar and Fernandez 2012). Figure 2 shows the class diagram of the components involved in normal communication in the SSL/TLS Server Authentication handshake. This is a basic handshake procedure where a certificate is required to be sent by the client to the participating server during the connection. The **SSL/TLS Authority Service** generates certificates which are assigned to **Clients** and **Server**s. Certificates can use **Public Keys** or **Symmetric Keys**.

The client sends first a ClientHelloMessage that contains a RandomNumber, the CipherSuite, the ClientTLSVersion and the CompressionMethod to the server (Figure 3). When the server receives this message, it responds with a ServerHelloMessage and exchanges the RandomNumber with the CipherSuite. Then it sends the CompressionMethod and its SSL certificate followed by the ServerDoneMessage. The client replies with a key exchange message for establishing the communication session, “KeyMaterial” in Figure 2, and begins computing the master secret along

with the server. The ChangeCipherSpec message is then sent by the client to indicate that the future messages will be encrypted and authenticated. Finally, the client sends the DoneMessage which the server will decrypt and verify. The server sends the ChangeCipherSpec and the DoneMessage to the client, who has the identical decryption and verification functions as the server. Finally the secure messages between the client and the server start flowing (Rahm 2014).

Problem

If the attacker can manage to intercept these messages, he can communicate with the server impersonating the client and get valuable information (such as client's information, account number, social security number, etc.). The attacker can also attach a malicious code that could be harmful to the server. The initial handshake between the client and the server is not protected and this is the main reason that the client sends the "Change Cipher Spec" message to the server before finishing the handshake. This message is also not protected. The attacker can modify the exchanged Change Cipher Spec message to stop client and server from updating it and then intercept the connection (Zhang, H.L. 2003)

The attack can take advantage of the following vulnerabilities:

- The system could be using an outdated security policy for its SSL/TLS which makes it vulnerable to this and several other attacks.
- The data exchanged between the server and the client for their initial contact can be read and modified. This step does not check the credentials of source or destination.
- It is possible that the server accepts the attacker's weak or Null Cipher Suite list and reply to it in order to establish the Handshake.The resulting damage will depend on the access the attacker gains from the server authentication
- The exchanged data might not be encrypted and the exchanged data of the attacked server might be stolen by using special software (Pelaez et al. 2007).
- There are still a large number of personal computers that are running outdated versions of the TLS protocols that could use the SSL 2.0 handshake. (Cluley 2014) indicates that by early 2014 about 27 % of all Windows OSs are XP. Even if inflated, this figure is significant. These computers are not receiving security updates, which means that an attacker has plenty of opportunities.

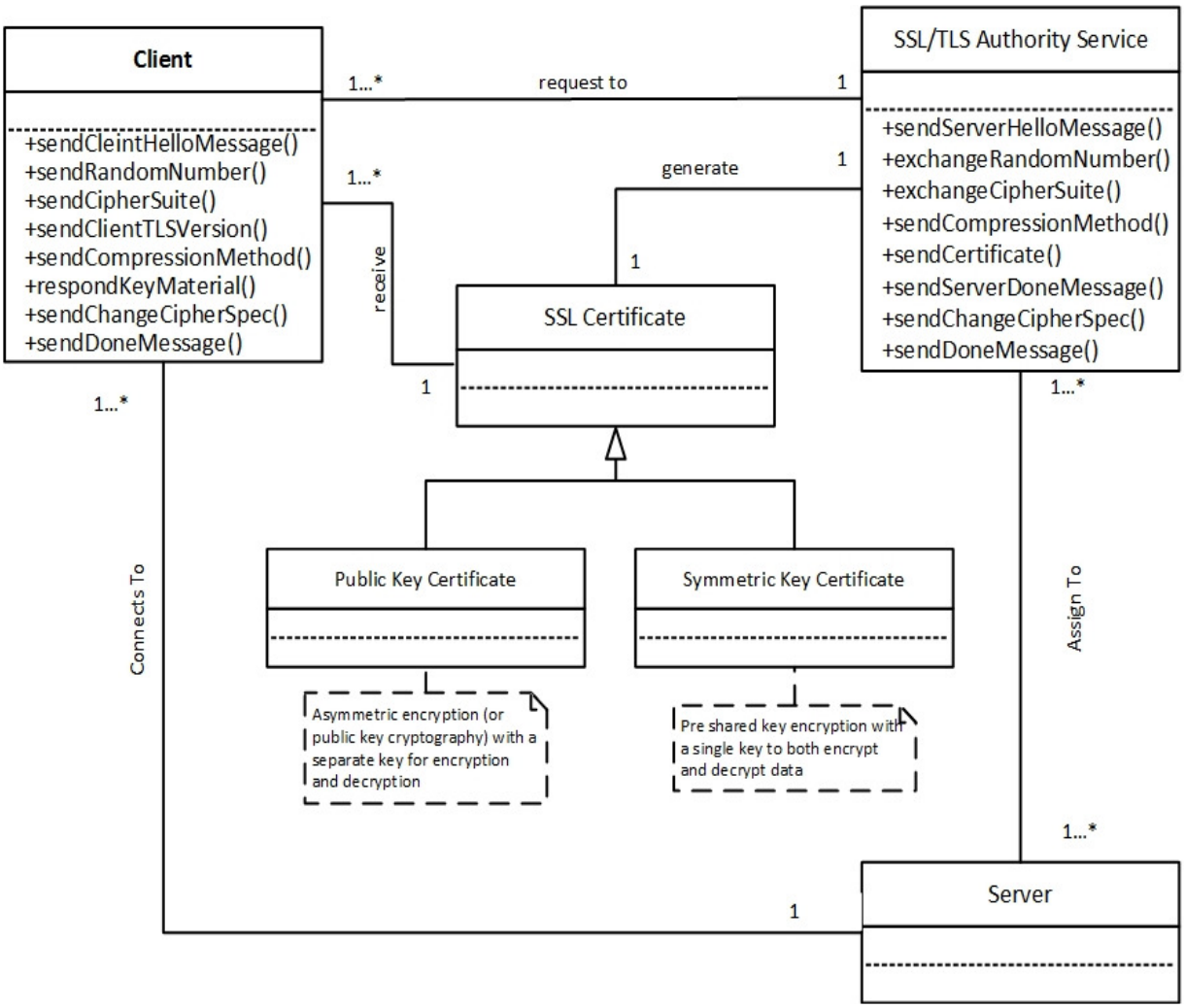

Fig. **2**. Class Diagram for SSL/TLS Server Handshake

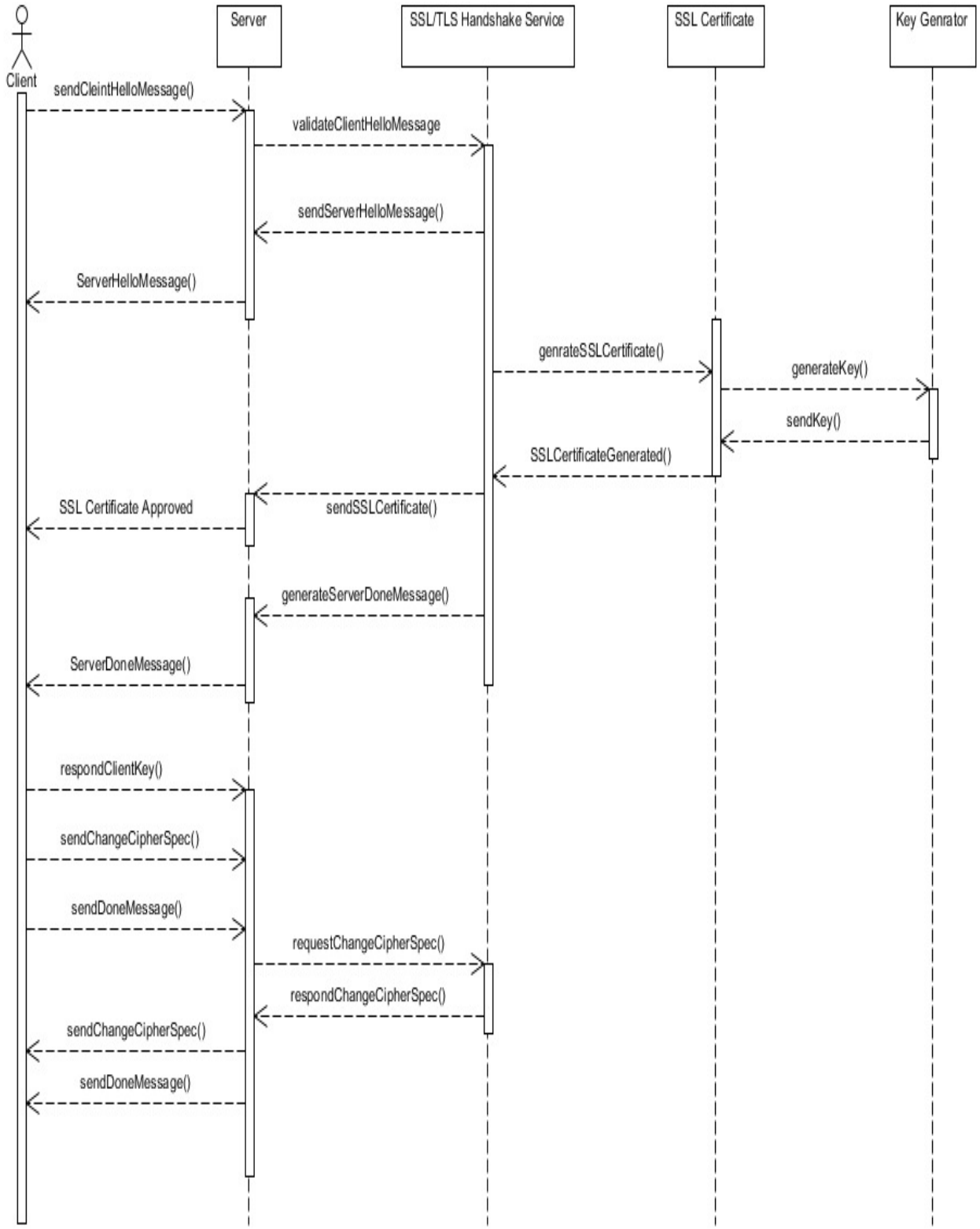

Fig. 3. Sequence Diagram for Server SSL/TLS Handshake

Solution

The SSL/TLS handshake begins by exchanging the ClientHelloMessage between the client and the server. This message contains the client's CipherSuite that has a list of cryptographic algorithms. The server receives this list and then a secure connection can begin. They communicate using the common Cipher Suite based on the server's selection that offers the highest level of security. . This procedure can be exposed to several attacks when the message is intercepted. When a Cipher Suite Rollback occurs in this setting, an attacker edits the list of the Cipher Suite in the ClientHelloMessage to force both endpoints to use a weaker form of encryption than they would choose (Zhang, H.L. 2003).

Figure 4 shows the class diagram of the Cipher Suite Rollback. It is performed on the client/server handshake protocol by intercepting the upcoming secure connection. The **Attacker** is represented as a class to show its connections; it is a type of client. As mentioned previously, this attack is feasible only in old implementations of the TLS 1.0 or TLS 1.1 that support the SSL 2.0 negotiation. Supporting SSL 2.0 was prohibited in March 2011, but still today many of these outdated implementations exist.

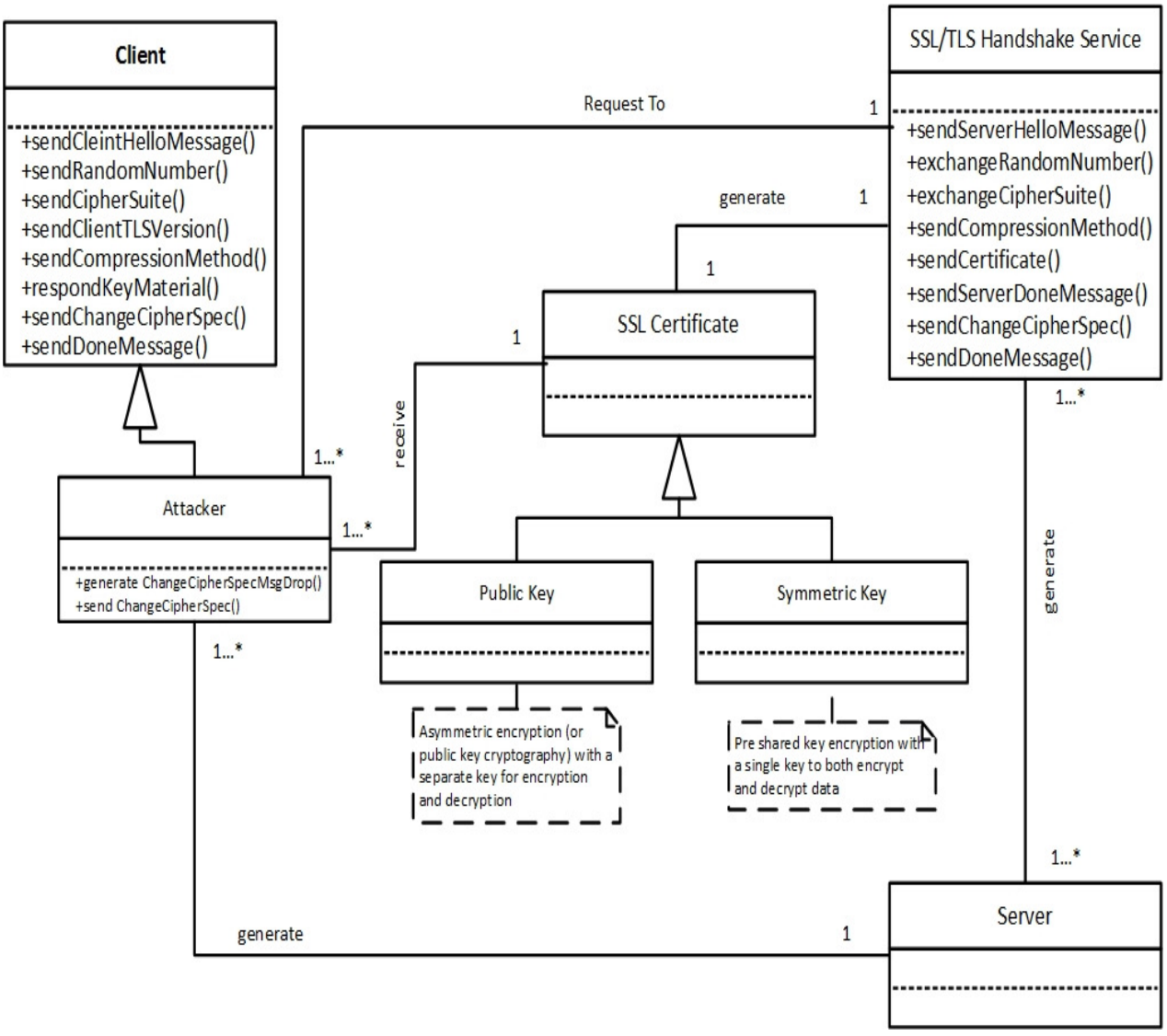

Fig. 4. Class diagram of the misuse pattern: Cipher Suite Rollback

The sequence diagram of Figure 5 shows the typical steps to perform a Cipher Suite Rollback misuse. The Client sends a request to initiate a secure connection with the server. As mentioned earlier, this communication will contain the Cipher Suite message. The attacker intercepts the message with his own CipherSuite-Attacker message and waits for the server's response. When the attacker establishes the secure connection with the server, he has the ability to collect and modify the exchanged data in the coming connections. The attacker can also redirect the client/server connection.

The server does not know whether the client on the other side of the connection is the actual client or an imposter. Based on the Cipher Suite that the server receives from the client, the server replies back with a ServerCipherSuite to establish the connection with the client as shown in Figure 5.

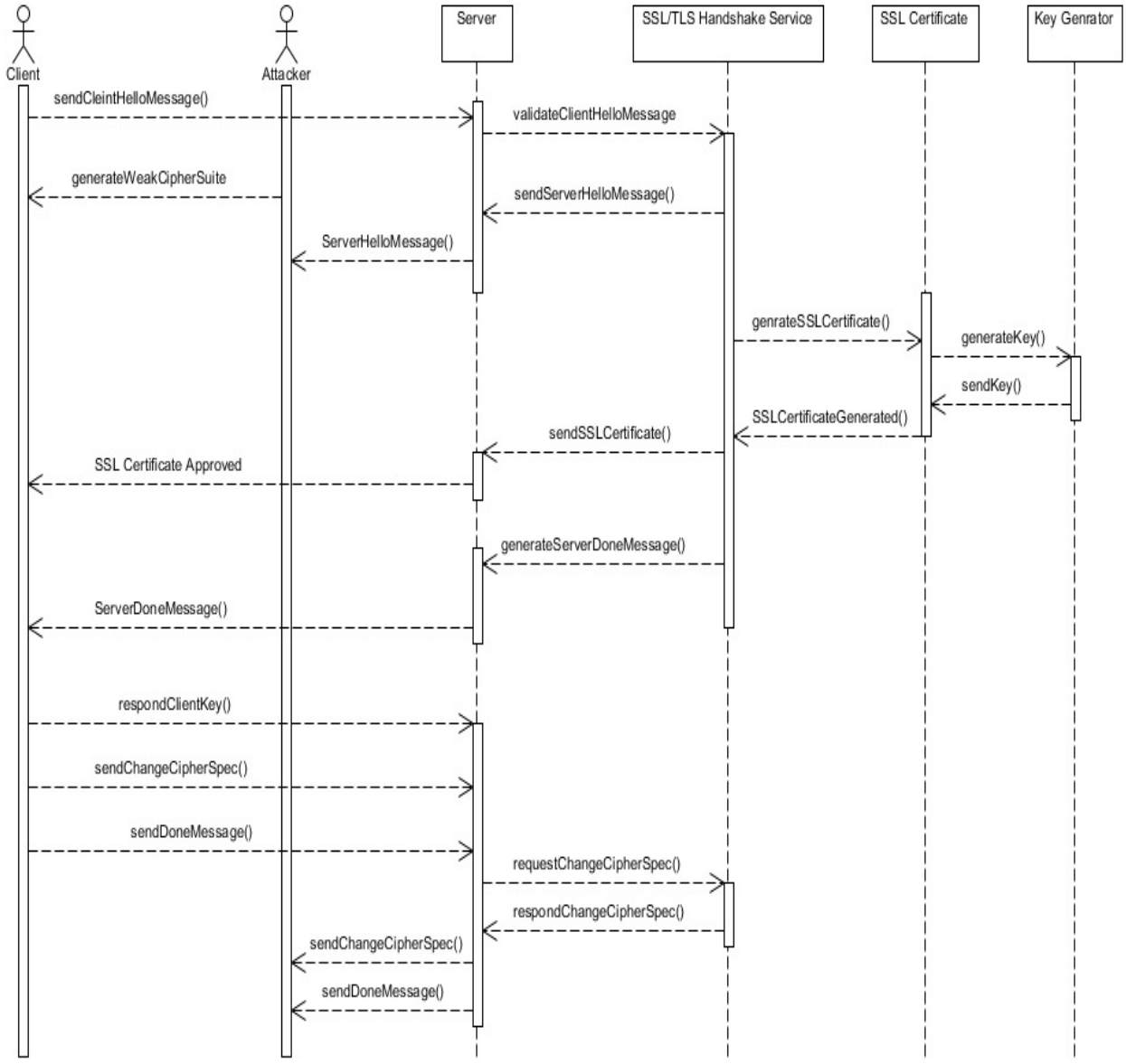

Fig. 5. Cipher Suite Rollback Attack Sequence Diagram

Known uses

(Hole et al. 2006) describe a brute force attack that intercepted the login password in Norway Online Banking System during 2003 and 2004 period. The attack used a large number of botnets that divided large numbers of Social Security Numbers across a network containing zombie PCs. It tried to login in an account by the assigned SSNs. The bank was not able to distinguish between the customer and the zombie PCs trying to log in. Another known issue of this attack is the email interception while connecting to an email server. This attack was performed by Martin Vuagnoux in 2002 by intercepting passwords sent to an IMAP server when checking emails with MS Outlook Express 6.x client using a secure connection (Zhang, H.L. 2003).

Consequences

The success of this attack results in:

- The attacker can impersonate a valid user, access the server, and view the client's information.
- The attacker can modify or read the intercepted messages and perform transactions as a valid user.
- The attacker can inject a malicious code to the intercepted messages to bring on further attacks.
- Once the attack is successful, the attacker may get the user credentials and can use those in attacking other systems

Disadvantages include:

- If the attacker does not know that the server has some logging mechanism that keeps track of all message requests from the client, he might be identified
- The defenses below can stop the attack.

Affected Components

When the attack is finished, several components will be affected:
- Cipher Suite Message: This message will not be trusted because it was intercepted and modified by the attacker. If the server selects a Null Cipher from the client's Cipher Suite List, the log should contain a message indicating the server's use of the Null Cipher. (Oracle, 2014)
- Client Credentials: After the attack is initiated and succeeds, the client will be compromised and his credentials should be revoked.
- Server to Client Certificate: When the attack is successful, the server issues a certificate to the attacker instead of to the client.

Countermeasures

. There are several countermeasures for defending against the Cipher Suite Rollback Attack:

- Message Authentication to prevent the messages from being modified (Fernandez,E.B. 2013).
- Sender Authentication that insures the non-repudiation of the messages.(Fernandez, E.B. 2013; Schumacher et al. 2006)
- Message integrity that prevents the messages from any modification (Fernandez, E.B. 2013).
- Confidentiality of traffic flow that prevents traffic analysis (Fernandez,E.B. 2013).

We can defend against this attack in the following patterns:
- The Authenticator pattern (Brown et al.1999), which describes the procedure of client server authentication in a distributed setting.
- A Security Pattern for the Transport Layer Security (TLS) Protocol (Kumar and Fernandez 2012), which includes authentication in the handshaking.
- Include the Change Cipher Spec in the server's finished authentication message for comparing the interchanged Cipher Spec with the original one sent by the client (Zhang, H.L. 2003).
- Authenticate all the messages of the handshake protocol by including a hash value of all the interchanged messages between the client and the server. This was done with the release of SSL 3.0. The hash value does not include the

interchanged messages (ChangeCipherSpec is an example) which leaves room for future attacks (Meyer and Schwenk 2013) such as SSL downgrading or use of fake SSL certificates.

Related Patterns

- Misuse Patterns in VoIP: These patterns include call hijacking, theft of service, and denial of service (DoS) on VoIP (Pelaez et al. 2007). Some of these misuses could be performed using TLS
- A Security Pattern for the Transport Layer Security (TLS) Protocol: This pattern presents the security measures of the TLS protocol (Kumar and Fernandez 2012)
- An authentication attack to a bank, based on web services, is described in (Muñoz-Arteaga et al. 2011).

3. CONCLUSIONS

The misuse pattern presented in the paper can give a better understanding of the client/server handshake protocol and how attacks can be performed on it. Future work will include writing patterns for other attacks on the SSL/TLS handshake protocol. These attacks will include the Version Rollback attack and Change Cipher Spec Message Drop attack (Zhang, H.L. 2003). This pattern can be generalized to other protocols that use unprotected handshakes (Barkan et al 2003).

ACKNOWLEDGEMENTS
Our shepherd, Prof. Antonio Maña, provided valuable comments and references that significantly improved this paper.

REFERENCES

Barkan, Elad, Eli Biham, and Nathan Keller. 2003. Instant ciphertext-only cryptanalysis of GSM encrypted communication. *Advances in Cryptology-CRYPTO 2003. Springer Berlin Heidelberg*. 600-616.

Brown, F.L., DeVietri, J., Diaz de Villegas, Graziella, and Fernandez, E.B. 1999. The Authenticator Pattern. In *Proceedings of the Conference on Pattern Language of Programs (PloP'99).*

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., and Stal, M. 1996. *Pattern-Oriented Software Architecture: A System of Patterns, Volume 1*, *Wiley*, Chichester, New York, N.Y.

Cluley, J., "With just days to go, just how mant PCs are still running Windows XP", April 2, 2014. http://www.welivesecurity.com/2014/04/02/with-just-days-to-go-just-how-many-pcs-are-still-running-windows-xp/

The Internet Engineering Task Force. 2014. Prohibiting Secure Sockets Layer (SSL) Version 2.0. http://tools.ietf.org/html/rfc6176

Fernandez, E.B. 2013. *Security patterns in practice - Designing secure architectures using software patterns.* Wiley Series on Software Design Patterns. *John Wiley & Sons.*

Hole, K. J., Moen, V., and Tjostheim, T. 2006. Case study: Online banking security, *IEEE Security and Privacy.* 4, 2, 14-20.

Kumar, A., and Fernandez, E.B. 2012. A Security Pattern for the Transport Layer Security (TLS) Protocol. *Procs. 19th. Int. Conference on Pattern Languages of Programs (PLoP2012).*

Meyer, C., and Schwenk, J. 2013. Lessons Learned From Previous SSL/TLS Attacks-A Brief Chronology Of Attacks And Weaknesses." *IACR Cryptology ePrint Archive 2013*, 49.

Microsoft Corporation. 2003. Overview of SSL/TLS Encryption. http://technet.microsoft.com/en-us/library/cc781476(v=ws.10).aspx.

Muñoz-Arteaga, J., Fernandez,E.B., and Caudel,H. 2011. Misuse Pattern: Spoofing Web Services. *In Proceedings of the Asian Pattern Languages of Programs Conference*.

Oracle Corporation. 2014. An Important Note Regarding Null Cipher Use in SSL. http://docs.oracle.com/cd/E14571_01/web.1111/e13705/practices.htm

Pelaez, J. C., Fernandez, E.B., Larrondo-Petrie, M.M., and Wieser, C. 2007. Attack patterns in VoIP", *Procs. of the 14th Pattern Languages of Programs Conference (PLoP2007).*

Rahm, J. 2010. SSL Profiles: Part 1. https://devcentral.f5.com/articles/ssl-profiles-part-1#.U36xji8WdhX.

Schumacher, M., Fernandez, E.B., Hybertson, D., Buschmann, F., and Sommerland, P. 2006. *Security Patterns: Integrating Security and Systems Engineering. John Wiley & Sons*.

Wagner, D., and Schneier, B. 1996. Analysis of the SSL 3.0 protocol. *Proceedings of the 2nd conference on Proceedings of the Second USENIX Workshop on Electronic Commerce - Volume 2 USENIX Association*. 4-4.

Zhang, H.L. 2003. Three attacks in SSL protocol and their solutions. https://www.cs.auckland.ac.nz/courses/compsci725s2c/archive/termpapers/725zhang.pdf