Istniejące systemy pracy na odległość

- TELNET,
- RSH,
- X11,
- systemy oparte na komercyjnych protokołach (np. Citrix).

Praca na odległość — potrzeby w zakresie bezpieczeństwa

- bezpieczna autentyfikacja osoby,
- bezpieczna autentyfikacja serwera,
- zabezpieczenie przed podszyciem się w trakcie trwania sesji,
- pufność danych przekazywanych podczas pracy.

Potrzeby te są zaspokajane przez protokół SSH.

Ogólne cechy SSH

- Protokół SSH działa w warstwie sesji modelu ISO/OSI.
- Opiera się na protokole transportu strumieniowego (TCP).
- Dodatkowo zapewnia kompresję.
- Zapewnia tunelowanie sesji terminala.
- Zapewnia tunelowanie protokołu X11.
- Zapewnia tunelowanie innych protokołów.
- Zapewnia negocjację algorytmów kryptograficznych.

Klucze maszyn

- Każdy serwer posiada klucz maszyny (co najmniej jeden).
- Dwa modele wiązania kluczy z maszynami
  – lokalna baza danych u klienta,
  – zarządca certyfikatów.
- Można skonfigurować, że przy pierwszym połączeniu klucz nie jest sprawdzany.

Konfiguracja

- Algorytmy szyfrowania, hasha kryptograficznego i kompresji są ustalane niezależnie dla każdego kierunku transmisji.
- Autentykacja serwera odbywa się za pomocą bezpiecznych algorytmów wymiany kluczy lub kryptografi klucza publicznego.
- Serwer wymaga od każdego użytkownika autentyfikacji, algorytm może zależeć od miejsca, z którego się on łączy.
- Muszą być zapewnione mechanizmy określania, które porty są przekazywane itp.

Składniki SSH

Protokół składa się z trzech zasadniczych komponentów

- Protokołu transportowego

  Zapewnia autentyfikacje serwera, poufność danych i ich integralność, opcjonalnie obsługuje kompresję.

- Protokołu autentyfikacji użytkownika

  Zapewnia autentyfikację klienta.

- Protokołu połączenia

  Multipleksuje różne połączenia.

Protokół transportowy — operacje

- ustanawianie połączenia,
- przesyłanie danych,
- wymiana kluczy,
- wymiana kluczy Diffiego-Hellmana,
- ponowna wymiana kluczy,
- żądanie usługi,
- inne (rozłączenie, zignorowane dane, debug, zarezerwowane).

Protokół transportowy — ustanawianie połączenia

- Serwer SSH nasłuchuje na porcie 22.
- Na początku następuje wymiana wersji protokołu SSH.
- Część tego tekstu jest używana w algorytmie Diffiego-Hellmana.
- Tutaj następuje potwierdzanie tożsamości serwera.
- Potem następuje dalsza negocjacja, w tym negocjacja kluczy.

Protokół transportowy — przesyłanie danych

Pakiet z danymi ma format

- długość pakietu (≤ 35000),
- długość uzupełnienia,
- dane (być może skompresowane, dł. ≤ 32768),
- uzupełnienie (minimalna jednostka transportu: 8 lub dł. bloku szyfrowania; minimalna długość danych 16 lub dł. bloku),
- MAC (message authentication code)

Protokół transportowy — kompresja

- MAC liczony ze skompresowanych danych;
- Kontekst kompresji jest inicjalizowany przy wymianie kluczy;
- Operacje uaktualnienia kontekstu — po każdym pakiecie.
- Metody kompresji: none, zlib;

Protokół transportowy — szyfrowanie

- Algorytm szyfrowania negocjowany przy wymianie kluczy.
- Szyfrowaniu podlegają: długość pakietu, długość uzupełnienia, dane, uzupełnienie.
- Kolejne pakiety tworzą jeden strumień szyfrowany (ważne dla CBC, CFB).
- Klucze są co najmniej 128-bitowe.
- Algorytmy szyfrowania: 3DES (CBC), BLOWFISH (CBC), TWOFISH256 (CBC), TWOFISH (CBC), TWOFISH192 (CBC), TWOFISH128 (CBC), AES256 (CBC), AES192 (CBC), AES128 (CBC), SERPENT256 (CBC), SERPENT192 (CBC), SERPENT128 (CBC), ARCFOUR, IDEA (CBC), CAST128 (CBC), NONE.

Protokół transportowy — integralność danych

- mac = MAC(klucz, numer_kolejny_pakietu . pakiet)
- MAC liczony jest z całego pakietu bez MAC.
- MAC liczony jest z danych niezaszyfrowanych.
- Numer kolejny jest zawsze liczony i nie jest kasowany przy renegocjacjach.
- Numer kolejny pierwszego pakietu to 0.
- MAC jest przesyłany bez szyfrowania.
- Stosowane algorytmy: hmac-sha1, hmac-sha1-96, hmac-md5, hmac-md5-96 (96 oznacza branie pierwszych 96 bitów wyniku).

HMAC

HMAC jest algorytmem budowy sumy kryptograficznej. Określamy tutaj

- K - klucz kryptograficzny używany do uwierzytelnienia;
- text - uwierzytelniany tekst;
- , - (przecinek) znak konkatenacji;
- B - długość bloku stosowanej funkcji haszującej (np. 64 bajty dla MD5);
- ipad - B razy powtórzony bajt 0x36;
- opad - B razy powtórzony bajt 0x5C;
- H(...) - algorytm liczenia hasza (czyli np. MD5);
- sumę kryptograficzną liczymy zgonie ze wzorem:

$$H(K \text{ XOR opad } .H(K \text{ XOR ipad . tekst})).$$

Protokół transportowy — negocjacje

- Na początku każda strona przesyła listę obsługiwanych algorytmów (wszystkie aspekty kryptograficzne i kompresja).
- Dodatkowo przesyłane jest losowe ciasteczko.
- Można przyspieszać negocjacje zgadując algorytmy.
- Potem wysyłany jest komunikat KEXINIT — właściwa negocjacja kluczy.
- Algorytm Diffiego-Hellmana.
- Wynikiem jest para: sekret (K) i hash (H).
- H jest używane jako identyfikator sesji.
- K i H służą do liczenia wszystkich kluczy oraz wektorów początkowych.
- Po negocjacji trzeba zatwierdzić wartości specjalnym komunikatem.

Protokół transportowy — algorytmy klucza publicznego

- Używane do certyfikacji serwera.
- Zastosowane algorytmy: DSS, RSA, X.509v3 z RSA, X.509v3 z DSS, SPKI z RSA, SPKI z DSS, PGP z RSA i PGP z DSS.

Protokół transportowy — żądanie usługi

- Obecnie zdefiniowane usługi
  - autentyfikacja użytkownika,
  - połączenie

Protokół autentyfikacji — umiejscowienie

- Stosowany po zażądaniu odpowiedniej usługi.
- Potrzebuje nazwy użytkownika, nazwy usługi oraz nazwy metody.
- Obsługiwane metody: none, publickey, password, hostbased.

Protokół autentyfikacji — omówienie metod

- none — pusta metoda autentyfikacji, każdy jest wpuszczany lub zatrzymywany.
- publickey — klient wysyła sygnaturę zaszyfrowaną swoim kluczem prywatnym, serwer ją odszyfrowuje odpowiednim kluczem publicznym i wpuszcza, jeśli odszyfrowanie się powiodło.
- password — klient przesyła do serwera zaszyfrowane hasło.
- hostbased — klient jest wpuszczany, gdy określony użytkownik loguje się z określonego komputera.

Protokół połączenia — do czego służy

- Ustanawianie kanałów komunikacyjnych: terminalowych, X11, tunelowych.
- Przesyłanie danych.
- Przekazywanie wartości zmiennych środowistkowych.
- Kontrola przepływu.
- Sygnały (jak w UNIX-ie).
- Kody wyjścia z programów.