Zdalna praca — protokół SSH

- TELNET nie zapewnia szyfrowania (ani przy przesyłaniu hasła, ani przy dalszej pracy);
- TELNET nie pozwala na „logowanie się” bez podania hasła;
- TELNET nie pozwala na posyłanie poleceń do wykonania na odległej maszynie bez logowania się;
- TELNET nie ma wyrafinowanych metod uwierzytelniania;
- TELNET nie zapewnia kompresji połączenia;
- punkt 2. i 3. można rozwiązać używając „poleceń na `r`” (`rlogin`, `rsh`), jednak nie rozwiązują one pozostałych problemów;
- obecnie panaceum zapewnia protokół SSH.

Ogólna charakterystyka protokołu

- klient i serwer komunikują się za pomocą TCP (implementacja używa gniazd);
- fazy działania protokołu
  - wymiana numerów wersji;
  - przesłanie kluczy serwera klientowi;
  - przesłanie zaszyfrowanego klucza klienta serwerowi;
  - weryfikacja tożsamości klienta;
  - przygotowanie sesji;
  - sesja;
  - zamknięcie sesji i odesłanie kodu zakończenia klientowi.

- po otrzymaniu zgłoszenia klienta, serwer wysyła swoją identyfikację (numer wersji); klient odpowiada swoim numerem wersji; identyfikacja wersji jest czytelna dla człowieka (napisy);
- po sprawdzeniu wersji następują dalsze negocjacje (pakiety binarne); serwer wysyła klucz RSA swojej maszyny, swój klucz RSA (zmieniany co godzinę) i inne dane; klient generuje 256–bitowy klucz sesji, szyfruje go za pomocą obydwu kluczy RSA i odsyła go (wraz z informacją o wybranym protokole szyfrowania serwerowi); dalsza komunikacja jest w całości szyfrowana za pomocą wybranego klucza i algorytmu;

- następną czynnością jest weryfikacja tożsamości klienta (użytkownika wywołującego klienta protokołu SSH); może ona (w zależności od konfiguracji serwera) być oparta na użyciu pliku `.rhosts` i/lub `/etc/hosts.equiv` (standardowo wyłączone), tegoż z dodatkowym potwierdzeniem tożsamości maszyny za pomocą RSA, klucza RSA użytkownika lub hasła użytkownika;
- po pomyślnej weryfikacji tożsamości klient żąda pewnych typowych usług (przydziału pseudoterminala, przekazywania połączeń X11 lub TCP/IP, przekazywania informacji autoryzacyjnej, wykonania polecenia na odległej maszynie);
- po rozpoczęciu sesji klient przesyła dane ze swojego wejścia serwerowi, wypisuje odpowiedzi; połączenie kończy się po przesłaniu przez serwer kodu zakończenia wykonywanego programu.

- protokół zaprojektowano z myślą o rozszerzalności:
  - na początku przesyłany jest numer wersji protokołu;
  - pierwszy pakiet wysłany przez każdą ze stron zawiera informacje o opcjach protokołu (co pozwala na uzgodnienie opcji zrozumiałych i akceptowalnych dla obu stron);
  - fazy autoryzacji i przygotowania sesji polegają na wysyłanie przez klienta żądań wobec serwera; serwer na każde odpowiada kodem świadczącym o sukcesie lub porażce; żądania niezrozumiałe dla serwera spowodują odesłanie kodu porażki, co pozwala na rozszerzanie obu stron o nowe możliwości bez naruszania możliwości współdziałania.

Format pakietów binarnych

- Po przesłaniu napisów identyfikacyjnych, obie strony przesyłają pakiety binarne o ustalonym formacie:
  - **długość** — 32 bity bez znaku, w postaci 4 oktetów, najpierw najstarszy bajt; nie uwzględnia pola długości ani oktetów dopełniających; maksymalna długość pakietu wynosi 262144 bajty;
  - **dopełnienie** — od 1 do 8 oktetów losowych danych (lub zer, gdy nie używa się szyfrowania); liczba oktetów wynosi $8 - (\text{długość} \mathtt{mod} 8)$; ma utrudnić ataki mające na celu deszyfrowanie pakietów;

- **typ pakietu** — 1 oktet (liczba całkowita bez znaku); wartość jest 255 zarezerwowana;
- **dane** — ciąg oktetów, o interpretacji zależnej od typu pakietu; liczba oktetów danych jest o 5 mniejsza niż wartość w polu długości;
- **suma kontrolna** — 32–bitowa suma CRC, w postaci czterech oktetów, najbardziej znaczący na początku; obejmuje pola dopełnienia, typu pakietu i danych; jest obliczana przez zaszyfrowaniem.

Szyfrowanie

- na początku sesji szyfrowanie jest wyłączone (oczywiste, gdyż trzeba wynegocjować rodzaj szyfrowania);
- szyfrowanie nie obejmuje pola długości pakietu;
- długość części zaszyfrowanej jest zawsze wielokrotnością ośmiu;
- szyfrowanie obejmuje ciąg pakietów traktowany jak jednolity strumień danych (nie obejmujący oczywiście kolejnych pól długości);
- szyfrowanie jest włączane po przesłaniu przez klienta klucza sesji; algorytm szyfrowania jest wybierany przez klienta.

Kompresja

- kompresja jest opcjonalna, może zostać wynegocjowana; używa algorytmu GZIP;
- kompresja dotyczy pola typu pakietu oraz pola danych;
- przy włączonej kompresji pole długości podaje długość skompresowanych danych plus 4 (dla pola CRC);
- ilość oktetów dopełniających jest obliczana na podstawie danych przed zaszyfrowaniem, zatem szyfruje się zawsze porcje o długości podzielnej przez 8;
- kompresja obejmuje ciąg pakietów traktowany jak jednolity strumień danych; między pakietami jest opróżniany jedynie bieżący blok kompresji, ale nie słownik kompresji;
- strumienie danych w obu kierunkach są kompresowane osobno.

Metody szyfrowania pakietów

- protokół przewiduje następujące metody szyfrowania:
  - brak szyfrowania;
  - IDEA w trybie CFB;
  - DES w trybie CBC;
  - potrójny DES w trybie CBC;
  - TSS, algorytm specyficzny dla SSH;
  - RC4;
- przy szyfrowaniu za pomocą algorytmu DES jako klucz używane jest początkowe 8 bajtów klucza sesji przekazanego przez klienta; najmniej znaczący bit każdego bajtu jest ignorowany, zatem klucz ma długość 56 bitów; wektor inicjalizacyjny składa się z samych zer;

- przy stosowaniu potrójnego DES-a stosuje się trzy osobne przebiegi z osobnymi wektorami inicjalizacyjnymi; dane są najpierw szyfrowane pierwszym szyfrem, potem deszyfrowane drugim, wreszcie szyfrowane trzecim; kluczem dla pierwszego szyfrowania jest pierwsze 8 bajtów klucza sesji, dla drugiego następne 8, dla trzeciego — trzecie 8; wszystkie wektory inicjalizacyjne skłądają się z zer;
- przy szyfrowaniu algorytmem IDEA jako klucz służy początkowe 16 bajtów klucza sesji; wektor inicjalizacyjny składa się z samych zer;

- przy szyfrowaniu algorytmem TSS wykorzystuje się cały 32-bajtowy klucz sesji; poziom bezpieczeństwa tego algorytmu nie jest znany; zaletą jest duża szybkość działania;
- przy szyfrowaniu algorytmem RC4 początkowe 16 bajtów klucza sesji służy do szyfrowania danych wysyłanych przez sewer klientowi, a pozostałe 16 bajtów do szyfrowania danych wysyłanych przez klienta; algorytm ten jest bardzo szybki.

Wymiana kluczy

- pierwszy komunikat serwera do klienta (po wymianie napisów z deklarujących wersję protokołu) zawiera klucz komputera serwera, klucz publiczny serwera, informacje o dostępnych metodach szyfrowania oraz znaczniki dostępnych rozszerzeń protokołu; ponadto jest wysyłana losowa wartość 64-bitowa (dla ochrony przed podsłuchem); komunikat ten nie jest szyfrowany;
- następnie obie strony wyliczają identyfikator sesji: