Struktura protokołu MKOI

Artur M. Brodzki, Kuba Guzek

17 stycznia 2018

1 Temat projektu

Kilka ogólnych reguł:

1. Każdy pakiet ma swój SID (*session ID*) o długości 64 bajty. SID jest losowany przez klienta przy zalogowaniu i aż do momentu wylogowania służy jako identyfikator sesji (lista otwartych sesji jest przechowywana przez serwer). Pakiety o nieznanym numerze sesji inne niż REQ-0 (próba zalogowania) są odrzucane bez odpowiedzi.

2. Oprócz tego, każde zapytanie do serwera (np. zapytanie o listę plików w katalogu użytkownika) posiada swój RID (*request ID*) o długości 64 bajty. Odpowiedź na zapytanie o RID = v ma też RID = v.

3. Serwer nasłuchuje prób zalogowania się do systemu na porcie 2314 (są to pierwsze cztery cyfry z hasha SHA-512 ze słów "Diffi-Hellman"). Komunikacja z zalogowanymi użytkownikami odbywa się na innych portach.

4. Protokół Diffiego - Hellmana wymaga ustalenia wspólnej liczby pierwszej p oraz podstawy potęgi g. Przyjmujemy $p = jakas - duza - liczba - pierwsza - majaca - 512 - bitow$ oraz $g = 2$ (bo potęgowanie dwójek jest chyba szybsze niż innych liczb?). Liczba pierwsza po której robimy modulo ma 512 bitów, czyli 64 B i tyle też ma każda liczba używana jako klucz w naszym protokole.

2 Opis wykorzystywanych algorytmów

2.1 Protokoł Diffiego - Hellmana

2.2 Algorytm Serpent

3 Architektura aplikacji - protokół komunikacyjny

3.1 Wstęp

3.2 Logowanie użytkownika

Klient najpierw wysyła swoją nazwę użytkownika. Jeżeli serwer posiada takiego użytkownika w systemie, następuje nawiązanie szyfrowanego połączenia poprzez zastosowanie protokołu Diffiego - Hellmana, a następnie potwierdzenie hasła. Jeżeli takiego użytkownika nie ma w systemie, następuje odmowa połączenia. Jeżeli hasło okaże się nieprawidłowe, również następuje odmowa połączenia.

1. Pakiet REQ-0 (129 B): wysyłany przez klienta w celu zalogowania się do systemu.
   - SID (64 B)
   - REQ-TYPE = 0x00 (1 B)
   - USERNAME (64 B) - 64 znaki ASCII.
2. Pakiet LOGIN-STATUS (1 B): serwer wysyła go w celu potwierdzenia poprawności loginu.
   - LOGIN-FLAG (1B): flaga jest równa 0x00 jeśli logowanie przebiegło poprawnie, lub 0xFF jeśli logowanie nie powiodło się. Protokół znajduje się nad TCP, zakładamy więc, że odebrane dane są zawsze poprawne i nie może wystąpić inna wartosć flagi jak 0x00 | 0xFF.
3. Pakiet DH-1 (64 B): Jeśli logowanie powiodło się, serwer losuje liczbę a i odpowiada klientowi swoim sekretem $g^a \ mod \ p$.
   - DH-SERVER-SECRET (64 B)
4. Pakiet DH-2 (128 B): klient losuje liczbę b i odpowiada serwerowi swoim sekretem $g^b \ mod \ p$.
   - SID (64 B)
   - DH-CLIENT-SECRET (64B)

   W tym momencie serwer i klient posiadają wspólny klucz szyfrowania symetrycznego, równy $g^{a\,b} = g^{b^a}$. Pozostałe komunikaty w ramach sesji są szyfrowane serpentem z użyciem tego klucza.

5. Pakiet PASWD-1 (128 B): klient wysyła serwerowi wartość funkcji SHA-512 z hasła.
    - SID (64 B)
    - PASSWD (64 B)
6. Pakiet LOGIN-STATUS (65 B): serwer wysyła go w celu potwierdzenia poprawności hasła. Struktura identyczna jak w 2.

3.3 Listowanie zawartości katalogu użytkownika

Klient wysyła zapytanie o listę plików w swoim katalogu. Serwer wysyła odpowiedź.

1. Pakiet REQ-1 (131 B): klient wysyła zapytanie o listę plików w swoim katalogu oraz port, na którym jest gotów odebrać listę.
    - SID (64 B)
    - RID (64 B)
    - REQ-TYPE = 0x01 (1 B)
    - PORT (2 B)
2. Pakiet LEN-1 (72 B): serwer wysyła długość listy plików w bajtach.
    - RID (64 B)
    - LENGTH (8 B)

    W tym momencie serwer nawiązuje nową sesję na porcie *PORT* klienta i wysyła tam *LENGTH* bajtów danych, zawierających JSON-a z listą plików. Każdy plik to obiekt JSON-a zawierający pola:
    - NAME
    - SIZE
    - LAST-MODIFICATION
    - HASH - wartość funkcji SHA-512 z pliku

3.4 Dodawanie pliku na serwer

Klient wysyła prośbę o wysłanie pliku na serwer. Serwer sprawdza, czy plik o takiej nazwie znajduje się już w katalogu użytkownika. Jeśli nie, wysyła klientowi pozwolenie na wysyłanie pliku wraz z numerem portu, na którym będzie przebiegać wysyłanie. Po otrzymaniu pozwolenia od serwera, klient rozpoczyna wysyłanie pliku.

1. Pakiet REQ-2 (193 B): klient wysyła prośbę o pozwolenie na dodanie pliku do serwera.

- SID (64 B)
- RID (64 B)
- REQ-TYPE = 0x02 (1 B)
- NAME (64 B) - 64 znaki ASCII

2. Pakiet RES-2 (67 B) : serwer wysyła zgodę wraz z numerem portu, na którym serwer jest gotów odebrać plik, lub brak zgody.

   - RID (64 B)
   - PERM-FLAG (1 B) - równe 0x00, jeśli plik może zostać wysłany na serwer, lub 0xFF jeśli plik nie może zostać wysłany na serwer.
   - PORT (2 B)

3. Pakiet LEN-1 (72 B): klient wysyła rozmiar wysyłanego pliku. Struktura identyczna jak w 2

   W tym momencie klient nawiązuje nową sesję TCP na porcie $PORT$ serwera i wysyła tam $LENGTH$ bajtów danych zawierających dodawany plik.

3.5 Pobieranie skrótu pliku z serwera

Klient wysyła prośbę o wysłanie skrótu (SHA-512) pliku o zadanej nazwie. Serwer odsyła żądany skrót, lub 0, gdy takiego pliku nie ma na serwerze.

1. Pakiet REQ-3 (193 B): klient wysyła prośbę o skrót zadanego pliku.

   - SID (64 B)
   - RID (64 B)
   - REQ-TYPE = 0x03 (1 B)
   - NAME (64 B)

2. Pakiet RES-3 (129 B): serwer odpowiada skrótem pliku, o ile plik istnieje.

   - RID (64 B)
   - EXISTS-FLAG (1 B) - równe 0x00, jeśli plik znajduje się na serwerze, lub 0xFF jeśli pliku brak
   - HASH (64 B) - równe skrótowi pliku, jeśli plik znajduje się na serwerze, lub 0x0...0 jeśli pliku brak.

3.6 Pobieranie pliku z serwera

Klient wysyła prośbę o pobranie pliku z serwera. Serwer sprawdza, czy plik o takiej nazwie znajduje się w katalogu użytkownika. Jeśli tak, wysyła klientowi pozwolenie na pobranie pliku wraz z numerem portu, na którym będzie przebiegać transmisja. Po otrzymaniu pozwolenia od serwera, rozpoczyna się pobieranie pliku.

1. Pakiet REQ-4 (193 B): klient prosi o możliwość pobrania pliku z serwera.
   - SID (64 B)
   - RID (64 B)
   - REQ-TYPE = 0x04 (1 B)
   - NAME (64 B)
2. Pakiet RES-4 (65 B): serwer odpowiada zgodą, o ile plik istnieje oraz długością przesyłanego pliku.
   - RID (64 B)
   - EXISTS-FLAG (1 B) - równe 0x00, jeśli plik istnieje, lub 0xFF jeśli pliku brak.
   - LENGTH (8 B)
3. Pakiet PORT-1 (66 B): klient przesyła serwerowi port, na którym jest gotów odebrać plik.
   - RID (64 B)
   - PORT (2 B)

   W tym momencie serwer nawiązuje nową sesję TCP na porcie *PORT* klienta i wysyła tam *LENGTH* bajtów danych zawierających pobierany plik.

3.7 Usuwanie pliku

Klient wysyła prośbę o wysłanie wskazanego pliku z serwera. Serwer odpowiada potwierdzeniem, jeśli plik istnieje i został usunięty.

1. Pakiet REQ-5 (193 B): klient wysyła prośbę o usunięcie wskazanego pliku.
   - SID (64 B)
   - RID (64 B)
   - REQ-TYPE = 0x05 (1 B)
   - NAME (64 B)
2. Pakiet RES-5 (72 B): serwer wysyła potwierdzenie usunięcia pliku lub stwierdza, że pliku nie było na serwerze.
   - RID (64 B)
   - DELETE-FLAG (1 B) - równe 0x00, jeśli plik został poprawnie usunięty, lub 0xFF jeśli pliku o zadanej nazwie nie było na serwerze.

3.8 Wylogowanie

Klient wysyła prośbę o wylogowanie z serwera i tym samym usunięcie identyfikatora sesji.

1. Pakiet REQ-6 (129 B) - klient wysyła prośbę o wylogowanie z serwera.
    - SID (64 B)
    - RID (64 B)
    - REQ-TYPE = 0x06 (1 B)

4 Stworzona aplikacja

4.1 Serwer

//TODO Kuba

4.2 Klient

4.3 Testy

5 Podsumowanie