Warsztaty z Sieci komputerowych
Lista 7

Zadanie dopuszczające do dalszych części (0 pkt.)

Uruchom dwie maszyny wirtualne *Virbian1* i *Virbian2*. Każda powinna mieć dwie karty sieciowe nazwane `enp-local` i `enp-remote`. Karty `enp-local` powinny być zmostkowane ze sobą, zaś za pomocą interfejsów `enp-remote` maszyny powinny być połączone (przez NAT) z Internetem.

- Aktywuj oba interfejsy obu maszyn. Interfejsom `enp-local` nadaj adresy IP odpowiednio `192.168.1.1/24` i `192.168.1.2/24`, zaś interfejsy `enp-remote` skonfiguruj za pomocą protokołu DHCP.
- Na obu maszynach uruchom Wiresharka nasłuchującego na wszystkich interfejsach.

Tutorial 1 (0 pkt.)

- Na maszynie *Virbian2* poleceniem

  ```
  V2$> telnet 192.168.1.1 7
  ```

  połącz się z serwerem echa maszyny *Virbian1*. Obejrzyj przesyłane pakiety w Wiresharku. Obejrzyj całą komunikację klikając prawym przyciskiem myszy jeden z pakietów należących do połączenia `telnet` i następnie wybierając z menu kontekstowego Wiresharka opcję *Follow* | *TCP stream*.
- Program `telnet` możesz zakończyć naciskając kombinację `Ctrl` + `]` i następnie wpisując `quit`. Warto zauważyć, że gdyby maszyny były podłączone do koncentratora a nie do przełącznika, to każda podłączona do tego koncentratora maszyna mogłaby odczytać tę komunikację.
- Na maszynie *Virbian1* włącz serwer SSH poleceniem

  ```
  V1#> systemctl start ssh
  ```

  a następnie połącz się z maszyny *Virbian2* z tym serwerem poleceniem

  ```
  V2$> ssh 192.168.1.1
  ```

  podając `user` jako hasło użytkownika `user`. Z jakim portem zostało nawiązane połączenie? Zauważ, że podczas pracy na zdalnej maszynie znak zachęty zawiera czerwony napis `[REMOTE]`.

- Będąc zalogowanym na maszynie *Virbian1* przez SSH wykonaj jakieś polecenie, np. wyświetl zawartość katalogu domowego poleceniem `ls`. Obejrzyj całą komunikację za pomocą opcji *Follow | TCP stream* Wiresharka. Czy potrafisz odczytać przesyłane dane?

- Skonfigurujemy teraz `ssh`, tak aby możliwe było łączenie się z maszyny *Virbian2* do maszyny *Virbian1* bez podawania hasła. W nowym terminalu poleceniem

  ```
  V2$> ssh-keygen
  ```

  wygeneruj klucz publiczny i prywatny. Zapisz je w domyślnych plikach (odpowiednio `.ssh/id_rsa.pub` oraz `.ssh/id_rsa`). Hasło zabezpieczające klucz pozostaw puste. (Zazwyczaj pozostawianie klucza prywatnego niezabezpieczonego hasłem to zły pomysł). Obejrzyj właśnie wygenerowane pliki z kluczami.

- Teraz wystarczy dopisać klucz publiczny do pliku `.ssh/authorized_keys` na serwerze SSH (maszynie *Virbian1*). W tym celu skopiuj ten klucz poleceniem

  ```
  V2$> scp .ssh/id_rsa.pub 192.168.1.1:keyfile
  ```

  Następnie używając SSH zaloguj się na maszynę *Virbian1*

  ```
  V2$> ssh 192.168.1.1
  ```

  Na maszynie *Virbian1* dopisz skopiowany właśnie klucz publiczny do pliku `.ssh/authorized_keys` poleceniami

  ```
  [REMOTE] V1$> mkdir .ssh
  [REMOTE] V1$> cat keyfile >> .ssh/authorized_keys
  [REMOTE] V1$> rm keyfile
  ```

  a następnie zamknij sesję SSH (wyloguj się z maszyny *Virbian1*).

- Sprawdź, czy działania odniosły skutek, tj. czy możesz zalogować się teraz z maszyny *Virbian2* na maszynę *Virbian1* bez podawania hasła. Polecenie

  ```
  V2$> ssh -v 192.168.1.1
  ```

  wyświetli kolejne etapy nawiązywania połączenia. Obejrzyj je również w Wiresharku. Na końcu zamknij sesję SSH.

Zadanie do zaprezentowania (2 pkt.)

Prostym sposobem zaszyfrowania połączenia jest wykorzystanie tunelowania strumienia danych w danych protokołu SSH.

- Na maszynie *Virbian2* utwórz tunel SSH łączący port `7777` lokalnej maszyny (*Virbian2*) z portem 7 maszyny *Virbian1*. W tym celu wykonaj polecenie

  ```
  V2$> ssh -N -L 7777:localhost:7 user@192.168.1.1
  ```

i pozostaw je uruchomione. Sprawdź, że po wpisaniu na maszynie *Virbian2* polecenia

```
V2$> telnet localhost 7777
```

odpowiada serwer echa maszyny *Virbian1*.

- Na podstawie Wiresharka odpowiedz na pytania (w każdym polu należy wpisać adres IP i port) dotyczące strumienia danych od maszyny *Virbian2* do maszyny *Virbian1*.
  - Na maszynie *Virbian2* strumień danych występuje w postaci niezaszyfrowanej jako pakiety przesyłane z ..... do .....
  - Pomiędzy maszyną *Virbian2* a maszyną *Virbian1* strumień danych występuje w postaci zaszyfrowanej jako pakiety przesyłane z ..... do .....
  - Na maszynie *Virbian1* strumień danych występuje w postaci niezaszyfrowanej jako pakiety przesyłane z ..... do .....

  Które z powyższych adresów IP są w wersji 4 a które w wersji 6?

- Zamknij sesję SSH tunelującą połączenie.

Tutorial 2 (0 pkt.)

- W tej części zapoznamy się z programem `gpg` będącym wolną implementacją standardu OpenPGP. Na maszynie *Virbian1* utwórz utwórz parę kluczy PGP, publiczny i prywatny, poleceniem

  ```
  V1$> gpg --gen-key
  ```

  Jako nazwę użytkownika wybierz `user1` a jako adres email wpisz `user1@mail.example.com`. Utwórz i zapamiętaj hasło chroniące klucz prywatny.

- Posiadane klucze (odpowiednio prywatne i publiczne) można wyświetlić poleceniami

  ```
  V1$> gpg --list-secret-keys
  V1$> gpg --list-keys
  ```

  Na razie będą tam widoczne tylko klucze użytkownika *user1*.

- Wejdź na stronę `https://www.veracrypt.fr/en/Downloads.html` i pobierz ten program (w dowolnej wersji) razem z odpowiadającym podpisem PGP (link *PGP Signature*). Zamiast programu Veracrypt możesz wybrać dowolny inny program podpisany kluczem PGP jego autora/autorów. Zapisz program w pliku `veracrypt.deb` a jego podpis w pliku `veracrypt.deb.sig`.

- Poleceniem

  ```
  V1$> gpg --verify veracrypt.deb.sig veracrypt.deb
  ```

  sprawdź, czy podpis jest poprawny. Otrzymasz komunikat o braku odpowiedniego klucza publicznego o identyfikatorze `5069A233D55A0EEB174A5FC3821ACD02680D16DE`.

- Pobierz ten klucz publiczny z ogólnodostępnego repozytorium kluczy poleceniem

  `V1$> gpg --recv-keys` *`identyfikator_klucza`*

  i wyświetl posiadane klucze publiczne poleceniem

  ```
  V1$> gpg --list-keys
  ```

  Zauważ, że przy Twoim kluczu publicznym jest napis `ultimate`, zaś przy kluczu publicznym opisanym jako *Veracrypt* jest napis `unknown`. Obie te wartości oznaczają poziom zaufania do tego, czy dany klucz należy do konkretnej osoby.

  Ponów próbę weryfikacji podpisu. Tym razem okaże się, że podpis jest poprawny, ale nie mamy żadnej gwarancji, że właśnie pobrany przez nas klucz publiczny faktycznie należy do autorów oprogramowania.

- Aby to naprawić, wejdź w tryb edycji tego klucza poleceniem

  ```
  V1$> gpg --edit-key Veracrypt
  ```

  Po znaku zachęty wpisz polecenie

  ```
  gpg> fpr
  ```

  wyświetlające skrót klucza publicznego. Teraz powinniśmy poprosić autorów oprogramowania o podanie nam zaufanym kanałem obliczonego po ich stronie skrótu klucza. Zamiast tego zadowolimy się porównaniem wyświetlanej funkcji skrótu z funkcją skrótu dostępną na ich stronie www.[1] Załóż, że posiadany klucz faktycznie należy do autorów oprogramowania i podpisz go poleceniem

  ```
  gpg> sign
  ```

  a następnie opuść tryb edycji poleceniem

  ```
  gpg> quit
  ```

  Zauważ, że jeśli teraz wyświetlisz dostępne klucze publiczne, to przy kluczu *Veracrypt* będzie informacja o pełnym (`full`) zaufaniu do tego klucza.

- Wykonaj kolejną próbę weryfikacji podpisu, tym razem powinna ona zakończyć się powodzeniem.

Zadanie do zaprezentowania (3 pkt.)

W tym zadaniu wygodnie jest myśleć, że maszyna *Virbian1* należy do użytkownika *user1*, zaś maszyna *Virbian2* do użytkownika *user2*.

[1] Od pewnego czasu skrót klucza publicznego jest zarazem jego identyfikatorem, więc wyświetlanym skrótem jest `5069 A233 D55A 0EEB 174A 5FC3 821A CD02 680D 16DE`.

- Zapisz klucz publiczny użytkownika *user1* z maszyny *Virbian1* w czytelnej postaci do pliku `user1-pgp-key` poleceniem

  ```
  V1$> gpg -a --export user1 > user1-pgp-key
  ```

- Na maszynie *Virbian2* wygeneruj klucz prywatny i publiczny, jako użytkownika podając `user2` a jako adres email `user2@mail.example.com`. Wyeksportuj klucz publiczny do pliku `user2-pgp-key`.

- Za pomocą SSH skopiuj plik `user1-pgp-key` na maszynę *Virbian2*, a następnie zaimportuj go do kluczy użytkownika *user2* za pomocą polecenia

  ```
  V2$> gpg --import < user1-pgp-key
  ```

  Wejdź w tryb edycji tego klucza, upewnij się, że jego funkcja skrótu jest odpowiednia i podpisz go kluczem prywatnym użytkownika *user2*.

- Wykonaj powyższy punkt, ale zamieniając role *user1* i *user2*: w efekcie klucz użytkownika *user2* powinien znaleźć się na maszynie *Virbian1*, zostać zaimportowany i podpisany kluczem użytkownika *user1*.

- Na maszynie *Virbian1* utwórz plik `message` umieść w nim jakąś treść. W celu podpisania wiadomości kluczem użytkownika *user1* i zaszyfrowania jej kluczem publicznym użytkownika *user2* wydaj polecenie

  ```
  V1$> gpg -a -r user2 -se message
  ```

  Szyfrogram zostanie zapisany do pliku `message.asc`, który należy skopiować za pomocą SSH na komputer *Virbian2*.

- Na maszynie *Virbian2* otrzymany plik `message.asc` należy odszyfrować kluczem prywatnym użytkownika *user2* i zweryfikować prawdziwość podpisu poleceniem

  ```
  V2$> gpg -d message.asc > deciphered_message
  ```

Lista i materiały znajdują się pod adresem http://www.ii.uni.wroc.pl/~mbi/dyd/.

Marcin Bieńkowski