Zaawansowane techniki programowania

Laboratorium 6

Politechnika Krakowska
im. Tadeusza Kościuszki

Piotr Jurek

15.04.2023
Systemy Inteligentne i Rozszerzona Rzeczywistość

Repozytorium z zadaniem

WPROWADZENIE

W laboratorium omówiono problem transportu danych między procesami za pomocą socketów. Poruszona została też tematyka protokołu SSL/TLS.

MATERIAŁY

Do wykonania zadania wykorzystano dołączone pliki:

- steamer.py
- blocks.py
- features.py

PROCEDURA

1. W tym zadaniu użyto protokołu TCP do wysłania z klienta na serwer kilku wersów tekstu piosenki. Wykorzystano kod wzorowany na pliku streamer.py. Wynik działania programu pokazał, że po uruchomieniu skryptu z argumentem "-c" dane zostały wysłane na serwer, po czym ten wypisał tekst piosenki na standardowe wyjście.

```
> python3 assigment1.py
Run this script in another window with "-c" to connect
Listening at ('127.0.0.1', 1060)
Accepted connection from ('127.0.0.1', 56602)
Received 108 bytes
Received zero bytes - end of file
Message:

I like my toast done on one side
And you can hear it in my accent when I talk
I'm an Englishman in New York
```

2. W drugim zadaniu użyto struktury blokowej do transportu danych, wzorując się na pliku blocks.py. Dane były wysyłane po trochę w porcjach, zamiast próbować wysłać wszystko naraz, jak w poprzednim zadaniu. Wynik działania programu pokazał, że po uruchomieniu tego samego skryptu z parametrem "-c" dane zostały przesłane na serwer w postaci blokowej.

```
> python3 assigment2.py
Run this script in another window with "-c" to connect
Listening at ('127.0.0.1', 1060)
Accepted connection from ('127.0.0.1', 34438)
Block says: b'I like my toast done on one side'
Block says: b'And you can hear it in my accent when I talk'
Block says: b"I'm an Englishman in New York"
```

3. W tym zadaniu zbadano, za pomocą skryptu features.py, moduł SSL, to jest moduł odpowiedzialny za komunikacje zgodna z protokołem Secure Sockets Layer, pozwalającym na lepsze zabezpieczenie połączenia sieciowego. Zapewnia on poufność i integralność danych przesyłanych między aplikacjami. Po uruchomieniu skryptu, na ekranie pokazały się następujące dane:

```
> python3 features.py
------------------------------ protocol ------------------------------
PROTOCOL_SSLv23                     2                              10
PROTOCOL_TLS                        2                              10
PROTOCOL_TLSv1                      3                              11
PROTOCOL_TLSv1_1                    4                              100
PROTOCOL_TLSv1_2                    5                              101
PROTOCOL_TLS_CLIENT                 16                             10000
PROTOCOL_TLS_SERVER                 17                             10001

---------------------------- verify_mode -----------------------------
CERT_NONE                           0                                    0
CERT_OPTIONAL                       1                                    1
CERT_REQUIRED                       2                              10

---------------------------- verify_flags ----------------------------
VERIFY_DEFAULT                      0                                    0
VERIFY_CRL_CHECK_LEAF               4                              100
```

```
VERIFY_CRL_CHECK_CHAIN             12                              1100
VERIFY_X509_STRICT                 32                              100000
VERIFY_ALLOW_PROXY_CERTS           64                              1000000
VERIFY_X509_TRUSTED_FIRST     32768                  1000000000000000
VERIFY_X509_PARTIAL_CHAIN     524288            10000000000000000000

------------------------------ options -------------------------------
OP_NO_SSLv2                        0                                    0
OP_SINGLE_DH_USE                   0                                    0
OP_SINGLE_ECDH_USE                 0                                    0
OP_IGNORE_UNEXPECTED_EOF           128                            10000000
OP_NO_TICKET                  16384                  100000000000000
OP_NO_COMPRESSION             131072                 100000000000000000
OP_ENABLE_MIDDLEBOX_COMPAT    1048576           10000000000000000000
OP_CIPHER_SERVER_PREFERENCE   4194304           100000000000000000000
OP_NO_SSLv3                   33554432          100000000000000000000000
OP_NO_TLSv1                   67108864    1000000000000000000000000
OP_NO_TLSv1_2                 134217728   10000000000000000000000000
OP_NO_TLSv1_1                 268435456   100000000000000000000000000
OP_NO_TLSv1_3                 536870912   1000000000000000000000000000
OP_NO_RENEGOTIATION           1073741824   10000000000000000000000000000
OP_ALL                        2147483728  10000000000000000000000001010000

------------------------ feature availability ------------------------
HAS_NPN                            0                                    0
HAS_SSLv2                          0                                    0
HAS_SSLv3                          0                                    0
HAS_ALPN                           1                                    1
HAS_ECDH                           1                                    1
HAS_NEVER_CHECK_COMMON_NAME        1                                    1
HAS_SNI                            1                                    1
HAS_TLSv1                          1                                    1
HAS_TLSv1_1                        1                                    1
HAS_TLSv1_2                        1                                    1
HAS_TLSv1_3                        1                                    1
```

Wypisane dane opisują działanie biblioteki SSL dostępnej w języku python. Pierwsza sekcja wypisuje listę protokołów obsługiwanych przez standard SSL/TLS. Wartości obok są binarną reprezentacją danych protokołów wykorzystywaną wewnętrznie przez bibliotekę SSL. Dalej widać reprezentacje binarne trzech trybów uwierzytelnienia wykorzystywanych przez agentów w trakcie komunikacji przez

protokół SSL. Jeszcze niżej wypisano flagi, kontrolujące zachowanie się protokołu SSL w czasie uwierzytelniania użytkownika. Kolejna sekcja zawiera opcje kontrolujące zachowanie się protokołu w trakcie komunikacji. Ostatnia sekcja opisuje dostępność różnych funkcjonalności biblioteki SSL.

PODSUMOWANIE

Podsumowując, w tym laboratorium omówiono różne metody transportu danych między procesami, w tym protokół TCP, strukturę blokową oraz zabezpieczanie połączeń sieciowych za pomocą protokołu SSL.