A Misuse Pattern for Transport Layer Security (TLS): Triple Handshake Authentication Attack

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

Transport Layer Security (TLS) is a cryptographic protocol that provides a secure communication channel between a client and a server. TLS is the successor to the Secure Sockets Layer (SSL) protocol. The secure communication prevents an attacker from eavesdropping an established client-server connection and it is used in most Internet communications for enabling secure web browsing. TLS security remains problematic even after several years of continuous development. TLS has several vulnerabilities and we present here the Triple Handshake Authentication attack where the TLS client connects to a server that has already been breached by an attacker. This attacker can then impersonate the new client by using her credentials to gain access to other servers accepting those credentials. This attack may allow the attacker to read sensitive cryptographic keys, login credentials and private data from other servers.

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 attacks, security, misuse patterns, security patterns.

1. INTRODUCTION

Transport Layer Security (TLS) is the successor of the Secure Sockets Layer (SSL) protocol, a cryptographic protocol that provides a secure communication channel between a client and a server. TLS establishes a point to point communication between client and server that uses encryption to provide message confidentiality. TLS has some flaws and the interchanged encrypted messages may be the object of attacks that include message scanning, message modification and message interruption, leading to the triple handshake attack found by (Bhargavan et al., 2014). In this attack the TLS client connects to a server that has already been breached by an attacker. The attacker can then impersonate the new client by using her credentials to gain access to other servers accepting those credentials. This attack may allow the attacker to read sensitive cryptographic keys, login credentials and private data from other servers. This attack is catalogued in Bugzilla (Bugzilla).

The TLS protocol consists of two sub-protocols which are called the handshake protocol and the record protocol. The handshake protocol is intended to authenticate the two communicating parties, namely the client and the server, and establish shared encryption keys between them. The record protocol uses those keys to exchange data securely (Green, 2014). The connection protection in the record protocol requires specification of a suite of algorithms, a master secret, and the client and server random values (Dierks and Rescorla, 2008).

In order to design a secure system, we first need to understand its possible threats. For that reason, we introduced the concept of misuse pattern (Fernandez, Pelaez, and Larrondo-Petrie, 2007) which describes how an information misuse is performed from the attacker’s point of view. Misuse patterns define where the attack is performed, how to stop the attack by providing countermeasures, and provides forensic information in order to trace the attack once it happens.

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.ed

Misuse patterns also describe the components of the system used in the attack with UML class and sequence diagrams. 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.

2. TEMPLATE FOR MISUSE PATTERNS

Since misuse patterns are not well known we present first the template that we use to describe them. This is based on the template used by (Buschmann et al., 1996).

2.1 Name

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

2.2 Intent or thumbnail description

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

2.3 Context

It 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 standard vulnerabilities of the system.

2.4 Problem for the attacker

From an attacker's perspective, the problem is how to find a way to attack the system. 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.

2.5 Solution

This section describes the solution of the attacker's problem, i.e., how the attack can reach its objectives and the expected results of the attack. UML class diagrams show the system units involved in the attack. Sequence or collaboration diagrams show the exchange of messages needed to accomplish the attack.

2.6 Affected system components (Where to look for evidence)

The pattern should indicate in the class diagram the role of all components that are involved in the attack. From a forensic viewpoint, this section describes what information can be obtained at each stage tracing back the attack and what can be deduced from this data.

2.7 Known uses

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

2.8 Consequences for the attacker

Discusses the benefits and drawbacks of a threat pattern from the attacker's viewpoint. The enumeration includes good and bad aspects and should match the forces.

2.9 Countermeasures

Describes the security measures necessary in order to stop, mitigate, or trace this type of attack. This implies an enumeration of which security patterns or other practical measures are effective against this attack.

2.10 Related Patterns

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

3. TRIPLE HANDSHAKE AUTHENTICATION ATTACK

3.1 Intent

An attacker compromises a server and waits for a new client to connect to it. The attacker can then impersonate the new client by using her credentials to gain access to other servers accepting those credentials. This attack may allow the attacker to read sensitive cryptographic keys, login credentials and private data from other servers

3.2 Context

The conditions for the attack are the following (Bhargavan et al., 2014):

- The RSA key exchange algorithm is used for server authentication and key exchange.
- The destination server chooses the Diffie-Hellman (DHE) parameters during the handshake.
- The completed handshake between the Client, Compromised Application Server (CAS) and the Target Application Server (TAS) have the same key materials and session parameters.

3.3 Problem

The client's main reason for this communication is to securely gain access to her personal bank account, email, health records or any other web services through the target server. The server's involvement in this procedure is to authenticate the client and provide access to the resources based on the client's credentials. In general, the client does not know if the server has been compromised. From the attacker's perspective, the problem is how to get the victim's credentials and initiate other connections to several servers which may accept the victim's credentials.

The attack can be performed by taking advantage of the following TLS vulnerabilities during the client/server handshake:

- The server session resumption on new connections uses an abbreviated handshake which only verifies that the client and server share the same master secret (MS), ciphersuite, and server-issued session ticket (SID) (Bhargavan et al., 2014).

The attack is facilitated by:

- Enabling the use of unsuitable ciphersuites during the TLS handshake (AlFardan and Bernstein, 2013).
- Ignoring a state change during the TLS handshake (Raymond and Stiglic, 2000).
- Using compressions during the TLS handshake (Rizzo and Duong, 2012).
- Creating two connections by the attacker with the client's encryption keys.
- Using the client's active handshake to establish a new connection with other trusted servers accepting the client's credentials.

3.4 Solution

The attack uses an active TLS client connection to a malicious server. The server presents a stolen client's credential to impersonate her at another server accepting this credential. The malicious server performs a man-in-the-middle attack on three successive handshakes between the client and server; he then succeeds in impersonating the client on the third handshake (Bhargavan et al., 2014).

3.4.1 Structure (Affected system components)

The class diagram in Figure 1 demonstrates the Triple Handshake use of vulnerabilities in a client/server connection. The Attacker first compromises the Application Server (there are several ways to accomplish this). Now the server become a Compromised Application Server (AS) and the Attacker waits for a Client to initiate a connection with it to take the Client's credential and uses the Compromised Application Server (AS) to act as a Client to initiate another connection with a Target Application Server (TS) that accepts the Client credentials.

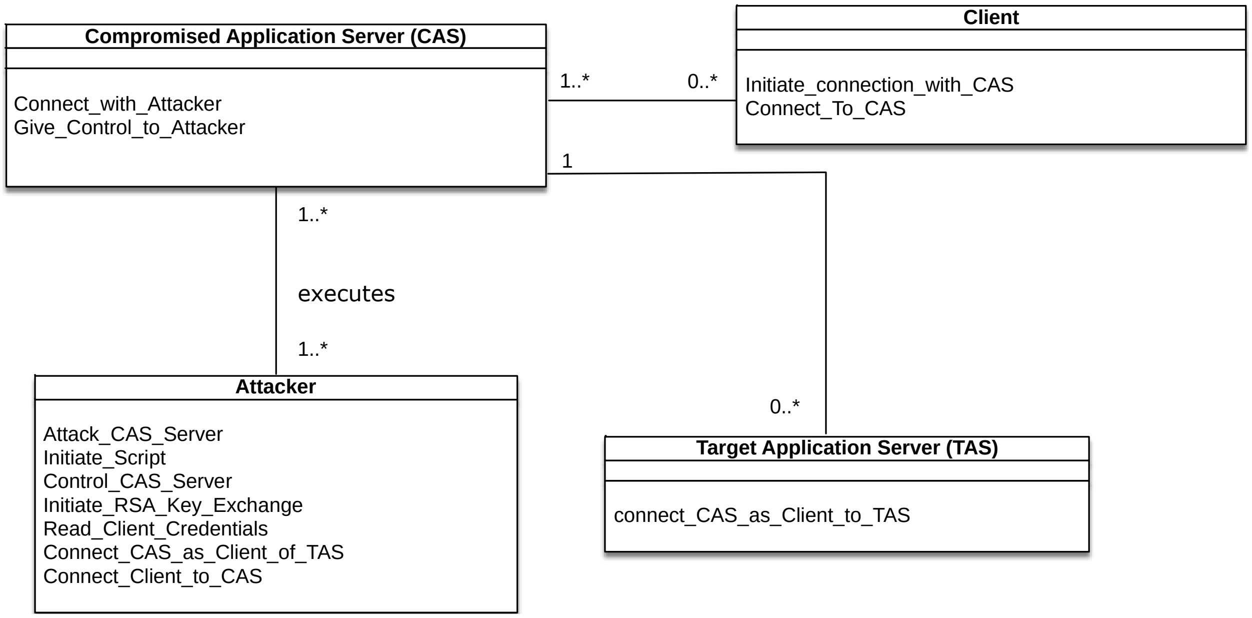

Figure 1: Class Diagram for Use Case "Triple Handshake Attack"

3.4.2 Dynamics

The following use case describes the sequence of the attack to make the compromised application server connect and get information of the target application server.

UC: Triple Handshake Attack (Fig. 2)

Summary: The Attacker takes control of an Application Server and makes it act as a Client to connect to another server in order to get sensitive information from it.

Actor: Attacker

Precondition: The Attacker must gain control of the Application Server.

Description:

- When the Attacker gains access to the Compromised Application Server (CAS), he injects a script containing malicious code into the server or a similar attack which gives her full control of this server.
- A Client visits the Compromised Application Server (CAS) to initiate a connection in order to access an application.
- The Attacker can force the Client and Target Application Server (TAS) through the Compromised Application Server (CAS) to use the RSA key exchange (and can then decrypt the connection) even if they would normally prefer a different cipher in the handshake (Aviram et al., 2016).
- The Attacker waits for the response from the Compromised Application Server (CAS), Target Application Server (TAS), and the Client to complete their handshake.
- The Attacker uses this handshake to make the Compromised Application Server (CAS) connect with the Target Application Server (TAS) as a Client with a third handshake.
- The Target Application Server does not know that the connection is with another server instead of a connection to a Client due to the Attacker forcing to use the RSA key exchange in the client/server initial handshake.
- The Attacker performs some misuse on the Target Application Server once he gets the credential of the client.

Post-condition: Client's credentials and Target Application Server (TS) are compromised (see possible benefits for the Attacker in Section 3.7).

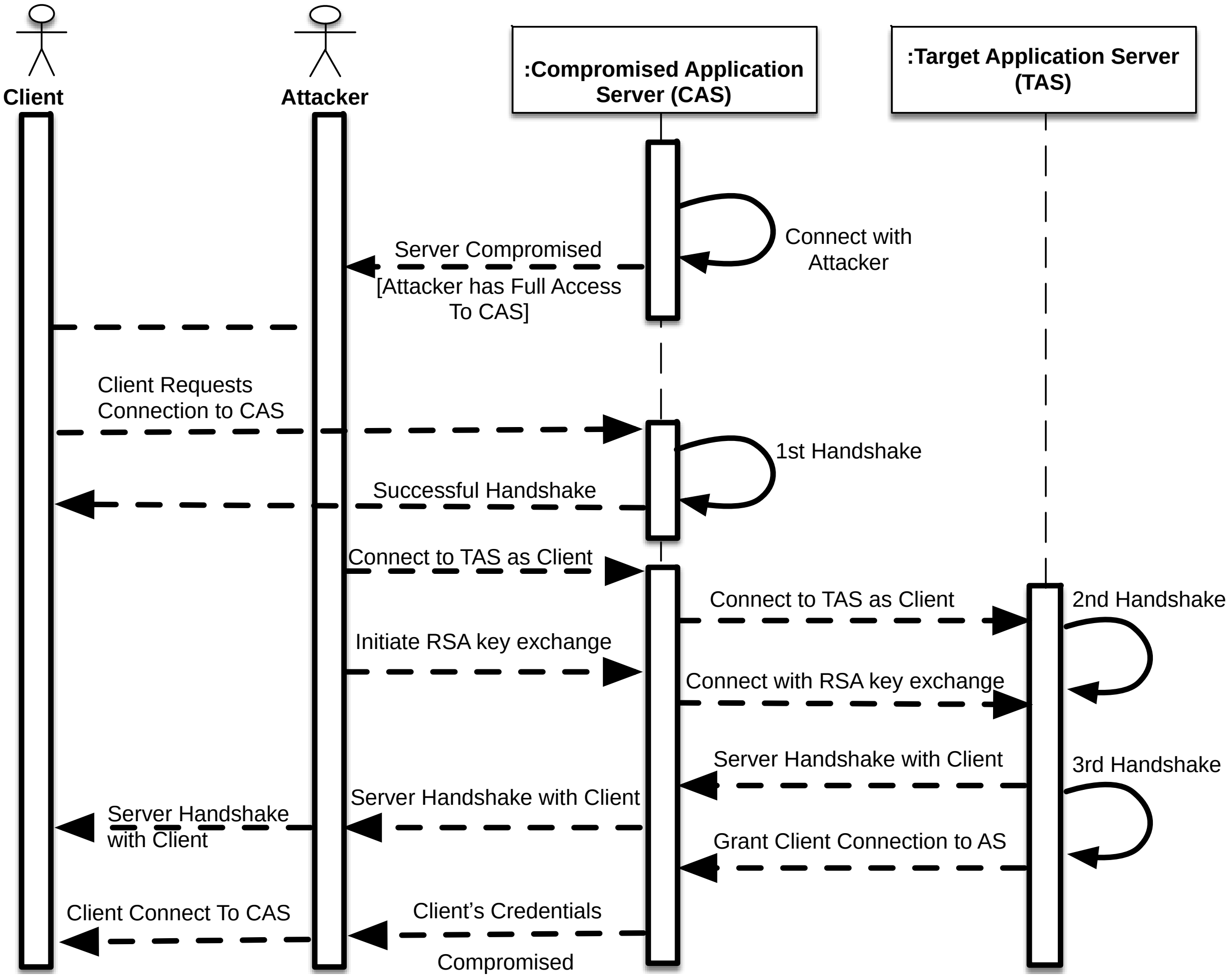
Client
Attacker
:Compromised Application Server (CAS)
:Target Application Server (TAS)
Connect with Attacker
Server Compromised [Attacker has Full Access To CAS]
Client Requests Connection to CAS
1st Handshake
Successful Handshake
Connect to TAS as Client
Connect to TAS as Client
2nd Handshake
Initiate RSA key exchange
Connect with RSA key exchange
Server Handshake with Client
3rd Handshake
Server Handshake with Client
Server Handshake with Client
Grant Client Connection to AS
Client Connect To CAS
Client's Credentials
Compromised

Figure 2. Sequence Diagram for the use case Triple Handshake Attack

3.5 Affected system components

- We can audit a compromised Target Application Server (TAS) connection and check which handshake method was used in the client/server communication for evidence of Triple Handshake Attack in application or connection logs.

3.6 Known uses

The attack was discovered by a group at INRIA, Paris, in 2014 (Bhargavan, K. et al.), and there are no known incidents where this attack was used.

3.7 Consequences

Some of the benefits of this pattern for the attacker are the following:

- The Attacker can access the Target Application Server by impersonating a client to obtain confidential information or perform one of the following misuses: perform session hijacking, steal sensitive information such as usernames and passwords, poison cookies, impersonate a web page to gather information from other clients later, or create a pivoting point for other attacks.

Possible sources of failure for the attacker include:

- The countermeasures below (section 3.8) would cause the attacker to fail.

3.8 Countermeasures

The Triple Handshake Attack can be stopped by the following countermeasures:

- Validate all user input (whitelist) and reject any connection requests that are not explicitly allowed on the server and the client (Sulatycki and Fernandez, 2013).
- Use of a Content Security Policy (CSP) is a possibility to avoid the Triple Handshake Attack. This policy instructs the client browser on the location and type of resources that are allowed to be loaded(Cobb, 2009).
- Validate all client/server certificates during the TLS handshake and abort the the handshake if they are not valid (Bhargavan et al., 2014).

3.9 Related patterns

- Cipher Suite Rollback: A Misuse Pattern for SSL/TLS Authentication Handshake Protocol (Alkazimi and Fernandez, 2014).
- Change Cipher Specification Message Drop Attack: A Misuse Pattern for the SSL/TLS Server Authentication Handshake Protocol (Alkazimi and Fernandez, 2014).

4. CONCLUSIONS

The misuse pattern presented in the paper can give a better understanding of the Triple Handshake Attack and how it can be performed. In order to design secure systems, designers need to understand the possible threats, vulnerabilities and the whole misuse performed to the systems. We are continuing developing threat/misuse patterns for application security to have a relatively complete catalog that can be used by application developers. Finally, we intend to incorporate these patterns into a secure systems design methodology.

REFERENCES

AlFardan, N., Bernstein, D. J., Paterson, K. G., Poettering, B., and Schuldt, J. C. (2013). On the security of RC4 in TLS. 22nd USENIX Security Symposium, 305-320.

Alkazimi, A., and Fernandez, E.B., (2014). Change Cipher Specification Message Drop Attack: A Misuse Pattern for the SSL/TLS Server Authentication Handshake Protocol. Proceedings of the 10th Latin American Conference on Pattern Languages of Programs (SugarLoaf PLoP 2014).

Alkazimi, A. and Fernandez, E.B., "Cipher Suite Rollback Attack: A Misuse Pattern for TLS Server Authentication Handshake Protocol", Procs. of the 21st Conf. on Pattern Languages of Programs (PLoP 2014)

Aviram, N., Schinzel, S., Somorovsky, J., Heninger, N., Dankel, M., Steube, J., Valenta, L., Adrian, D., Halderman, J.A., Dukhovni, V. and Käsper, E., DROWN: Breaking TLS using SSLv2. Retrieved May 23, 2016, from https://drownattack.com/

Bhargavan, K., Delignat-Lavaud, A., Fournet, C., Pironti, A., and Strub, P. Y. (2014). Triple handshakes and cookie cutters: Breaking and fixing authentication over TLS. Proceedings of the IEEE Symposium on Security and Privacy (98–113). http://doi.org/10.1109/SP.2014.14

Buschmann, F., Meunier, R., Rohnert, H., Sommerlad, P., and Stal, M. (1996). Pattern-Oriented Software Architecture - A System of Patterns. Network.

Cobb, M. (2009). Cross-site scripting explained: How to prevent XSS attacks. Retrieved June 12, 2016, from http://www.computerweekly.com/tip/Cross-site-scripting-explained-How-to-prevent-XSS-attacks

Dierks, T., and Rescorla, E. (2008). RFC 5246 - The transport layer security (TLS) protocol - Version 1.2. In Network Working Group, IETF, 1–105.

Fernandez, E.B., Pelaez, J.C., and Larrondo-Petrie, M.M., Attack patterns: A new forensic and design tool, Procs. of the Third Annual IFIP WG 11.9 Int. Conf. on Digital Forensics, Orlando, FL, Jan. 29-31, 2007. www.cis.utulsa.edu/ifip119. Chapter 24 in Advances in Digital Forensics III, P. Craiger and S. Shenoi (Eds.), Springer/IFIP, 2007, 345-357.

Goodin, D. (2014). Critical crypto bug in OpenSSL opens two-thirds of the Web to eavesdropping | Ars Technica. Retrieved February 22, 2016, from http://arstechnica.com/security/2014/04/critical-crypto-bug-in-openssl-opens-two-thirds-of-the-web-to-eavesdropping/

Green, M. (2014). Attack of the Week: Triple Handshakes (3Shake). Retrieved May 11, 2016, from http://blog.cryptographyengineering.com/2014/04/attack-of-week-triple-handshakes-3shake.html

Raymond, J. F., and Stiglic, A. (2000). Security issues in the Diffie-Hellman key agreement protocol. IEEE Transactions on Information Theory, 22, 1-17.

Red Hat Bugzilla, Triple Handshake Attack, Bug 1151046
https://bugzilla.redhat.com/show_bug.cgi?id=CVE-2014-6457

Schumacher M., Fernandez, E.B., Hybertson, D., Buschmann, F., and Sommerlad P., Security Patterns: Integrating security and systems engineering, Wiley Series on Software Design Patterns, 2006.

Sulatycki Rohini and. Fernandez Eduardo B, A threat pattern for the “Cross-Site Scripting (XSS)” attack, 22nd Conference on Pattern Languages of Programs 2015, Pittsburgh, PA, October 24-26, 2015