Kodowanie i bezpieczeństwo informacji

Laboratorium - lista nr 3 na 12-23 marca 2012

Pytania, które będą zadawane oddającym dowolne zadanie z tej listy. Nieznajomość odpowiedzi na którekolwiek z poniższych zagadnień może skutkować odebraniem (wszystkich) punktów.

Wyjaśnij w kontekście zadania 4:

- czym jest *Extended validation* dla certyfikatów SSL?
- kto da się nabrać na taki atak?
- czym są CRL, OCSP?
- co się stanie, gdy ktoś pozna klucz tajny serwera www?
- co się stanie, gdy ktoś pozna klucz tajny CA, który podpisywał certyfikat serwera www?
- co się stanie, gdy ktoś pozna klucz tajny jakiegoś CA?
- co się stanie, gdy pewne CA wydaje certyfikaty w oparciu o MD5?

Zadanie 1 (3 pkt) Wykonaj wszystkie czynności:

1. wygeneruj (np. korzystając z *openssl* klucz $\mathcal{A}$ służący do podpisywania (wybierz pomiędzy DSA a RSA), np. odpowiednio zmodyfikuj komendę:

   ```
   openssl genrsa -out privkeyA.pem,
   ```

   pamiętaj aby wybrać długość klucza na poziomie bezpieczeństwa odpowiadającej 128 bitów (NIST SP 800-57).
2. wygeneruj żądanie certyfikatu (CSR - certificate signing request):

   ```
   openssl req -new -key privkeyA.pem -out certA.csr.
   ```

Powtórz czynności $1-3$, generując klucz $\mathcal{B}$, ale tym razem utwórz certyfikat "self signed", tj. zostań *CA - certificate authority*;

```
openssl req -new -x509 -key privkeyB.pem -out CAcert.crt -days 15,
```

a następnie kluczem $\mathcal{B}$ wygeneruj certyfikat dla klucza $\mathcal{A}$ z pliku CSR:

```
openssl x509 -req -days 45 -in certA.csr -CA CAcert.crt -CAkey privkeyB.pem -set_serial 01 -out certA.crt.
```

Zadanie 2 (2 pkt) Wykorzystaj uzyskany certyfikat do podpisania programu z listy 2 (wykorzystaj *jarsigner* bądź jego odpowiednik). Wykorzystaj certyfikat do podpisywania stworzonych przez siebie dokumentów np. OpenOffice/Word/PDF. Być może przyda się:

```
openssl pkcs12 -export -in certA.crt -inkey privatekeyA.pem -certfile CAcert.crt -name "Imie Nazwisko-out ImN.p12
```

Zadanie 3 (3 pkt) Wykonaj wszystkie czynności:

1. Zainstaluj certyfikat odpowiadający kluczowi *B CA* w przeglądarce (w sekcji "authorities"). Nie importuj jednak certyfikatu dla klucza *A*.
2. Na swoim prywatnym serwerze www (może to być Twój komputer) "zainstaluj" klucz *A* i certyfikat (*certA.crt*), aby działał adres np. "https://www.moj.serwer.pl".
3. Spraw, aby przeglądarka nie zgłaszała ostrzeżenia (w zadaniu pierwszym musisz podczas generowania *CSR* wpisać odpowiedni adres (*Common Name*/*hostname*) - może to być np. localhost, albo adres uzyskany przez usługi typu *dynamic dns*).

Zadanie 4 (2 pkt) Stwórz stronę "phishingową" działającą na twoim serwerze (np. laptopie) przechwytującą wprowadzane hasła np. do serwera poczty studenckiej/Gmaila/... wykonaj w tym celu czynności z zadań 1 i 3.

- strona musi działać w oparciu o protokół https,
- przeglądarka ma akceptować certyfikat.

W tym celu możesz odpowiednio zmodyfikować lokalny plik z hostami (*/etc/hosts* bądź jego odpowiednik w systemie, z którego korzystasz).