CRIME Attack

OHAD AGADI and YUVAL LIHOD, Tel-Aviv University, israel

This report provides an overview of the CRIME attack against TLS/SSL protocols and describes the exploit with an End-To-End attack demonstration. The demonstration simulates a real-world scenario where a malicious actor has successfully performed a cross-site-scripting attack (XSS) and has access to a vulnerable browser as an 'Encryption Oracle', setting the grounds for Chosen-Plain-text-Attacks (CPA).

ACM Reference Format:
Ohad Agadi and Yuval Lihod. 2018. CRIME Attack. *ACM Trans. Graph.* 37, 4, Article 111 (August 2018), 2 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

The **CRIME** CVE-2012-4929 [CRI 2012] attack is one that allows extraction of secret session-tokens from vulnerable browsers using TLS <1.2. It uncovers a broad class of attacks, that utilizes compression - induced - vulnerabilities as side-channel attack vectors, to extract sensitive information. The attack is performed on the assumption of existing 'Cross-Site-Scripting' (XSS) attack/privileges (victim is forced to send arbitrary requests). The malicious actor uses a vulnerable browser (the victim) as a 'Compression Oracle', making educated guess of the compressed & encrypted session token, and sniffing the length to follow up. The work presented here uses the assumption of a 'Compress+Encrypt Oracle', to better shine light on CRIME. A full setup of the XSS part is presented in our work on the **BEAST** attack, which shares the same setup requirements.

2 Related Work

Fundamentals and descriptions of **CRIME** are explained by [CRI 2014].
Implementation is derived from https://github.com/mpgn/CRIME-poc/blob/master/CRIME-rc4-poc.py and https://shainer.github.io/crypto/2017/01/02/crime-attack.html

3 Challenge Description

This CTF challenge focuses on implementing the CRIME (Compression Ratio Info-leak Made Easy) attack. The challenge demonstrates how compression leaks information about encrypted data.

4 Setup

- A vulnerable client browser, using TLS <1.2.
- An attacker with the ability to sniff outbound networking of the client, and an XSS abilty (can force the client to send chosen data to chosen destination).
- Encryption type - with SSL all communication is encrypted, for out purposes the algorithm is either RC4 or CBS-AES. That induces two setup modes, and two difficulty levels for the challenge.

5 Challenge Modes

5.1 Normal Mode

In the normal mode, the server uses RC4 stream cipher for encryption. This mode is more straightforward because:

- RC4 doesn't use padding
- The ciphertext length exactly matches the plaintext length

5.2 Hard Mode

The hard mode uses AES in CBC mode, which introduces additional complexity:

- Requires padding, ciphertext length is rounded to the next 16-byte block.
- Challengers must account for padding.

6 Challenge Components

6.1 organizer.py

The organizer script serves as an encryption oracle over HTTP. Key features:

- Listens on URL:443
- Accepts POST requests with plaintext data
- Compresses the data using zlib
- Encrypts using either RC4 or AES-CBC based on mode
- Returns the encrypted data.

6.2 oracle_demonstration.py

This script provides examples of interacting with the encryption oracle:

- Demonstrates proper HTTP request formatting
- Shows how to interpret the server responses
- Includes examples of payload construction

Example interaction:

```
def demonstrate_oracle():
    payload = "secret_flag: CTF_FLAG{b"
    encrypted_length = self.get_response_length(payload
```

6.3 exploit.py

Template for challengers to implement their solution:

- Provides basic structure for the attack

7 Attack Implementation

7.1 Core Concept

The CRIME attack exploits the fact that when two strings share a common substring, the compressed size will be smaller than if they were completely different. This allows an attacker to guess secret

Authors' Contact Information: Ohad Agadi, ohadagadi@mail.tau.ac.il; Yuval Lihod, yuvallihod@mail.tau.ac.il, Tel-Aviv University, israel.

ACM 1557-7368/2018/8-ART111
https://doi.org/XXXXXXX.XXXXXXX

values one byte at a time by observing changes in the encrypted data length.

7.2 Implementation Steps

To successfully complete the challenge, participants should:

(1) Implement the main attack loop:
   - Generate candidate guesses
   - Send payloads to the oracle
   - Compare encrypted lengths
   - Select the best candidate based on compression ratio
(2) Handle different encryption modes:
   - Handle block alignment in AES-CBC mode

8 Conclusion

The core fundamentals of CRIME explained above, and the CTF challenge presented should give the challenger a good understanding of this class of attacks. The inclusion of both NORMAL and HARD modes is significant to the challenger's understating of the implementation details and intricacies of the CRIME attack.

References

2012. *CRIME SSL/TLS attack.* https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2012-4929
2014. *CRIME SSL/TLS attack.* https://securityalley.blogspot.com/2014/07/ssltls-crime.html