24.02.2017

Projekt i implementacja algorytmu uwierzytelniania z użyciem zaufanej trzeciej strony na podstawie protokołu BAUER-BERSON-FEIERTAG

Wykonanie:
Jan Kurzydło

Dokumentacja

1. FUNKCJONALNOŚĆ
    a. Uwierzytelniania z użyciem zaufanej trzeciej strony.
    b. Implementacja protokołu BAUER-BERSON-FEIERTAG
2. URURCHAMIANIE
    a. Aby uruchomić program należy włączyć 3 karty w konsoli.
        Następnie wpisać w każdej kolejno polecenie:
        1 karta) ./trent.py
        2 karta) ./bob.py
        3 karta) ./alice.py "Moja wiadomosc" lub ./alice.py
        W pkt 3) pierwsza opcja spowoduje wysłanie naszej wiadomości do Boba podanej w argumencie wykonania,
        druga spowoduje wysłanie do Boba przykładowej wiadomości zapisanej w kodzie programu
        W momencie gdy program poprosi o podanie loginu i hasła należy podać jedna z 3 opcji z pliku users.txt dołączonego do projektu.
3. OPIS DZIAŁANIA
    a. Alicja nawiązuje połączenie z Bobem.
    b. Bob wysyła jej swoje ID oraz wygenerowany NONCE
    c. Alicja wysyła do TRENTA swoje ID i swój NONCE , oraz ID BOBa oraz NONCE BOBa
    d. Trent generuje KLUCZ SESYJNY Alicji i Boba.
    e. Następne wysyła 2 komunikaty do Alicji.
    f. Jeden zaszyfrowany kluczem alicja-trent (session_key, Bob_id, alice_nonce)
    g. Drugi zaszyfrowany kluczem trent-bob (session_key, Alice_id, bob_nonce)

h. Alicja 1 wiadomość deszyfruje za pomocą klucza alicja-ternt 2 wiadomość wysyła do boba.
i. Bob deszyfruje wiadomość kluczem trent-bob.
j. Po deszyfracji Bob i Alicja posiadają KLUCZ SESYJNY
k. Następnie uruchamiany jest proces autentykacji na serwerze Bob'a.
l. Alicja musi podac nazwe uzytkownika i haslo (przykładowe znajdują się w pliku users.txt)
m. Przeprowadzana jest autentykacja.
n. Po udanej autentykacji Alicja wysyła do Boba zaszyfrowana kluczem sesyjnym wiadomość (podana przy uruchamianiu lub przykładowa).
o. Bob po otrzymaniu wiadomości odsyła odpowiedz plus wiadomosc od siebie zaszyfrowana kluczem sesyjnym.
p. Alicja sprawdza poprawność odpowiedzi i zamyka połączenie.

Opis działania modułów

Alice.py:

Moduł klienta Alicji, który umożliwia połączenie się z Bobem.
Zawiera on logikę utworzenia połączenia poprzez socket

```
trent_sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
```

Posiada adres serwerów Boba i Trenta.

```
trent_address = ('localhost', 3000)
```

Funkcje do pobierania danych:

```
def get_response(sock):
    response = sock.recv(512)
    if response:
        return response
```

oraz ich dalszego rozpakowania i przerabiania:

```
def get_id(response):
    params = response.split(',')
    return params[0]

def get_nonce(response):
    params = response.split(',')
    return params[1].strip()

def get_session_key(data):
    params = data.split(',')
    return params[0]

def check_response(response):
```

```
if response == msg + " - Pozdrowienia od Boba!":
    print "Response correct."
else:
    print "Response wrong."
```

Następnie wykonywany jest algorytm opisany w punkcie 3.

AuthenticationEngine.py:

Moduł umożliwiający przeprowadzenie autentykacji.
Dla bezpieczeństwa moduł nie korzysta z jawnie wyświetlanego hasła. Dziła na wartości hash'a danego hasła.
Funkcja `compare_hashes` przeprowadza porównania otrzymanego zahashowanego hasła uzytkownika z zapisanym u siebie hashem hasła, jeżeli jest zgodność tzn że wysłano prawidłowe hasło. Do odpowiedzi zostaje dołożony Nonce (wg algorytmu BBF)

CommunicationModule.py

Moduł wprowadzony do uproszczenia komunikacji oraz programowania w myśl zasady Don't Repeat Yourself.
Otrzymuje w konstruktorze połączenie.
Za pomocą metody `send_response(data)` pozwala na wysyłanie danych, natomiast metoda `get_request` umożliwia odbiór danych.

CryptoEngine.py

Moduł dostarczający mechanizm szyfrowania i rozszyfrowania wiadomości.
Działa na prostej zasadzie podmiany liter w tekście.
Dostępne funkcje to `encrypt(plain_text)` oraz `decrypt(ciphered_text).`

HashEngine.py

Moduł zapewniający generowanie oraz porównywanie Hash'y.
Do generowania wykorzystany został algorytm md5.

Trent.py

Jest to imitacja serwera autentykacyjnego. Zna on klientów którzy mogą się z nim połączyć oraz dzieli z nimi tajemnice która uwierzytelnia obie strony. Działa jak został opisany w punkcie 3. Jego głównym zadanie to potwierdzenie tożsamości klientów oraz wygenerowanie dla nich klucza sesyjnego. Komunikacja odbywa się przez socek natomiast reszta funkcjonalności bazuje na wyżej wymienionych modułach.

Bob.py
Moduł analogiczny do Alice.py

Referencja

Protocols for Authentication and Key Establishment
By Colin Boyd, Anish Mathuria

Link do wybranego fragmentu:
https://books.google.pl/books?id=EEmqCAAAQBAJ&pg=PA88&dq=bauer+berson+feiertag&hl=en&sa=X&ved=0ahUKEwirlPaiq6nSAhWMFywKHb87A48Q6AEIHzAA#v=onepage&q=bauer%20berson%20feiertag&f=false