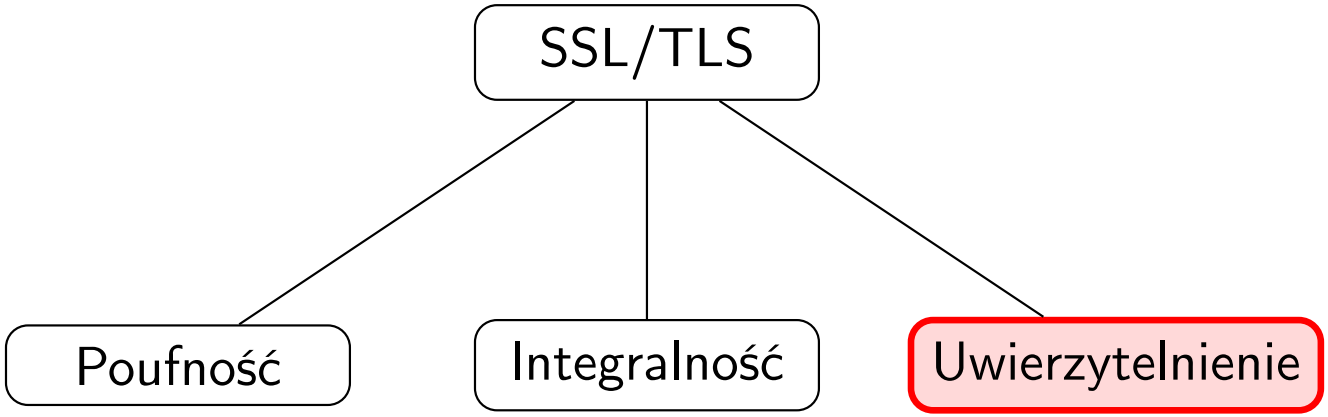
Kryptoanaliza stosowana

Frankencerts - Jak testować certyfikaty SSL/TLS

Rafał Leja (UWr)

26 listopada 2025

SSL/TLS



Rysunek: SSL/TLS gwarantuje poufność, integralność i uwierzytelnienie.

SSL/TLS - uwierzytelnienie

- SSL/TLS używa certyfikatów X.509
- Certyfikaty są wydawane przez zaufane podmioty zwane Urzędami Certyfikacji (CA).
- Przeglądarka posiada listę zaufanych CA
- Certyfikat jest zweryfikowany jeśli:
  - jest ważny
  - został wydany przez zaufanego CA

SSL/TLS - certyfikaty X.509

- Certyfikat X.509 zawiera (między innymi):
  - Nazwę podmiotu (np. nazwę domeny).
  - Klucz publiczny podmiotu.
  - Dane o wydającym CA.
  - Data wystawienia certyfikatu.
  - Okres ważności certyfikatu.
  - Podpis cyfrowy wydającego CA.

SSL/TLS - weryfikacja certyfikatu

- Przeglądarka musi zweryfikować certyfikat domeny, sprawdzając:
  - Czy certyfikat jest podpisany przez zaufanego CA.
  - Czy nazwa domeny w certyfikacie pasuje do odwiedzanej strony.
  - Czy certyfikat nie wygasł.
  - Czy certyfikat nie został unieważniony (CRL, OCSP).
- Trzeba również sprawdzić certyfikaty pośrednie w łańcuchu zaufania.

SSL/TLS - Podatności

- Podatności weryfikacji w SSL/TLS mogą wynikać z:
  - Błędów implementacji weryfikacji SSL/TLS.
  - Słabych algorytmów kryptograficznych (np. RC4, MD5).
- Przy błędniej weryfikacji, atak MitM staje się możliwy, nawet przy użyciu SSL/TLS.

Jak testować poprawność implementacji

- Skąd brać różne certyfikaty do testów?

Jak testować poprawność implementacji

- Skąd brać różne certyfikaty do testów?
  - Prawdziwe certyfikaty z Internetu $\Rightarrow$ Tylko poprwane certyfikaty.
  - Fuzzing $\Rightarrow$ Większość certyfikatów będzie odrzucona na poziomie parsowania.
  - Manualne tworzenie certyfikatów $\Rightarrow$ Czasochłonne.
- Potrzebne jest automatyczne generowanie certyfikatów z kontrolowanymi błędami.

Jak testować poprawność implementacji

- Jak interpretować wyniki testów?

Jak testować poprawność implementacji

- Jak interpretować wyniki testów?
  - Manualna analiza wyników $\Rightarrow$ Czasochłonna i podatna na błędy.
  - Automatyczna analiza wyników $\Rightarrow$ Wymaga bezbłędnego zaimplementowania SSL/TLS.
- Potrzebujemy "Wyroczni" weryfikującej certyfikaty.

Frankencerts

- Frankencerts to podejście do automatycznego testowania implementacji SSL/TLS.
- Składa się z dwóch głównych komponentów:
  - Generatora certyfikatów z kontrolowanymi błędami.
  - Wyroczni do weryfikacji certyfikatów.

Generowanie Frankencerts

- Zbieramy prawdziwe certyfikaty (243,246)

Generowanie Frankencerts

- Zbieramy prawdziwe certyfikaty (243,246)
- Dzielimy je na części składowe (pola, rozszerzenia)

Generowanie Frankencerts

- Zbieramy prawdziwe certyfikaty (243,246)
- Dzielimy je na części składowe (pola, rozszerzenia)
- Miksujemy części z różnych certyfikatów, tworząc nowe certyfikaty z kontrolowanymi błędami

Generowanie Frankencerts

Algorithm 1 Generating a single frankencert

```
1:  procedure FRANKENCERT(certs, exts, issuer)
2:      new_cert ← Create a blank cert
3:      for all field ∈ new_cert do
4:          if field ="key" then
5:              new_cert.key ← Create a random key
6:          else if field ="issuer" then
7:              new_cert.issuer ← issuer
8:          else
9:              random_cert ← CHOICE(certs)
10:             new_cert.field ← random_cert.field
11:         end if
12:     end for
13:     num_exts ←RANDOM(0, 10)
14:     for i ∈ 1..num_exts do
15:         random_id ←CHOICE(exts)
16:         random_val ←CHOICE(exts[random_id])
17:         new_cert.extensions[i].id ← random_id
18:         new_cert.extensions[i].val ← random_val
19:         if RANDOM < 0.05 then
20:             FLIP( new_cert.extensions[i].critical)
21:         end if
22:     end for
23:     SIGN(new_cert, issuer.key)
24:     return new_cert
25: end procedure
```

Generowanie Frankencerts

Algorithm 2 Generating a chain of frankencerts

```
1:  procedure FRANKENCHAIN(certs, ca, length)
2:      issuer ← ca
3:      chain ← ∅
4:      exts ←GETEXTENSIONS(certs)
5:      for i ∈ 1..length do
6:          chain[i] ←FRANKENCERT(certs, exts, issuer)
7:          issuer ← chain[i]
8:      end for
9:      return chain
10: end procedure
```

Generowanie Frankencerts

Rezultat?
8,127,600 przetestowanych certyfikatów z różnymi kombinacjami błędów!

Wyrocznia

- Testowanie "różnicowe"
- Testujemy wiele implementacji SSL/TLS na raz
- Dla każdego wygenerowanego Frankencert, sprawdzamy czy wszystkie implementacje zgadzają się co do jego ważności

Wyrocznia

Testowane implementacje:

- OpenSSL 1.0.1e
- PolarSSL 1.2.8
- GnuTLS 3.1.9.1
- CyaSSL 2.7.0
- MatrixSSL 3.4.2
- NSS 3.15.2
- cryptlib 3.4.0-r1
- OpenJDK 1.7.0 09-b30
- Bouncy Castle 1.49
- Firefox 20.0
- Chrome 30.0.1599.114 p1
- WebKitGTK 1.10.2-r300
- Opera 12.0
- Safari 7.0
- IE 10.0

Wyniki

- 8,127,600 przetestowanych certyfikatów
- Generowanie oraz testowanie certyfikatów: $\approx$ 11.75 dni
- 208 unikatowych kombinacji błędów
- 62,022 certyfikatów powodujących niezgodności

Wyniki

Problem	Certificates triggering the problem occur in the original corpus	OpenSSL	PolarSSL	GnuTLS	CyaSSL	MatrixSSL	NSS	OpenJDK, Bouncy Castle	Browsers
Untrusted version 1 intermediate CA certificate	No	reject	reject	**accept**	reject	**accept**	reject	reject	reject
Untrusted version 2 intermediate CA certificate	No	reject	reject	reject	reject	**accept**	reject	reject	reject
Version 1 certificate with valid basic constraints	No	accept	reject	accept	accept	accept	reject	reject	Firefox: reject Opera, Chrome: accept
Intermediate CA not authorized to issue further intermediate CA certificates, but followed in the chain by an intermediate CA certificate	No	reject	reject	reject	reject	**accept**	reject	reject	reject
...followed by a leaf CA certificate	No	**reject**	**reject**	accept	**reject**	accept	**reject**	**reject**	**reject**
Intermediate CA not authorized to issue certificates for server's hostname	No	reject	reject	**accept**	**accept**	**accept**	reject	reject	reject
Certificate not yet valid	Yes	reject	**accept**	reject	reject	reject	reject	reject	reject
Certificate expired in its timezone	Yes	reject	**accept**	reject	reject	**accept**	reject	reject	reject
CA certificate not authorized for signing other certificates	No	reject	reject	**accept**	**accept**	**accept**	reject	reject	reject
Server certificate not authorized for use in SSL/TLS handshake	Yes	reject	**accept**	**accept**	**accept**	**accept**	reject	reject	reject
Server certificate not authorized for server authentication	Yes	reject	**accept**	**accept**	**accept**	**accept**	reject	reject	reject
Certificate with unknown critical extension	No	reject	reject	**accept**	**accept**	**accept**	reject	reject	reject
Certificate with malformed extension value	No	accept	reject	accept	accept	accept	reject	reject	reject
Certificate with the same issuer and subject and a valid chain of trust	No	reject	reject	**accept**	reject	**accept**	reject	reject	reject
Issuer name does not match AKI	No	reject	accept	accept	accept	accept	reject	reject	reject
Issuer serial number does not match AKI	No	reject	accept	reject	accept	accept	reject	reject	reject

Wyniki

TABLE VI: Error code(s) returned by Web browsers and SSL/TLS libraries for certificates with various combinations of Bad Issuer (I), Expired (E), and Bad Name (N). Security vulnerabilities are highlighted in **bold**.

Certs	Firefox 20	Chrome 30 (Linux)	Opera 12 (Linux)	Opera 20 (Mac)	Safari 7	Chrome 30 (Mac)	IE 10	OpenSSL	PolarSSL	GnuTLS	CyaSSL	MatrixSSL	NSS
E	E	E	E	!E	!E	E	E	E	E	E	E	E	E
I	I	I	I	!I	!I	I	I	I	I	I	I	**	I
IE	IE	**E**	I#	*	**!E**	*	*	I	I	IE	**	**	**E-**
IN	IN	I	I#	!I	!I	I	IN	I-	I-	I-	I-	*-	I-
IEN	IEN	**N**	I#	*	**!E**	*	*	I-	IE-	**-	**-	**-	**E-**
N	N	N	N	+	!N	N	N	-	-	-	-	-	-
NE	NE	N	**E#**	**!E**	**!E**	N	NE	E-	**-	E-	E-	E-	E-

Źródła

- C. Brubaker, S. Jana, B. Ray, S. Khurshid, and V. Shmatikov. Using Frankencerts for automated adversarial testing of certificate validation in SSL/TLS implementations. In 35th IEEE Symposium on Security and Privacy, pages 114–129. IEEE Computer Society, 2014.
- Wikipedia: Transport Layer Security
- Wikipedia: X.509
- Cloudflare: What is a TLS handshake?
- Cloudflare: What is an SSL certificate?