Lucky Thirteen

Justus Benedict Siegert

8445024, s222312@student.dhbw-mannheim.de

Inhaltsverzeichnis

1 Einleitung

Das Verschlüsseln von Daten ist seit der Nutzung des Internets ein wichtiges Anliegen der Nutzenden. Das am häufigsten genutzte Protokoll ist das Transport Layer Security Protokoll (TLS). Lucky 13 ist eine kryptografische Schwachstelle des TLS-Protokolls in der Version 1.1, sowie dem Vorgänger TLS 1.0, SSL, die Maßnahmen gegen Padding Oracle Attacks enthalten und dem Datagram Transport Layer Security (DTLS) [2, S.2], bei dem die Daten kompromittiert werden. Durch die Schwachstelle können Angreifer Daten ausspionieren, die verschlüsselt sind und vermeintlich sicher seien. Die Sicherheitslücke wurde 2013 von den Wissenschaftlern Nadhem J. AlFardan and Kenneth G. Paterson, der Security Group der Royal Holloway University of London entdeckt.
Der Lucky 13 Angriff ist eine Art von Man in the Middle attack verbunden mit einer Padding Oracle Attack, weil der Angreifer zwischen den Client und Server muss, um die verschlüsselten Nachrichten zu lesen und zu senden. Gleichzeitig beruht der Angriff auf einem Padding Oracle Attack, wo verschiedene Paddings an ein Server geschickt werden, um das richtig formatierte zu erfahren. Dabei ist das Problem, dass das Padding nach der Berechnung des message authentication code (MAC) hinzugefügt wird und somit unauthentifizierte Daten im verschlüsselten Klartext bildet. Im Folgendem werden verschiedene Aspekte zum verständnis der Sicherheitslücke ausgeführt.

1.1 Man-in-the-Middle-Attack

Eine Man-in-the-Middle-Attack (MITM) ist ein Angriff, bei dem eine dritte Partei eine vermeintlich direkte Verbindung von zwei anderen zwischen schaltet. Bei einem solchen Angriff, wird unerlaubt auf den Datenverkehr zweier Kommunikationspartner zugegriffen, in dem der Angreifer versucht zwischen den Datenverkehr der beiden Kommunikationspartner zu gelangen und diesen abzufangen oder zu manipuliert. Die Gefahr dabei ist, dass es nicht im Rahmen einer normalen Datenübertragung zu erkennen ist, wenn jemand sich dazwischen geschaltet hat.

1.2 Padding Oracle aAttack

Die Padding Oracal Attck wurde 2002 von Vaudenay entdeckt und dient als Basis für diesen Angriff. Bei einem Padding Oracle Attack wird ausgenutzt, dass einige Chiffren ein Padding benötigen, um eine Blockverschlüsslung durchzuführen, damit die Nachricht auf Blocklänge ist. Der Angreifer schickt dann Nachrichten an den Server mit verschiedenen Paddings und schaut, ob der Server das Padding akzeptiert. Bei einem korrektem Padding kann dann die Nachricht genutzt werden, um die Verschlüsslung der Datenpakete zu brechen. Durch die präzise Antwort des Servers, dass es zu einem Entschlüsselungsfehler kam, resultiert der name Orakel[1]. Um den gewöhnlichen Angriff zu verhindern, wurden die Server so konfiguriert, dass allgemein nur eine Fehlerrückmeldung gesendet wird, die nicht auf die Art des Fehlers schließen lässt.
Bei dem Beispiel Paper von Nadhem J. AlFardan and Kenneth G. Paterson haben sie lediglich die Padding Oracle Attack in verbindung der Blockchiffre CBC verwen-

det. Dies ist ein typisches Beispiel für einen Verschlüsselungsalgorithmus, bei dem der Angriff möglich ist.

2 (Datagram) Transport Layer Security

Der vorgänger von TLS, Secure Socket Layer (SSL) 1.0 erschien im Jahr 1994, neun Monate nach der ersten Mosaic version. In den folgenden zwei Jahre wurden zwei weiter Versionen entwickelt und dann SSL in TLS 1.0 umbenannt.
TLS besteht aus zwei Hauptkomponenten: dem TLS Handshake, der für den sicheren Schlüsselaustausch zwischen dem Client und dem Server zuständig ist, und dem TLS Record, dieser verwendet die beim Handshake ausgehandelten Schlüssel, um eine sichere Datenübertragung zu ermöglichen. Die Daten werden verschlüsselt und mit einem MAC gegen manipulation geschützt. Die grundlegende Funktionsweise besteht darin, dass der Client eine Verbindung mit dem Server aufbaut, welcher sich mit einem Zertifikat authentifiziert. Der Client überprüft die Vertrauenswürdigkeit und vergleicht die Daten des Zertifikats, mit den des Servers.
DTLS basiert auf TLS, kann aber im Gegensatz zu TLS auch Transportprotokolle wie UDP übertragen, hat aber grundlegend dieselben Eigenschaften und Sicherheitsmerkmale

2.1 Cipher Block Chaining Mode Mode

Cipher Block Chaining Mode (CBC) ist eine Betriebsart in der Blockchiffre, dabei wird vor dem Verschlüsseln des Klartextblocks dieser mit dem vorausgehenden verschlüsselten Block per XOR verknüpft1. Die Vorteiler dieses Modus sind, dass gleich Klartextblöcke unterschiedliche verschlüsselte Blöcke ergibt, allerdings können so nicht gleichzeit mehrere Blocks verschlüsselt werden, da sie von einander abhängen. Im weiteren Verlauf wird unter dem CBC-Mode beschrieben.

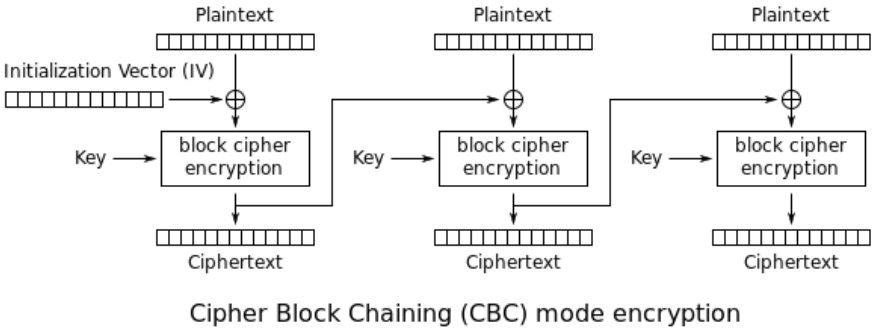

Abbildung 1: Verschlüsselung im CBC-Mode

3 Voraussetzungen für einen Angriff

Damit der Angriff gelingt, muss die TLS version 1.1 oder eine ältere Version mit Maßnahmen gegen Padding Oracle Attacks. Der Angriff ist darauf ausgelegt die Maßnahmen gegen einen normale Padding Oracle Attack zu umgehen, indem die Padding-Überprüfung beim Server nicht über den Inhalt der Antwort analysiert wird, sondern über die Verarbeitungszeit. Bei anderen älteren Versionen ist der Angriff deshalb nicht Empfehlenswert, weil es andere einfachere Sicherheitslücken zum Ausnutzen gibt.
Lucky 13 nutzt den Zeitkanal aus, welche nur getestet wurde im Lokalen Netzwerk des Servers. Um eine genaue Zeitmessung zu ermöglichen, muss man das Rauschen des Netzwerks analysieren, damit man die Antwortdauer der Pakete unterscheiden kann. Außerhalb dieses Netzwerks sind andere Variablen, die die Zeit beeinflussen und deshalb bisher nicht getestet wurden von in dem Paper der Sicherheitswissenschaftler[2, S. 11]. Des Weiteren muss man zwischen den Client und Server gelangen, um die Pakete abzufangen und an die Sequenznummern zu kommen, da diese inkrementiert wird.

4 Angriffsverlauf

Der Angriff ist gleich aufgebaut wie ein normaler Padding Oracle Attack, nur dass statt die Antwort des Servers, die Antwortzeit ausschlaggebend ist für die analyse. Dieser Unterschied entsteht dadurch, dass bei korrekten Paddings eine MAC prüfung durchgeführt werden muss und bei falschen nicht. Ein Problem des Angriffs ist, dass Fehler bei TLS immer fatal sind und zu einem session abbruch führen, beidem alle Schlüssel sowie kryptografisches Material entsorgt werden. Bei DTLS kann ein solcher Fehler auch als nicht sehr schwerwiegend eingestuft werden, weshalb es zu keinem Abbruch der Verbindung kommt [2, S.5]. Deshalb wird zur Anfrage für die Zeitmessung immer ein DTLS Heartbeat geschickt, damit dieser Analysiert werden kann.

4.1 Aufbau der TLS Nachricht

Zur Bildung der Nachricht wird von den genau 13 Byte (8 Byte Sequenznummer (SQN); 2 Byte Version, 1 Byte Typ, 2 Byte Länge (HDR)) und der Inhalt der Nachricht der Message Authentication Code gebildet. Dieser kann je nach Hashalgorithmus verschiedene Längen (16 bei HMAC-MD5, 20 bei HMAC-SHA-1 oder 32 Byte bei HMAC-SHA-256) annehmen und wir nennen ihn T. Der Inhalt (R), der MAC (T) und ein Padding (pad) werden dann mit einer Blockchiffre verschlüsselt. Das Padding wird so gewählt, dass die Gesamtlänge von R||T||pad gleich der Blocklänge der Chiffre ist. Bei 3DES beträgt die Blocklänge 8 und bei AES 16 Byte. Das Padding besteht aus n Byte mit dem Wert von (n-1), dadurch entstehen Paddings wie "0x00", "0x01||0x01" oder "0x02||0x02||0x02". Die fertige TLS-Nachricht besteht dann 1. aus der Inhalt der nachricht 2. aus der Berechneten MAC und 3. dem Padding. Alle diese Teile werden dann verschlüsselt und bilden die gesamte Nachricht [2, S.4].

Nach dem der Aufbau der TLS Nachrichten beschrieben wurde, wird im Folgendem auf die Klartextwiederherstellung eingegangen. Unsere Grundannahmen sind dabei, dass es sich um den CBC-Mode handelt, dass wir eine Blocklänge von 16 haben, also AES verwenden und HMAC-SHA-1 verwendet wird,also der MAC 20 Byte groß ist. Außerdem wird nur TLS betrachtet, allerdings treffen die Funktionen auch auf DTLS zu [2, s.7].

4.2 Wiederherstellung

Die Entschlüsselung von Nachrichten beim Server findet rückwärts wie dir Verschlüsselung 1 statt. Dabei wird der verschlüsselte Block entschlüsselt und dann mit dem vorherigen verschlüsselten Block per XOR verknüpft2.
Wir haben jetzt zum Beispiel eine Gesamtlänge von 64 Byte die verschlüsselt wurden, dann testen wir, ob das Padding "0x00" ist. Da wir eine Blocklänge von 16 haben, probieren wir beim 48. Byte alle Möglichkeiten aus. Bei allen anderen wird kein MAC-Test durchgeführt, weil das Padding länger ist und ein Padding-Error entsteht. Dann entfernen wir 20 Byte vom Ende, weil dies der MAC-Tag ist. Damit ist der Inhalt 43 Byte groß und dazu die 13 Byte (8 Byte Sequenznummer (SQN); 2 Byte Version, 1 Byte Typ, 2 Byte Länge (HDR)). Die MAC verifizierung findet dann an einer maximal 56 Byte großen Nachricht statt [2, S.7].
Wenn das Padding eine Mindestlänge von 2 hat, hat der Inhalt eine maximale Größe von 42, weil 64(Gesamtläng)-20(MAC-Tag)-2(Padding)=42. Damit findet dann der MAC-Test an einer größe von maximal 52 Byte (13(SQN||HDR)+ 42(Inhalt))[2, S.7]. In allen anderen Fällen, handelt es sich um kein vorschriftsgemäßes Padding und es werden nur die 20 Byte MAC-Tag abgezogen und an einer Länge von 57 Byte überprüft[2, s.7]. Da im 2. Fall nur 4 Auswertungen durchgeführt werden müssen, erkennt man diesen Fall über den Zeitkanal [2, s.7]. Um den 2. Fall zu finden wird wie beim 2 der vorletzte Block die letzten beiden Bytes durch probiert, bis ein solcher Zeitunterschied festgestellt wird. Für die weiteren Bytes kann dann das Verfahren von Vaudenay verwendet werden, von dem normalen Padding Oracle Attack[**?**,].

5 Sicherheitsbewertung

Die schwachstelle Lucky 13 wird allgemein mit einer geringen Sicherheitsbewertungsstufe angegeben. Begründungen hierfür sind in erster Linie auch direkt die Erkennung des richtigen Paddings,welches über die Dauer der Antwort des Servers ermittelt wird. Das ist allerdings äußerst unwahrscheinlich, wenn man sich nicht im selben Netzwerk, wie der Server befindet. Nadhem J. AlFardan and Kenneth G. Paterson haben die Analysen für ihr Paper auch nur im selben Netzwerk durchgeführt. Suse schreibt dazu "Please note that an exploit for this requires physical proximity of an attacker to the server, which is usually unlikely", was Übersetzt bedeutet: Bitte beachten Sie, dass ein Exploit für dieses Problem die physische Nähe eines Angreifers zum Server erfordert, was in der Regel unwahrscheinlich ist.
Die Bewertung von Schwachstellen findet anhand des Common Vulnerability Scoring

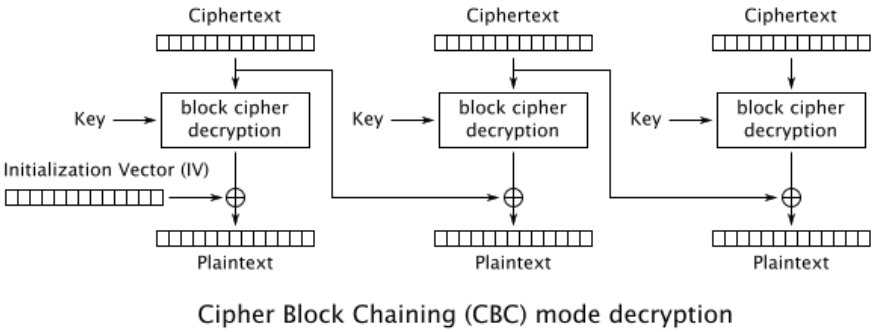

Abbildung 2: Entschlüsselung im CBC-Mode

System (CVSS v2) statt, dies ist der Industriestandard zur Bewertung für mögliche und tatsächliche Sicherheitslücken. Dort werden verschiedene Kategorien, die in 1 zusehen sind, bewertet und dann nach einer Formel ein Wert zwischen 0 und 10 ermittelt. Je geringer der Wert ist, desto kleiner ist das Risiko. Auf der Skala ist das Risiko bis 3,9 gering, zwischen 4,0 und 6,9 ist es medium und ab 7,0 ist das Risiko hoch. Deshalb wird von Suse der CVSS-Vector wie Folgt gesetzt:

Exploitability Metrics		**Impact Metrics**	
Access Vector	Network	**Confidentiality Impact**	Partial
Access Complexity	High	**Integrity Impact**	None
Authentication	None	**Availability Impact**	None

Tabelle 1: Bewertung von Lucky 13 durch Suse in der CVSS-Skalar (Version 2)

Das ergibt den Vektor AV:N/AC:H/Au:N/C:P/I:N/A:N mit einem CVSS-Score von 2.6. IBM bewertet Lucky 13 gleich, außer in Hinsicht auf die Access Complexity, die als medium gewertet wird, wodurch die Gesamtbewertung auf 4.3 ansteigt. Die beiden Bewertungen fallen gering aus, die von Suse ist im niedrigen Risiko und die von IBM im mittleren Bereich, wodurch zusagen ist, dass es unwahrscheinlich ist Opfer durch den Lucky 13 Angriff zu werden.

6 Gegenmaßnahmen

Es gibt mehrere Maßnahmen, die ergriffen werden können, um einen Angriffserfolg zu minimieren. Die Maßnahmen müssen verhindern, dass man eine Zeitanalyse durchführen kann oder sie muss das benötigte Padding in der Blockverschlüsslung umgehen.

6.1 Zufällige Zeitverzögerungen

Bei einem Angriff, der auf die Dauer einer Reaktion abzielt, liegt eine zufällige verzögerung nahe. Allerdings ist dies ineffektiv wie N. AlFardan und K. G. Paterson schreiben [2, S.13].

6.2 Rons's Code 4 Chiffre

Die Ron's Code 4 Chiffre (RC4) wird zum bei der Veröffentlichung von N. AlFardan und K. G. Paterson als eine Möglichkeit zum Schutz erwähnt, weil kein Padding benötigt wird. Allerdings wurde auch gesagt, dass dies keine Möglichkeit für DTLS ist [2, S.13]. 2015 wurde von der Internet Engineering Task Force (IETF) die Nutzung und Kommunikation über RC4 in verbindung mit TLS verboten, weil es Sicherheitsmängel aufweist.

6.3 Implementierung der MEE-TLS-CBC-Entschlüsselung

Dies ist die von den Publishern des Angriffs empfohlene Methode. Da bei geht es darum, die Verarbeitungszeit nicht von dem Inhalt oder dem Padding abhängig zu machen, sondern nur von der Länge der chiffrierten Nachricht [2, s.13]. Hier muss ich noch weiter schreiben.

Literatur

[1] Juliano Rizzo; Thai Duong. Practical padding oracle attacks. *USENIX WOOT*, 2010.

[2] Nadhem J. AlFardan; Kenneth G. Paterson. Lucky thirteen: Breaking the tls and dtls record protocols. *IEEE Symposium on Security and Privacy*, 2013.