Change Cipher Specification Message Drop Attack: A Misuse Pattern for the SSL/TLS 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 client/server secure connection prevents an attacker from eavesdropping an established connection. 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. It is used in most internet communications for enabling secure web browsing. We present here a misuse pattern for the SSL/TLS Server Authentication Handshake Protocol: Change Cipher Specification Message Drop Attack, where the attacker performs a Man in The Middle attack to intercept and drop the ”ChangeCipherSpec” message which causes the server to never leave its pending state and the communication will not happen.

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 Network Layer uses connection-oriented protocols that provide secure channels between the clients and the servers to protect their interactions. The SSL/TLS is the most common protocol used in the Internet for this purpose. The SSL/TLS Handshake protocol is a sub protocol in the SSL/TLS layer (Figure 1). It is divided into two layers: The first layer is the Handshake layer which contains three sub protocols: Handshake Protocol, Change Cipher Specification Protocol, and the Alert Protocol. The second layer in the SSL/TLS protocol is the Record Layer which sends secure data from the application using the Transport Layer (Microsoft 2003). We concentrate here only on the Handshake Protocol which is used to negotiate the session parameters between the client and the server.

In order to design a secure system, we first need to understand possible threats to our system. Several methods have been developed to identify threats, e.g. (Braz, Fernandez, VanHilst 2008). Once identified, we need to describe how these threats are realized to accomplish a misuse according to the goals of the attacker. A misuse pattern describes how a misuse is performed from the point of view of the attacker (Fernandez2013). It defines the environment where the attack is performed, countermeasures to stop it, and it provides forensic information in order to trace the attack once it happens. Misuse patterns are useful for developers because once they determine that a possible attack can happen in the environment, a corresponding misuse pattern will indicate what security mechanisms are needed as countermeasures. Also, misuse patterns can be very useful for forensic examiners to determine how an attack is performed, and where they can find useful evidence information after the attack is done. An important value of misuse patterns is that they describe the components of the system where the attack is performed using class diagrams and sequence diagrams, relating the attack to specific system units.

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

We describe here a misuse pattern for the SSL/TLS Handshake protocol: Change Cipher Specification Message Drop Attack. During the Handshake phase, the client sends a “ChangeCipherSpec” message to the server informing that the following communication will be secured. This step comes after both parties go through a series of validating messages in order to establish the secure communication (Alkazimi and Fernandez 2014). The server will be in a pending state immediately after receiving the ChangeCipherSpec message from the client (Meyer and Schwenk, 2013). Dropping this message will leave it in this state indefinitely and the communication will never start.

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).These patterns can be used by network administrators, users, testers, and researchers.

.

Fig.1. SSL/TLS Protocol Layers

. .

# TEMPLATE FOR MISUSE PATTERNS

This section describes each part of the template for misuse patterns.

## Name

The name of the pattern should correspond to the generic name given to the specific type of attack in standard attack repositories such as CERT.

## Intent or thumbnail description

A short description of the intended purpose of the pattern (what problem it solves for an attacker).

## Context

The context describes the generic environment including the conditions under which the attack may occur. This may include minimal defenses present in the system as well as typical vulnerabilities of the system. The context can be specified using a deployment diagram of the relevant portions of the system as well as sequence or collaboration diagrams that show the normal use of the system. A class diagram may show the relevant system structure. We can list specific preconditions for an attack to happen.

## Problem

From an attacker’s perspective, the problem is how to find a way to attack the system. An additional problem occurs whenever a system is protected by some defense mechanisms. The forces indicate what factors may be required in order to accomplish the attack and in what way; for example, which vulnerabilities can be exploited. Also, which factors may obstruct or delay accomplishing the attack.

## Solution

This section describes the solution of the hacker’s problem, i.e., how the attack can reach its objectives and the expected results of the attack. UML class diagrams show the involved units of the system under attack. Sequence or collaboration diagrams show the exchange of messages needed to accomplish the attack. State or activity diagrams may add further detail.

Affected system components (Where to look for evidence) (targets)

This is a new section compared to standard security patterns. The solution should not be a comprehensive representation of all components and relationships involved in an attack. Rather, the solution should represent all components that are involved in the attack, are important to prevent the attacks, or are essential to the forensic examination. This can be represented by a class diagram that is a subset or superset of the class diagram of the context.

## Known uses

Specific incidents where this attack occurred are preferred but for new vulnerabilities, where an attack has not yet occurred, specific contexts where the potential attack may occur are enough.

## Consequences

Discusses the benefits and drawbacks of a misuse pattern from the attacker’s viewpoint. Is the effort and cost of the attack commensurate with the results obtained? This is an evaluation that must be made by the attacker when deciding to perform the attack; the designers should evaluate the risk to their assets using some risk analysis approach. The enumeration includes good and bad aspects and should match the forces.

## Countermeasures and Forensics

This section describes the security measures necessary in order to stop, mitigate, or trace this type of attack. This implies an enumeration of which security patterns are effective against this attack. From a forensic viewpoint, it describes what information can be obtained at each stage tracing back the attack and what can be deduced from this data in order to identify this specific attack. Finally, it may indicate what additional information should be collected at the involved units to improve forensic analysis.

## Related Patterns

Discusses other misuse patterns with different objectives but performed in a similar way or with similar objectives but performed in a different way.

1. CHANGE CIPHER SPECIFICATION MESSAGE DROP

Intent

The attacker performs a Man in the Middle attack to prevent the client and the server to start a communication. To achieve this goal, the attacker intercepts and drops the client’s “ChangeCipherSpec” message sent by the client to the server. This message informs both parties to activate the pending state immediately after receiving this notification (Meyer and Schwenk 2013). This is the final stage in client/server Handshake process to establish the secure channel in order to communicate through it. If the attack is successful in dropping this message, the client and the server will never be able to start their communication.

Context

This client/server authentication handshake can be used on some TLS clients and servers that support negotiating the use of Secure Sockets Layer (SSL) version 2.0 (IETF 2011). Also, it can be used on some 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. We assume here that we’re establishing a secure connection between a browser and a server that uses an implementation of TLS 1.0 or TLS 1.1, which were allowed to support the SSL 2.0 negotiation. 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 from the client to the participating server during the connection.

The client sends 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).

The SSL/TLS Handshake begins by exchanging the ClientHelloMessage between the client and server. The message has the CipherSuite and a security list that is exchanged between both parties. The server finds the security list and matches the client’s and starts the connection. They connect through highest security level through the Cipher Suit for the client/server connection that determines the cryptographic primitives. If the client server want to establish a new state of secure communication, it is necessary for both parties to send a ChangeCipherSpec message (Figure 3) The client sends an informative message to the server explaining that the following communication is secured by the previously agreed parameters. The pending state is activated immediately after the ChangeCipherSpec message is received by the server followed by the DoneMessage. The attacker can delete the clinet’s ChangeCipherSpec message which makes the server’s pending state inactive and which makes the DoneMessage from the server unavailable (Meyer and Schwenk 2013, Zhang 2003). Intercepting and dropping an email message is a good example of this attack



Fig. 2. Class Diagram for SSL Server Handshake

Problem

How can we perform the Change Cipher Specification Message Drop attack on the client/server handshake protocol in order to delay the server’s pending state and prevent it from sending the finished message to establish the secure communication with the client?

The attack can take advantage from the following vulnerabilities:

1. The Change Cipher Specification Message is not encrypted and the attacker can access it, which allows him to destroy or reroute it.

Solution

If the attacker manages to intercept the client’s Change Cipher Specification message, he can delete the Change Cipher Specification message and this defeats the authentication protection between the client/server communications which has the impacts described in Consequences.

The class diagram in Figure 4 shows the Change Cipher Specification Message Drop attack. The attack is performed on the client/server handshake protocol

The sequence diagram in Figure 4 shows the steps to perform a Change Cipher Specification Message Drop attack. It begins when the client requests to connect with the server by using secure channel provided by the SSL/TLS handshake protocol. There are several steps to start the connection negotiation until the client sends the ChangeCipherSpec message to the server and waits for the response. The attacker here intercepts this message by generating a ChangeCipherSpecMsgDrop attack and he either drops the client’s message or sends his own modified ChangeCipherSpec message to connect with the server. The attacker now can collect the server’s information and modifies the connection to be unencrypted for further future attacks.

The server does not know that the message is coming from the attacker. The server had several checks before getting into the step of receiving the Change Cipher Specification Message from the client. It does not send a confirmation upon receiving the Change Cipher Specification Message from the client but instead it sends the DoneMessage to the client immediately after receiving the ChangeCipherSpec from any source. The attack is explained in Figure 5.



Fig. 3. Sequence Diagram for Server SSL Handshake



Fig. 4. Class diagram of the misuse Pattern: Change Cipher Specification Message Drop Attack



Fig. 5. Change Cipher Specification Message Attack Sequence Diagram

Known uses

This attack was used for password interception on an email server while the client was connecting with it. It 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). The attack is also used for SSL Spoofing and it is a form of a Man in the Middle attack on SSL in the server/client handshake protocol (OWASP).

Consequences

If the attack is successful the attacker can do the following:

* The attacker can create a Denial of Service attack preventing communications between users and servers.
* If the Change Cipher Specification message is not updated, the message authentication or the encryption in the record layer for the upcoming packets will not be enabled to initiate secure communication channel (Wagner and Schneier 1996).
* The attacker can modify the transmitted data since there is a lack of authentication protection between the client and the server (Wagner and Schneier 1996).
* The resulting of the attack can make the system more vulnerable to several attacks.
* The attacker can replace the server’s certificate and establish a secure connection with the client (OWASP).
* The attacker can perform a cache poisoning attack on the client’s browser where the attacker could download the server’s Java Script file, modify it (poison it) and serve it to the client which gives the attacker the ability to access the client’s sensitive data (iSec Partners)

Some disadvantages include:

* The attack will not take place if there is an acknowledgement from the server for receiving the client’s Change Cipher Specification message.
* If the server has a mechanism of keeping track of the client’s source when requesting the Change Cipher Specification messages, the attacker can be identified.
* Some of the defenses below can stop the attack.

Affected Components

When the attack is finished, several components will be affected:

* Change Cipher Specification Message: This message will not be trusted because it was intercepted and modified by the attacker. This message will contain the server’s information of the trusted communication.
* Client Credentials: After the attack is initiated and succeeds, the client will be compromised and his credentials should be revoked.
* Server to Client Secure Connection: When the attack is successful, the server to client connection is unencrypted which leads to insecure connectivity. The attacker can read, modify or drop the messages between the client and the server. .

Countermeasures

To defend against this attack, we could use the following countermeasures:

* Authentication of the messages; in this case the Change Cipher Specification Message; for preventing the messages from being masqueraded (Fernandez, E.B. 2013).
* Authorization to prevent reading and modification sof the interchanged messages between the client and the server (Fernandez, E.B. 2013).

We can find these countermeasures 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).
* Include the Change Cipher Spec in the server’s Done message (Zhang 2003).
* Authenticate all the messages of the handshake protocol (Meyer and Schwenk 2013).

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).

- A Security Pattern for the Transport Layer Security (TLS) Protocol: This pattern presents the security measures of the TLS protocol (Kumar and Fernandez 2012)

- Cipher Suite Rollback Attack: A Misuse Pattern for Server Authentication Handshake Protocol: This pattern presents a misuse pattern for the SSL/TLS Handshake Protocol 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. The attacker can now gain access to the exchanged data between the client and the server (Alkazimi and Fernandez 2014)

# Conclusion

The misuse pattern presented in the paper will give a better understanding of the client/server handshake protocol and how the attack is 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 (Zhang, H.L. 2003). This pattern may be generalized to other protocols using unprotected handshakes (Barkan et al 2003

ACKNOWLEDGEMENTS

We thank our shepherd, Eduardo Guerra, who under time pressure was able to provide insightful comments that significantly improved this paper.

REFERENCES

Alkazimi, A., and Fernandez, E.B. 2014. Cipher Suite Rollback Attack: A Misuse Pattern for TLS Server Authentication Handshake Protocol. *Procs. 21st Int. Conference on Pattern Languages of Programs (PLoP2014).*

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.

Braz, F., E.B.Fernandez, and M. VanHilst, "Eliciting security requirements through misuse activities" *Procs. of the*  *2nd* *Int. Workshop on Secure Systems  Methodologies using Patterns (SPattern'08)*.  In conjunction with the *4th  International Conference onTrust, Privacy & Security in Digital Busines(TrustBus'08)*,  Turin, Italy, September 1-5, 2008.  328-333.

Brown, F.L., DeVietri, J., de Villegas, Graziella Diaz, 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.

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

iSec Partners. 2012. An Attack on SSL Certificate. <https://isecpartners.com/blog/2012/december/an-attack-on-ssl-client-certificates.aspx.>

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](http://technet.microsoft.com/en-us/library/cc781476%28v%3Dws.10%29.aspx).

Open Web Application Security Project (OWASP). 2014. SSL Spoofing. [https://www.owasp.org/images/7/7a/SSL\_Spoofing.pdf.](https://www.owasp.org/images/7/7a/SSL_Spoofing.pdf.%20)

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](https://devcentral.f5.com/articles/ssl-profiles-part-1%23.U36xji8WdhX)

Rescorla, Eric, *SSL and TLS:* *Designing and building secure systems*, Addison-Wesley, 2001

Wagner, D., and Schneier, B. 1996. Analysis of the SSL 3.0 protocol. In *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>