POODLE ATTACK
Exploiting the SSL 3.0 Fallback

Ioana-Andreea Ciupitu and Andrei-Catalin Ilie
Politehnica University of Bucharest, Computer Science Department, Romania
Email: ioanaandreea.1996@gmail.com, and96x@gmail.com

Abstract

In this paper, we will present an exploit of SSL 3.0, which by now is an already obsolete and insecure protocol, and has been replaced afterwards by the TLS protocol, although in some practical cases, if a client and a server both support TLS, according to possible interoperability bugs, a downgrade to SSL is needed. An attacker can exploit this change due to the fact SSL is insecure.

Encryption in SSLv3 uses either RC4 or block cipher in CBC mode. RC4 is known to have exploits such as Beast attack, or Lucky 13, but for those attacks there are some reasonable workarounds. With CBC however, the attack I will present on the following paragraph has no workaround, which is why SSLv3 needs to be avoided completely.

I. Introduction

POODLE, or Padding Oracle On Downgraded Legacy Encryption is a CBC originated attack that allows an attacker to steal "secure" HTTP cookies, or even worse, tokens from a HTTP request such as authorization tokens from the header.

This vulnerability affects every piece of software that can be forced into communicating with SSLv3. This means that any software that implements a fallback mechanism that includes SSLv3 support is vulnerable and can be exploited.

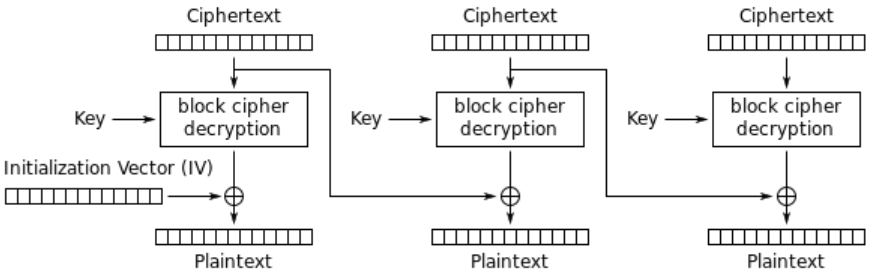

Fig. 1: CBC model for decryption

The CBC cipher encrypts and decrypts plaintext or ciphertext block by block. As depicted above, CBC decryption XORs each plaintext block with the previous ciphertext block. The mathematical formula for CBC decryption is:

$$P_i = D_K(C_i) \oplus C_{i-1}$$

$$C_0 = IV$$

II. Related work

CRIME (Compression Ratio Info-leak Made Easy) is a security exploit against secret web cookies over connections using the HTTPS and SPDY protocols that also use data compression.[1] When used to recover the content of secret authentication cookies, it allows an attacker to perform session hijacking on an authenticated web session, allowing the launching of further attacks.

BREACH (a backronym: Browser Reconnaissance and Exfiltration via Adaptive Compression of Hypertext) is a security exploit against HTTPS when using HTTP compression. BREACH is built based on the CRIME security exploit.

While the CRIME attack was presented as a general attack that could work effectively against a large number of protocols, only exploits against SPDY request compression and TLS compression were demonstrated and largely mitigated in browsers and servers. The CRIME exploit against HTTP compression has not been mitigated at all, even though the authors of CRIME have warned that this vulnerability might be even more widespread than SPDY and TLS compression combined.

III. Attack description

A possible attack on this cipher can be described by using this paper [1] as follows: A client wants to connect to a server using CBC as a method of security. The server will respond for every block if there is a valid padding. As an attacker, the only information we know is correct is the of the server. In a more theoretical way, the server is called an Oracle (you know this gives you the correct answer, but you don't know why, or how the server is choosing a response).

For any block of the ciphertext, C, as a series of words w1, w2 ... wn, we pick a series of random words r1, r2 ... rn instead of that. Starting with rn, and finishing with r1 the attacker will try to find the words in this order. He firstly tries to find rn. By testing all the characters in the ASCII code, he discovers one where the oracle tells it has valid padding. The plaintext behind rn is rn XOR 1 (it gave a correct padding with padding 1). After finding rn, the attacker proceeds next with rn-1, and so on, until he deciphers the plaintext behind the block C.

In SSL 3.0, the ciphertext uses a Message Authentication Code (MAC) to check the integrity behind the received message to decipher.

Adapting the CBC attack to POODLE, "the most severe problem of CBC encryption in SSL 3.0 is that its block cipher padding is not deterministic, and not covered by the MAC" [2], meaning that the padding in the ciphertext is not covered by the MAC, leaving the attacker the window to apply the CBC pad attack. In this case, the oracle function is a server response that can be an OK, or a SSL HMAC error.

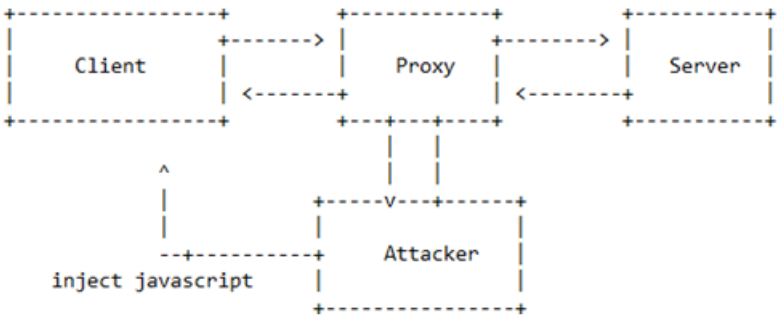

Fig. 2: Man-in-the-middle POODLE attack workflow

Briefly explained, POODLE, as a man-in-the- middle attack can be summarized with the explanations below.

Following Figure 2, in a very practical way, the attacker runs a Javascript agent on **http://example.com** to get the victim's browser, and send different HTTPS requests which include cookies to **http://example.com** with the scope of intercepting the victim's traffic (the attacker can use a proxy server that has become already the proxy server for the victim's browser with the help of the Javascript agent for that matter) and modify the SSL records that come from the browser, so that he can apply the CBC attack on any block of the encrypted HTTPS request and get relevant data.

As mentioned above, every time the server gets a wrong padding, it sends back to the client a SSL HMAC. If the padding is correct, then the server will transmit an OK message as a response to the request. This property becomes in this case an oracle function.

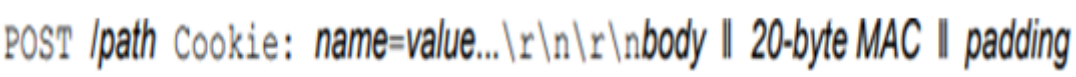

Fig. 3: HTTP request format

IV. Proof-of-Concept description and implementation

For this project, according to Figure 2, we created a TCP client, and 2 TCP servers, one of them being a secure HTTP server, and the other being the attacker's Proxy server

Firstly, we simulated from the code that the attacker uses a JavaScript agent to redirect the traffic of the client to the Proxy, by making the Proxy accept 2 types of connections: one with the client, and one with the secure server. The client tries to send to the server a pseudo HTTP request containing a cookie, a username, and a password, using a SSL 3.0 type of connection (keep in mind that if the client, or the server don't support a SSL 3.0 connection this attack would not work), as shown in Figure 3. In a more practical way, this could mean a request of authentication to a server.

The attacker's main objective in this case is to obtain the client's username and password for finally breaking into the server. After the client sends a request, the attacker captures the ciphertext and tries to apply the CBC pad attack. Every time the server gets a wrong pad, it will transmit a SSL HMAC error. The client tries to send the request until he sees an OK response.

The Proxy needs to client to transmit this request until the information the attacker wants is available, so he will continue to transmit a HMAC error until the attacker finishes applying the POODLE attack on the server.

V. Results

In practice, a server can limit the number of attempts a client has to send correctly a request (this may limit the attacker's capabilities to decipher all the plaintext from the request), that is why by applying the property of the CBC pad as being a block cipher, an attacker can apply the CBC pad attack just on a single block, not all of them.

By knowing the format of the request, and by seeing its length, the man in the middle can easily deduce which blocks of the cipher to attack in order to gain the information he wants.

We added in the implementation this kind of support, so now the attacker will just extract from the request the blocks of ciphertext that contain the username and the password of the client. This will reduce significantly the number of failed requests to the server, as well as the time to gain the necessary information.

Finally, the attacker gains the username and the password of the client, and can connect to the server, as shown below.

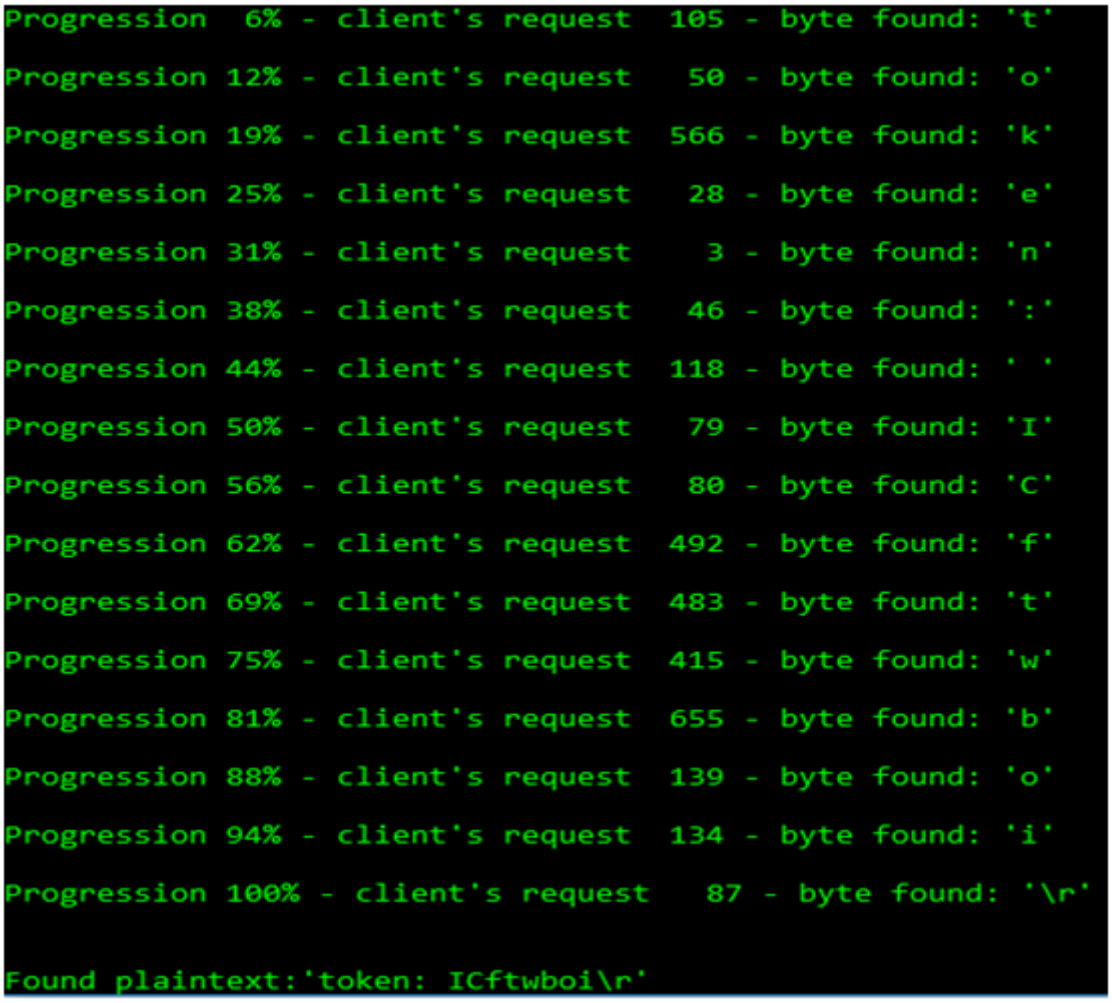

Fig. 4: POODLE proof of concept

Paper [1] explains that "On average, we have to try W/2values.", W is the number of possible words for an ri, i = [1 … n]. In this case the CBC block length is 16 bytes, and the HTTP request has 9 plaintext blocks. Each ri has 256 maximum values, so that could make a approximate 19000 requests to the server. By applying the block property of the CBC pad cypher, and by knowing that the username and the password of the client are found in 2 blocks, this reduces the number of requests to 4200, reducing the time needed significantly.

References

[1] Serge Vaudenay, Swiss Federal Institute of Technology (EPFL), Serge.Vaudenay@epfl.ch, Security Flaws Induced by CBC Padding – Applications to SSL, IPSEC, WTLS