Warsztaty nr 7 z Sieci komputerowych

Konfiguracja początkowa

Uruchom trzy maszyny *Virbian0*, *Virbian1* i *Virbian2*, każda z jedną kartą sieciową podłączoną do wirtualnej sieci `local0`. Maszyna *Virbian0* będzie serwerem pocztowym. Wygodnie jest myśleć, że maszyna *Virbian1* należy do użytkownika `user1`, zaś maszyna *Virbian2* do użytkownika `user2`.

Uruchom też dodatkową maszynę *Virbian3* z jedną kartą z domyślną konfiguracją sieciową (NAT).

Tutorial 1

Na początku przyjrzymy się bliżej protokołowi TLS. Wykorzystamy do tego maszynę *Virbian3*.

- Na maszynie *Virbian3* uzyskaj konfigurację sieciową poleceniem `dhcpcd` lub `dhclient`.
- Uruchom Wiresharka nasłuchującego na wszystkich interfejsach. Poleceniem

  ```
  V3$> dig example.com A
  ```

  sprawdź, jakie adresy IP odpowiadają nazwie `example.com`; oznaczymy je przez `adres_IP_1`, `adres_IP_2` itd. W polu filtra Wiresharka wpisz

  ```
  ip.addr == adres_IP_1 || ip.addr = adres_IP_2
  ```

- Pobierz stronę `https://example.com/` poleceniem[1]

  ```
  V3$> curl -s https://example.com/
  ```

  W Wiresharku zaobserwuj poszczególne pakiety. Zwróć uwagę na początkowe komunikaty nawiązujące połaczenie TCP: jakie flagi są ustawione w tych segmentach? Z jakim portem serwera łączy się klient?

- Następnie w Wiresharku zaobserwuj segmenty zawierające komunikaty protokołu TLS. Pierwszym z nich powinien być komunikat *Client Hello*, zaś drugim komunikat *Server Hello*.

  Kliknij komunikat *Client Hello* i rozwiń jego zawartość.

  - Jakie wersje protokołu TLS proponuje klient (pole *Extension: supported versions*)?
  - Ile metod szyfrowania proponuje klient (pole *Cipher Suites*)?
  - Zauważ, że pole *Extension: server name* zawiera niezaszyfrowaną nazwę domeny (możesz to sprawdzić dodatkowo klikając dowolny segment TCP i wybierając z menu kontekstowego *Follow* | *TCP Stream*).

  Kliknij komunikat *Server Hello* i rozwiń jego zawartość.

  - Jaką wersję protokołu TLS wybrał serwer ?
  - Jaki zestaw szyfrów z proponowanych przez klienta wybrał serwer?

  W przypadku wersji TLS 1.3, certyfikat serwera będzie przesyłany w kolejnych segmentach w zaszyfrowanej postaci. Żeby go wyświetlić zamiast Wiresharka wykorzystamy program `openssl`.

[1] Jeśli masz starszą wersję Virbiana i polecenie `curl` nie jest dostępne, możesz wykorzystać polecenie `wget` lub zainstalować pakiet `curl` za pomocą `apt`.

- W polu filtra Wiresharka wpisz `tls` i odśwież widok, żeby wyświetlić tylko komunikaty TLS. Połącz się z serwerem `example.com` na porcie `443` (HTTPS) poleceniem

  ```
  V3$> openssl s_client -connect example.com:443
  ```

  Obejrzyj nawiązywanie połączenia na podstawie informacji wyświetlonych w terminalu

  - Jaka wersja protokołu TLS została wynegocjowana (pole *Protocol*)?
  - Jaki zestaw szyfrów został wybrany (pole *Cipher*)?
  - Czy powyższe dwa pola są takie same, jak w poprzednim punkcie, gdy wykorzystywaliśmy polecenie `curl` i Wiresharka?
  - Ile podpisów liczy łańcuch certyfikatów (*Certificate chain*)?
  - Który z podpisów jest podpisem klucza publicznego `example.com`? A który został wykonany przed główny urząd certyfikacji (*root CA*)?

  Wyświetlane przez `openssl` informacje są na wyższym poziomie abstrakcji niż surowe komunikaty TLS widoczne w Wiresharku. Jeśli chcesz zobaczyć dokładnie, jakie komunikaty TLS zostały wysłane i odebrane, dodaj do polecenia `openssl` opcję `-msg`.

- Przerwij program `openssl`, jeśli jeszcze sam się nie rozłączył. Wykonaj ponownie powyższe polecenie `openssl`, ale tym razem po nawiązaniu połączenia wpisz zapytanie HTTP (zakończone pustym wierszem):

  ```
  GET / HTTP/1.1
  Host: example.com

  ```

  Sprawdź, czy otrzymujesz taką samą odpowiedź jak w przypadku polecenia `curl`.

- Wejdź na stronę `https://example.com/` w przeglądarce Firefox. Kliknij ikonę kłódki obok adresu i wybierz opcję *Connection secure* | *More information*. W zakładce *Security* kliknij przycisk *View Certificate* i sprawdź, czy informacje o łańcuchu certyfikatów są zgodne z tym, co wyświetlał program `openssl`. Obejrzyj wszystkie części łańcucha zwracając uwagę w szczególności na pola *Subject* i *Issuer*.

- Strona `https://badssl.com/` zawiera specjalne testowe adresy z celowo błędnymi certyfikatami TLS. Kliknij kilka znajdujących się na tej stronę linków i sprawdź, jaki błąd sygnalizuje przeglądarka w każdym przypadku. W szczególności odwiedź następujące adresy:

  - `https://expired.badssl.com/` (wygasły certyfikat),
  - `https://self-signed.badssl.com/` (certyfikat podpisany samodzielnie bez zaufanego CA),
  - `https://wrong.host.badssl.com/` (certyfikat wystawiony dla innej domeny).

  Każdorazowo wyświetlaj też szczegóły łańcucha certyfikatów i sprawdzaj, czy wyświetlane przez Firefoksa ostrzeżenie jest z nim zgodne.

Tutorial 2

W tej części przyjrzymy się działaniu protokołu pocztowego SMTP. Wykorzystamy do tego maszynę *Virbian0* jako serwer pocztowy, a maszyny *Virbian1* i *Virbian2* jako klientów pocztowych.

- Na maszynach *Virbian0*, *Virbian1* i *Virbian2* zmień nazwę interfejsu sieciowego na `en0` i aktywuj go. Interfejsowi `en0` na maszynie *Virbian0* przypisz adres `10.0.0.1/8`, zaś pozostałym dwóm interfejsom przypisz dowolne adresy z sieci `10.0.0.0/8`. Na wszystkich maszynach dodaj wpis

  ```
  10.0.0.1  mail.example.com
  ```

  do pliku `/etc/hosts`.

- Na maszynie *Virbian0* uruchom serwery SMTP i POP3 poleceniami

  ```
  V0#> systemctl start postfix
  V0#> systemctl start dovecot
  ```

- Na maszynie *Virbian1* skonfiguruj program *Thunderbird* do korzystania z adresu `user1@mail.example.com`.[2] W tym celu w Thunderbirdzie wykorzystaj kreator tworzenia konta pocztowego: zostaje on uruchomiony przy pierwszym starcie programu, ale można go również uruchomić ponownie wybierając z menu *New Account* | *Email*. Wpisz w nim swoje imię i nazwisko w polu *Your full name*, w polu *Email address* wpisz `user1@mail.example.com`, zaś w polu *Password* wpisz `pass1`. Następnie kliknij link *Configure manually* i uzupełnij pozostałe pola w następujący sposób.

  - W części *Incoming Server* powinien być wybrany protokół POP3, serwer `mail.example.com`, port 110, *wyłączone* szyfrowanie, w polu *Authentication method* wybrana opcja *Normal password*, a jako użytkownik `user1@mail.example.com`.[3]
  - W części *Outgoing Server* powinien być wybrany serwer `mail.example.com`, port 25, *wyłączone* szyfrowanie, a w polu *Authentication method* wybrana opcja *No authentication*.

  Po kliknięciu przycisku *Done* należy przeczytać i następnie zignorować ostrzeżenie o używaniu nieszyfrowanych protokołów.

- Wykonaj powyższy punkt z odpowiednimi zmianami, tak żeby skonfigurować program *Thunderbird* na maszynie *Virbian2* do korzystania z adresu `user2@mail.example.com` (z hasłem `pass2`).

- W Thunderbirdzie na maszynie *Virbian1* kliknij przycisk *New Message*, napisz i wyślij mail do `user2@mail.example.com`). Następnie wyślij kolejny mail, ale tym razem dołącz do niego dowolny plik.

- Na maszynie *Virbian0* i znajdź wysłane maile w katalogu `/var/spool/mail/vhosts/mail.example.com/user2`.[4] Obejrzyj i porównaj ich budowę.

  - Które pole odpowiada za datę wysłania maila?
  - Gdzie jest data jego odebrania przez serwer SMTP?
  - Jaka jest wartość pola `Content-Type` w nagłówku obu maili?
  - Jaka jest wartość pól `Content-Type` w poszczególnych częściach maila z załącznikami?
  - Jaka jest wartość pól `Content-Transfer-Encoding` w obu mailach?

[2] Jeśli korzystasz ze starej wersji Virbiana, zamiast kont `useri@mail.example.com` użyj kont `studenti@mail.example.com`. Hasła do nich to `studenti`.
[3] W przypadku starej wersji Virbiana ostatnim polu należy wpisać `student1`, tj. bez `@mail.example.com`.
[4] W przypadku starej wersji Virbiana będzie to katalog `/var/spool/vmail/student2`.

- Odbierz ten mail w Thunderbirdzie uruchomionym na maszynie *Virbian2* (klikając na ikonie chmury obok przycisku *New Message*), odpowiedz na niego i sprawdź, czy odpowiedź dotarła do Thunderbirda na maszynie *Virbian1*.
- Włącz Wiresharka na maszynie *Virbian1*. Ponownie wyślij maila do `user2@mail.example.com` i obejrzyj przesłane pakiety w Wiresharku. Znajdź jeden z przesyłanych segmentów TCP i wybierając z kontekstowego menu opcję *Follow* | *TCP Stream* sprawdź, jakie komunikaty zostały wymienione między maszyną *Virbian1* a serwerem SMTP uruchomionym na maszynie *Virbian0*. Która część komunikacji zawiera wysyłany mail? Nie zamykaj okna z komunikacją; przyda się nam w kolejnych punktach tego tutoriala.
- Poleceniem

  ```
  V1$> telnet mail.example.com 25
  ```

  połącz się z portem SMTP i wykorzystaj powyższą komunikację do wysłania maila do adresu `user2@mail.example.com`. Zauważ, że treść maila (po poleceniu `DATA`) musi być zakończona pojedynczą kropką. Możesz pominąć pola nagłówka lub wpisać tylko niektóre. Na maszynie *Virbian2* sprawdź w Thunderbirdzie, czy mail dotarł.
- Włącz teraz szyfrowanie TLS protokołu SMTP w Thunderbirdzie na maszynie *Virbian1*. W tym celu w lewym panelu okna programu kliknij prawym przyciskiem myszy nazwę konta `user1@mail.example.com` i z menu kontekstowego wybierz opcję *Settings*. W oknie konfiguracji z menu po lewej stronie wybierz opcję *Outgoing Server (SMTP)*, kliknij przycisk *Edit*, a następnie w części *Connection security* wybierz opcję *STARTTLS* i zatwierdź zmiany przyciskiem *OK*. Zrestartuj Thunderbirda; w przeciwnym przypadku prawdopodobnie zmiany nie zostaną zastosowane.[5]
- Wyślij ponownie mail testowy do `user2@mail.example.com` i zaobserwuj przesyłane za pomocą protokołu SMTP dane w Wiresharku (*Follow* | *TCP Stream*). Poza samym początkiem komunikacji późniejsze dane powinny być zaszyfrowane i nie powinno się ich dać odczytać. Obejrzyj przesyłane przez protokół TLS komunikaty i porównaj je z komunikatami TLS, które zaobserwowaliśmy w poprzednim tutorialu.
- Wyślemy teraz mail wykorzystując szyfrowane połączenie. Wykonaj polecenie:

  ```
  V1$> openssl s_client -quiet -crlf -connect mail.example.com:25 -starttls smtp
  ```

  i wyślij maila posługując się poleceniami protokołu SMTP (`MAIL FROM`, `RCPT TO` i `DATA`). Obejrzyj przesyłane dane w Wiresharku i upewnij się w Thunderbirdzie na maszynie *Virbian2*, że mail został dostarczony.

Tutorial 3

W tej części zapoznamy się z programem `gpg` będącym implementacją standardu OpenPGP. Wykorzystamy do tego maszynę *Virbian3* z dostępem do Internetu.

- Utwórz utwórz parę kluczy PGP (publiczny i prywatny) poleceniem

  ```
  V3$> gpg --gen-key
  ```

  Jako nazwę użytkownika wybierz `user3`, a jako adres mail wpisz `user3@mail.example.com`. Utwórz i zapamiętaj hasło chroniące klucz prywatny.

[5] Jeśli to też nie pomoże, można skasować konto i założyć je ponownie, wybierając w ustawieniach serwera SMTP od razu połączenie szyfrowane.

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

```
V3$> gpg --list-secret-keys
V3$> gpg --list-keys
```

  Na razie będą tam widoczne tylko klucze użytkownika `user3`.

- Wejdź na stronę `https://veracrypt.io/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. Zapisz program w pliku `veracrypt.deb` a jego podpis w pliku `veracrypt.deb.sig`.

- Poleceniem

```
V3$> 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

```
V3$> gpg --recv-keys identyfikator_klucza
```

  i wyświetl posiadane klucze publiczne poleceniem

```
V3$> 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/instytucji.

  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

```
V3$> gpg --edit-key Veracrypt
```

  Po znaku zachęty wpisz polecenie

```
gpg> fpr
```

  wyświetlające skrót klucza publicznego.[6] Teraz powinniśmy poprosić autorów oprogramowania o podanie nam zaufanym kanałem obliczonego po ich stronie skrótu klucza. Zamiast tego sprawdź, czy wyświetlana wartość jest zgodna z informacją na stronie WWW oprogramowania. Podpisz klucz 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.

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

Wyzwanie

- Wygeneruj parę kluczy PGP dla użytkownika `user1` na maszynie *Virbian1*, jako adres email podając `user1@mail.example.com`.
- Zapisz klucz publiczny użytkownika `user1` z maszyny *Virbian1* w czytelnej postaci do pliku `user1-pgp-key.asc` poleceniem

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

  Wyślij ten klucz mailem do użytkownika `user2@mail.example.com`.
- Analogicznie na maszynie *Virbian2* wygeneruj parę kluczy PGP, jako użytkownika podając `user2`, a jako adres mail `user2@mail.example.com`. Wyeksportuj klucz publiczny do pliku `user2-pgp-key.asc` i wyślij użytkownikowi `user1`.
- Na maszynie *Virbian1* zaimportuj klucz publiczny użytkownika `user2` za pomocą polecenia

  ```
  V1$> gpg --import < user2-pgp-key.asc
  ```

  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 na maszynie *Virbian2* zamieniając role `user1` i `user2`.
- Na maszynie *Virbian1* utwórz plik `message` i umieść w nim jakąś treść. W celu podpisania wiadomości prywatnym kluczem użytkownika `user1` i zaszyfrowania jej kluczem publicznym użytkownika `user2` wykonaj polecenie

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

  Szyfrogram zostanie zapisany do pliku `message.asc`, który można wysłać mailem do użytkownika `user2`.
- Na maszynie *Virbian2* otrzymany plik `message.asc` należy odszyfrować kluczem prywatnym użytkownika `user2` i zweryfikować autentyczność podpisu poleceniem

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

Niepunktowane zadanie dodatkowe

Każdorazowe wpisywanie poleceń `gpg` jest mało wygodne; w tym zadaniu skonfigurujemy obsługę kluczy PGP w Thunderbirdzie.

- Niestety Thunderbird wykorzystuje inny format kluczy i przechowuje je w innym miejscu niż `gpg`. Wykorzystamy tutaj już istniejące klucze (można też wygenerować nowe bezpośrednio w Thunderbirdzie).

  Wyeksportuj klucz prywatny użytkownika `user1` poleceniem

  ```
  V1$> gpg --export-secret-key user1 > user1-pgp-private-key.gpg
  ```
- W Thunderbirdzie (na maszynie *Virbian1*) wejdź w tryb edycji ustawień konta `user1@mail.example.com` klikając prawym przyciskiem myszy nazwę konta (w lewym panelu okna programu) i wybierając z menu kontekstowego opcję *Settings*.

W oknie konfiguracji z menu po lewej stronie wybierz opcję *End-To-End Encryption*, a następnie w sekcji *OpenPGP* kliknij przycisk *Add Key...*. Wybierz *Import an existing OpenPGP Key* a następnie plik `user1-pgp-private-key.gpg`. (Klucza publicznego nie trzeba importować osobno).

W tej samej zakładce *End-To-End Encryption*, w sekcji *OpenPGP* wybierz nowo dodany klucz.

- Wykonaj poprzednie dwa punkty, ale dla użytkownika `user2` na maszynie *Virbian2*.
- W Thunderbirdzie na maszynie *Virbian1* wyślij mailem klucz publiczny użytkownika `user1` do użytkownika `user2`. W tym celu wybierz z menu opcję *Tools* | *OpenPGP Key Manager*, kliknij prawym przyciskiem myszy klucz użytkownika `user1` i wybierz opcję *Send Public Key(s) by Email*.
- W Thunderbirdzie na maszynie *Virbian2* odbierz ten mail, kliknij prawym przyciskiem myszy załącznik i wybierz opcję *Import OpenPGP Key*. W kolejnym oknie zaznacz opcję *Accepted (unverified)* i kliknij przycisk *Import*.

  Następnie w Thunderbirdzie na maszynie *Virbian2* wybierz z menu opcję *Tools* | *OpenPGP Key Manager*, kliknij podwójnie w klucz użytkownika `user1`. Jeśli wyświetlany fingerprint jest poprawny, zaznacz opcję *Yes, I've verified in person this key has the correct fingerprint* i kliknij przycisk *OK*.
- Wyślij zaszyfrowany i podpisany mail od użytkownika `user2` do użytkownika `user1`. W tym celu wystarczy napisać mail w Thunderbirdzie i w menu na górze zaznaczyć opcję *Encrypt*.
- Na maszynie *Virbian0* znajdź ten mail w katalogu `/var/spool/mail/vhosts/mail.example.com/user1`. Obejrzyj jego budowę w edytorze tekstowym. Jakie części występują w tym mailu?
- Na maszynie *Virbian1* odbierz powyższy mail. Zostanie on automatycznie odszyfrowany a zawarty w nim podpis zweryfikowany. Klikając przycisk *OpenPGP* po prawej stronie można wyświetlić okno *Message Security - OpenPGP* z informacją, że podpisu nie można zweryfikować (bo klucz publiczny użytkownika `user2` nie jest znany użytkownikowi `user1`). Okazuje się, że ten klucz został również dołączony do maila i można go od razu zaimportować klikając przycisk *Import*.
- Taki klucz należy jednak zweryfikować, tak jak robiliśmy to w symetrycznej sytuacji: klikając przycisk *View signer key* otworzymy okno z informacją o kluczu publicznym użytkownika `user2`.
- Ponownie otwórz maila od `user2`: tym razem w oknie *Message Security - OpenPGP* powinna znaleźć się informacja, że podpis został poprawnie zweryfikowany.
- Od tego momentu można też wysyłać zaszyfrowane i podpisane maile w drugim kierunku. Sprawdź to wysyłając mail od użytkownika `user1` do `user2`.

Materiały do kursu znajdują się w systemie SKOS: `https://skos.ii.uni.wroc.pl/`.

Marcin Bieńkowski