CS 427, Final Project
POODLE
Padding Oracle on Downgraded Legacy Encryption

Cody Malick
malickc@oregonstate.edu

March 13, 2017

Abstract

This paper outlines the POODLE exploit. It gives details on how it allows an adversary to repeatedly query a server using SSL version 3.0, and eventually decrypt a block one byte at a time. The exploit was first published by the Google Security Team on October 14, 2014 by Bodo Moller, Thai Duong, and Kryzsztof Kotowicz. While SSL version 3.0 is quite old, it is still in use today for browser backward compatibility. Later, an updated version of the POODLE exploit was published showing successful attacks against TLS.

Introduction

POODLE is an attack on Secure Socket Layer version 3.0 that was made public in 2014. It was announced on Google's Security Blog, co-published by Bodo Moller, Thai Duong, and Kryzsztof Kotowicz. POODLE allows an attacker targeting SSL 3.0 to decrypt one byte of an encrypted SSL block one out of two-hundred fifty-six attempts. While this is an average, being able to decrypt a byte of an encrypted block that quickly is quite a security flaw.[1] It was later found that the vulnerability extended to TLS version 1.0. It is more of an implementation problem. Some implementations of TLS in popular software did not correctly check the padding after decryption. [2]

The goal of SSLv3 was to encrypt and keep secret communication between a web browser, and a web server. This is accomplished through an initial secret-sharing hand shake. It then computes a MAC from the shared secret. Once message exchange begins, it uses AES CBC-Mode to encrypt and decrypt messages.[3] The goal of the protocol was to provided secrecy, authenticity, and integrity.

The Attack

Context

To completely understand the attack, the context should first be made clear. SSL version 3.0 is quite dated, and should for all intents and purposes be completely retired. Modern browsers, however, use SSLv3 as a fall back for compatibility purposes. The browser will first try to connect to the web server using TLS version 1.2, the de facto standard for web communication. If that fails, it is default behavior in browsers to fall back to earlier standards, such as SSLv3.[1]

While browsers continue to support SSLv3, this attack will continue to be an issue. A savvy attacker with the proper resources could catch web requests with a man in the middle attack, and prevent all non-SSLv3 requests from going through. This would force the use of SSLv3 if it is supported by the web server.[4]

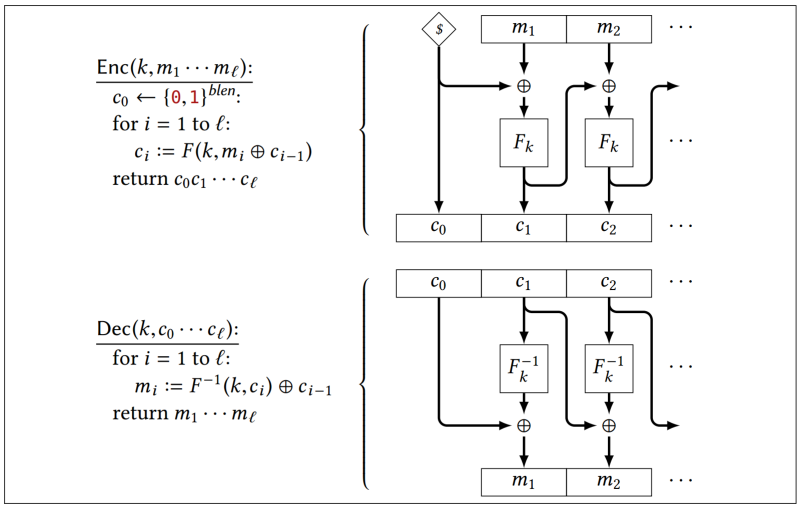
Padding

The primary point of vulnerability that the attack exploits is in the padding of the encrypted message. Padding in SSLv3 is accomplished by measuring the difference between the last block size, and the required size of the block. In this case, the block size must be a multiple of sixteen. In this case, the AES CBC mode block size is sixty-four bits. The SSLv3 implementation then measures the difference d, and fills the space $d - 1$ with random byte values. The last byte is then filled with the size of the padding.[3] SSLv3 does not check

AES CBC Mode

Cipherblock Chaining mode is one of the most popular modes for AES operation. It provides chosen plaintext attack security, and non-deterministic encryption using an initialization vector. Here is a diagram from the class textbook that illustrates how this process works:[5]

Construction 9.2
(CBC Mode)



The key portion of this mode that we will be examining in the attack is that the previous encryption block is used to encrypt the following block.

Exploit

The exploit is accomplished by executing the following steps:

1.

The added layer of difficulty to this attack is that the attacker must have two requirements in place: the attacker must control some part of the client side connection, and the attacker must have visibility of the resulting ciphertext. These two pre-requisites require a fairly compromised system to begin with.[6]

The Fix

Bibliography

References

[1] K. K. Bodo Moller, Thai Duong. (2014, October) This poodle bites: Exploiting the ssl 3.0 fallback. [Online]. Available: https://www.openssl.org/ bodo/ssl-poodle.pdf

[2] I. Ristic. (2014, December) Poodle bites tls. [Online]. Available: https://blog.qualys.com/ssllabs/2014/12/08/poodle-bites-tls

[3] P. K. A. Freier, P. Karlton. (2011, August) The secure sockets layer (ssl) protocol version 3.0. [Online]. Available: https://tools.ietf.org/html/rfc6101

[4] R. Barnes. (2014, October) The poodle attack and the end of ssl 3.0. [Online]. Available: https://blog.mozilla.org/security/2014/10/14/the-poodle-attack-and-the-end-of-ssl-3-0/

[5] M. Rosulek, *The Joy of Cryptography.* School of Electrical Engineering and Computer Science, Oregon State University, 2017, vol. Draft, January 3, 2017.

[6] U. S. C. E. R. Team. (2014, October) Ssl 3.0 protocol vulnerability and poodle attack. [Online]. Available: https://www.us-cert.gov/ncas/alerts/TA14-290A