TLS-Attacker

Systematic Fuzzing and Testing of TLS Libraries

Juraj Somorovsky
Transport Layer Security

- The most important crypto protocol
- HTTP, SMTP, IMAP ...
Secure Sockets Layer (SSL), SSLv2

SSLv3

Transport Layer Security

1995

Wagner, Schneier: Analysis of SSLv3

Bleichenbacher’s attack

Padding oracle attack

2000

2005

TLS 1.1

TLS 1.2

BEAST, CRIME, BREACH, Lucky 13

2010

2015

TLS 1.3
Questions

- How can we test these attacks?
- Can we find such attacks automatically?
Approach [SP2-17]

1. Collect TLS libraries
2. 
3. Profit
Approach [SP2-17]

1. Collect TLS libraries
2. 
3. Profit
Approach [SP2-17]

1. Collect TLS libraries
2. 
3. Profit

Bounty Qualification

The project maintainers have final decision on how to reward researchers who succeed in this challenge. Please respect their decision, and we ask that you follow the reward determination by the project.

- **Critical**: $5,000+
- **High**: $2,500
- **Moderate**: $1,000
- **Low**: $500
Approach [SP2-17]

1. Collect TLS libraries

2. 

3. Profit
Contributions

- Flexible TLS framework
- Fuzzing, testing, writing attacks ...
- High impact vulnerability in OpenSSL
- Additional vulnerabilities in Botan, MatrixSSL...

• [https://github.com/RUB-NDS/TLS-Attacker](https://github.com/RUB-NDS/TLS-Attacker)
Overview

1. TLS Protocol
2. Framework Prerequisites
3. TLS-Attacker Design
4. Results
5. Conclusions
TLS RSA Handshake

ClientHello

ServerHello

Certificate

ServerHelloDone

ClientKeyExchange

ChangeCipherSpec

(Client-) Finished

ChangeCipherSpec

(Server-) Finished

Application

Application
TLS is complex ...

• Different versions
• Crypto primitives: RSA, EC, AES, 3DES, RC4, Chacha, Poly1305, New Hope
• Extensions
• Protocol flows
TLS is complex ...

ClientHello

ServerHello

Certificate

ServerKeyExchange

ServerHelloDone

CertificateVerify

ClientKeyExchange

ChangeCipherSpec

(Client-) Finished

ChangeCipherSpec

(Server-) Finished

Heartbeat

Heartbeat

Juraj Somorovsky. TLS-Attacker
Recent Attacks on TLS

• Not only crypto attacks ...

• Attacks on TLS state machines
  – FREAK
  – Early CCS

• Buffer overflows / overreads
  – Heartbleed
  – CVE-2016-6307 (High) -> CVE-2016-6309 (Critical)

• Tool for flexible protocol executions needed
Overview

1. TLS Protocol
2. Framework Prerequisites
3. TLS-Attacker Design
4. Results
5. Conclusions
Framework Prerequisites

- Flexible protocol flow definition
- Message modifications
- Invalid behavior detection
- Protocol flow reproduction
Overview

1. TLS Protocol
2. Framework Prerequisites
3. TLS-Attacker Design
4. Results
5. Conclusions
High-Level Overview

Modules

- Attacks
- ModifiableVariable
- TLS
- Transport
- Fuzzer

Packages

- crypto
- protocol
- record
- workflow
- alert
- application
- heartbeat
- handshake
Modifiable variables

• Define basic data types (integer, byte, arrays) with modifications

• Example:

```java
ModifiableInteger i = new ModifiableInteger();
i.setValue(30);
i.setModification(new AddModification(20));
System.out.println(i.getValue()); // 50
```

• Further modifications: xor, shuffle, delete, ...
Protocol messages

• ClientHello

<table>
<thead>
<tr>
<th>ClientHelloMessage</th>
</tr>
</thead>
<tbody>
<tr>
<td>cipherSuites: ModifiableByteArray</td>
</tr>
<tr>
<td>cipherSuiteLength: ModifiableInteger</td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td>getCipherSuites()</td>
</tr>
<tr>
<td>getCipherSuiteLength()</td>
</tr>
</tbody>
</table>

• Stored in a message list
• Serializable in XML
Defining a protocol flow

<protocolMessages>
  <ClientHello>
    <supportedCipherSuites>
      <CipherSuite>TLS_RSA_WITH_AES_128_CBC_SHA</CipherSuite>
    </supportedCipherSuites>
  </ClientHello>
  <ServerHello/>
  <Certificate/>
  <ServerHelloDone/>
  <RSAClientKeyExchange/>
  <Application/>
  <ChangeCipherSpec/>
  <Finished/>
  <ChangeCipherSpec/>
  <Finished/>
</protocolMessages>
Defining a protocol flow

<protocolMessages>
  <ClientHello>
    <supportedCipherSuites>
      <CipherSuite>TLS_RSA_WITH_AES_128_CBC_SHA</CipherSuite>
    </supportedCipherSuites>
  </ClientHello>
  <ServerHello/>
  <Certificate/>
  <ServerHelloDone/>
  <RSAClientKeyExchange/>
  <ChangeCipherSpec/>
  <Finished/>
  <ChangeCipherSpec/>
  <Finished/>
  <Heartbeat/>
</protocolMessages>
TLS-Attacker used for...

- Attacks
- Fuzzing (only server, sorry)
- Test suite
Overview

1. TLS Protocol
2. Framework Prerequisites
3. TLS-Attacker Design
4. Results
5. Conclusions
Results

• Padding oracle attack
  – OpenSSL (CVE-2016-2107)
  – Botan 1.11.21 (CVE-2015-7824)
  – MatrixSSL 3.8.2

• Bleichenbacher attack
  – MatrixSSL 3.8.2

• Missing length checks
  – GnuTLS 3.4.9
  – OpenSSL 1.0.1

• Out-of-bound reads / writes
  – OpenSSL-1.1.0-pre1 (stack overflow)
  – Botan 1.11.28 (Out-of-bound read)
Overview

1. TLS Protocol
2. Framework Prerequisites
3. TLS-Attacker Design
4. Results
5. Conclusions
Conclusions and future work

• Maintaining a crypto library is hard
• Systematic fuzzing and evaluation needed
• TLS-Attacker
  – For researchers, pentesters
  – For developers
  – Integrated in Botan and MatrixSSL
• Development / fuzzing improvements needed
  – TLS client-side tests
  – Better fuzzing strategies
Questions

More security research talks?

4.5. – 5.5. 2017

Non-profit security conference