Formal Verification of Real-World Security Protocols

SSL

Zhang Mingmin - 2 October 2017

**Introduction**

**MITM**

****

**Data**

Setup:

Server IP: 192.168.0.115

Client IP: 192.168.0.119

Attacker IP: 192.168.0.113

Server and client were running windows 7, and attacker was running Kali with bettercap as the attack tool.

*Normal Data*

TLS1.0 captured from server

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Time** | **Source** | **Destination** | **Protocol** | **Length** | **Src port** | **Dest port** | **Info** |
| 2017-10-11 13:28:02.953952 | 192.168.0.119 | 192.168.0.115 | TLSv1 | 158 | 49290 | 443 | Client Hello |
| 2017-10-11 13:28:02.954620 | 192.168.0.115 | 192.168.0.119 | TLSv1 | 882 | 443 | 49290 | Server Hello, Certificate, Server Hello Done |
| 2017-10-11 13:28:02.955152 | 192.168.0.119 | 192.168.0.115 | TLSv1 | 380 | 49290 | 443 | Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message |
| 2017-10-11 13:28:02.960080 | 192.168.0.115 | 192.168.0.119 | TLSv1 | 113 | 443 | 49290 | Change Cipher Spec, Encrypted Handshake Message |

The process can be described as follow:

C -> Client Hello -> S

S -> Server Hello, Certificate, Server Hello Done -> C

C -> Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message -> S

S -> Change Cipher Spec, Encrypted Handshake Message -> C

C -> Data -> S

*Data with Attack*

TLS1.0 capture from client

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| 0.406516 | 192.168.0.119 | 192.168.0.115 | TLSv1 | 190 | 61640 | 443 | Client Hello |
| 0.406744 | 192.168.0.115 | 192.168.0.119 | TCP | 54 | 443 | 61640 | 443 → 61640 [ACK] Seq=1 Ack=137 Win=30336 Len=0 |
| 0.406928 | 192.168.0.115 | 192.168.0.119 | TLSv1 | 199 | 443 | 61640 | Server Hello, Change Cipher Spec, Encrypted Handshake Message |
| 0.407194 | 192.168.0.119 | 192.168.0.115 | TLSv1 | 113 | 61640 | 443 | Change Cipher Spec, Encrypted Handshake Message |
| 0.409969 | 192.168.0.119 | 192.168.0.115 | TLSv1 | 667 | 61640 | 443 | Application Data |

TLS1.0 captured form server

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |  |  |
| 0.765827 | 192.168.0.113 | 192.168.0.115 | TLSv1 | 285 | 35804 | 443 | Client Hello |
| 0.765988 | 192.168.0.115 | 192.168.0.113 | TLSv1 | 894 | 443 | 35804 | Server Hello, Certificate, Server Hello Done |
| 0.766386 | 192.168.0.113 | 192.168.0.115 | TCP | 66 | 35804 | 443 | 35804 → 443 [ACK] Seq=220 Ack=829 Win=32128 Len=0 TSval=3563971882 TSecr=10615785 |
| 0.766757 | 192.168.0.113 | 192.168.0.115 | TLSv1 | 392 | 35804 | 443 | Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message |
| 0.768675 | 192.168.0.115 | 192.168.0.113 | TLSv1 | 125 | 443 | 35804 | Change Cipher Spec, Encrypted Handshake Message |
| 0.779995 | 192.168.0.113 | 192.168.0.115 | TLSv1 | 684 | 35804 | 443 | Application Data |

The process can be described as follow:

C -> Client Hello -> S(A)

S(A) -> Server Hello, Change Cipher Spec, Encrypted Handshake Message -> C

C -> Change Cipher Spec, Encrypted Handshake Message -> S(A)

C -> Data -> S(A)

A -> Client Hello -> S

S -> Server Hello, Certificate, Server Hello Done -> A

A -> Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message -> S

S -> Change Cipher Spec, Encrypted Handshake Message -> A

A -> Data -> S

We can observe that when the attack was carried out, the client still thought it was talking with the server, while the truth is the attacker has hijacked client’s traffic and negotiated with the server on its own TLS session. The TLS version the attacker used may not be the same with client’s initial request version.