Bezpieczeństwo Sieci Komputerowych - laboratorium

Ćwiczenie 3: Bezpieczne usługi sieciowe, wirtualne sieci prywatne.

Cel ćwiczenia: Opanowanie umiejętności instalacji i konfiguracji bezpiecznego serwera usług www. Opanowanie umiejętności instalacji i konfiguracji bezpiecznego serwera usług terminalowych oraz budowy prostych tuneli VPN typu komputer-komputer.

Wprowadzenie

Ćwiczenia związane jest z bezpieczeństwem usług sieciowych, głównie WWW. Transmisja HTTP nie jest w żaden sposób szyfrowana, co umożliwia łatwe odczytywanie przechwyconych danych. Bezpiecznym odpowiednikiem tego protokołu jest HTTPS, który zapewnia przesłanie danych HTTP bezpiecznym tunelem SSL (w uproszczeniu HTTPS = SSL + HTTP).

Zdalna praca (na routerach, serwerach) możliwa jest za pomocą protokołów usług terminalowych. Najpopularniejszy z nich Telnet nie zapewnia żadnych mechanizmów bezpieczeństwa (komendy, ich rezultaty, hasła, itp. są przesyłane w sieci jawnym tekstem). Bezpieczeństwo usług terminalowych zapewnia natomiast protokół SSH, dzięki któremu zbudować można szyfrowane połączenie ze zdalnym urządzeniem.

Zarówno Telnet jak i SSH pracują w trybie klient-serwer. Na serwerze musi być uruchomiony proces (demon) serwera SSH, klient musi dysponować oprogramowaniem klienckim (klient telnet jest na ogół dostępny w systemach operacyjnych, ‘uniwersalnym’ klientem zarówno Telnet jak i SSH jest program putty). Tworzony jest szyfrowany tunel pomiędzy serwerem a klientem. SSH zapewnia również wiele innych, na ogół nie znanych użytkownikom, funkcjonalności, np. przekierowywanie portów, usługi pośredniczące.

Typowe protokoły komunikacyjne (np. typu peer-to-peer) zwykle nie oferują żadnych mechanizmów ochrony, przez co komunikacja pomiędzy urządzeniami nie jest bezpieczna (w szczególności poufna). Poprawić ich bezpieczeństwo można korzystając właśnie z tuneli. Budowę i obsługę tuneli umożliwiają (oprócz SSH) protokoły takie jak IPSec czy SSL oraz liczne aplikacje implementujące te rozwiązania.

Wymagane informacje

- Znajomość zasad działania algorytmów kryptograficznych (hybrydowego), podpisów cyfrowych oraz obsługi certyfikatów.
- Umiejętność pracy w Linux (instalacja usług, edycja plików, uruchamianie programów)
- Znajomość typowych numerów portów dla usług (szczególnie HTTP i SSL).

Ćwiczenia do wykonania

- Do realizacji ćwiczeń należy wykorzystać system Linux Ubuntu oraz drugi (dowolny) system operacyjny. Ubuntu powinien otrzymać adres IP automatycznie, z puli laboratorium. Wymagany będzie dostęp do Internetu.
- Wykonując zadania proszę wszędzie gdzie to możliwe umieszczać **dane członków grupy** (nazwiska i numery indeksów) – w danych serwera i certyfikatach, nazwach tworzonych plików, katalogów, kont użytkowników, treści plików, komunikatach, itp. Ich obecność na zrzutach ekranu w sprawozdaniu jest potwierdzeniem wykonania ćwiczeń.

[ZE] oznacza konieczność umieszczenia odpowiedniego zrzutu ekranu w sprawozdaniu.

1. Na Ubuntu zainstalować serwer http (pakiet apache2), zmodyfikować stronę www (utworzyć własną prostą stronę w html, zawierającą nazwiska członków grupy), zweryfikować poprawność działania [ZE] i ocenić poufność transmisji za pomocą Wireshark [ZE]. Pliki konfiguracyjne serwera http znajdują się na ogół w katalogu */etc/apache2*.
2. Zainstalować i skonfigurować protokół SSL do bezpiecznych połączeń http:

```
apt-get install openssl
```

- Wygenerować i zapisać do plików (**uwaga: poniższe nazwy plików i ścieżek są <u>przykładowe</u>**):

  <u>klucz rsa</u> (podać plik wynikowy i długość klucza):

```
openssl genrsa -out /etc/apache2/klucz.key 2048
```

  <u>certyfikat ssl</u> (określić czas ważności certyfikatu opcją –days; format certyfikatu x509 oznacza certyfikat 'samopodpisany' – self-signed):

```
openssl req -new -x509 -days 365 –key /etc/apache2/klucz.key
                                  -out /etc/apache2/certyf.crt
```

- Upewnić się, że usługa będzie nasłuchiwała na porcie 443 (plik `/etc/apache2/ports.conf`).
- Uruchomić moduł ssl

```
a2enmod ssl
```

- Utworzyć domyślną stronę https (np. `/etc/apache2/sites-available/stronassl`)

  Plik można utworzyć modyfikując plik default lub default-ssl. Plik musi zawierać następujące wpisy:

```
<virtualhost *:443>                #numer portu ssl
SSLEngine On                       #włączenie silnika ssl
SSLCertificateFile /etc/apache2/certyf.crt #plik z certyfikatem
SSLCertificateKeyFile /etc/apache2/klucz.key #plik z kluczem
```

- Uruchomić stronę:

```
a2ensite stronassl
```

  Zgodnie z podpowiedzią aktywować nową konfigurację:

```
service apache2 reload
```

Z drugiego komputera połączyć się ze stroną przez ssl. Po zweryfikowaniu poprawności działania strony (tj. strona https 'się otwiera' w przeglądarce, połączenie jest szyfrowane) [ZE] **zgłosić wykonanie ćwiczenia Prowadzącemu.**

3. Wyświetlić (w przeglądarce) i przeanalizować szczegóły certyfikatu ssl. W jaki sposób przeglądarka wykorzystuje i obsługuje certyfikaty? Jakie algorytmy kryptograficzne wykorzystane zostały do zestawienia tej sesji ssl [ZE]? Czy wystawca certyfikatu Państwa witryny został dodany do listy zaufanych wystawców w przeglądarce? Dlaczego przeglądarka sygnalizuje że „Połączenie nie jest bezpieczne" (proszę szczegółowo wyjaśnić na czym polega problem)? Za pomocą Wireshark ocenić bezpieczeństwo połączenia ssl (czy przesyłane treści są szyfrowane, czy certyfikat jest szyfrowany, itp.).
4. Przed nawiązaniem połączenia ssl serwer oraz przeglądarka negocjują parametry kryptograficzne sesji (m.in. algorytmy szyfrowania i podpisywania). Zarówno serwer jak i przeglądarka dysponują listą dopuszczalnych zestawów kryptograficznych. Informacje o wynegocjowanych parametrach sesji można uzyskać wybierając opcję 'więcej informacji' – przy ikonce bezpiecznej sesji w przeglądarce.
    a) wyświetlić listę zestawów kryptograficznych w przeglądarce (strona konfiguracji: *about:config*, proszę użyć filtra *ssl*) [ZE].
    b) ustalić listę na serwerze ssl - użyć komendy *openssl ciphers*, konfiguracji zapisanej w pliku *ssl.conf* (katalog */etc/apache2/mods-available*, opcja *SSLCipherSuite*) oraz instrukcji dostępnej na stronie: https://wiki.openssl.org/index.php/Manual:Ciphers(1) . Opisać w sprawozdaniu.
    c) Skonfigurować po stronie serwera ssl możliwość użycia tylko jednej opcji zestawu algorytmów (z akceptowanych przez przeglądarkę i inną niż wynegocjowana w dotychczasowych sesjach – np. AES256 zamiast AES128) [ZE]. Nawiązać połączenie i wyświetlić parametry kryptograficzne sesji w przeglądarce [ZE].
5. Utworzyć konta 2 użytkowników (nazwy kont mają pochodzić od nazwisk członków grupy) na serwerze Ubuntu (komputer 1).
6. Instalacja i uruchomienie demona telnet w Ubuntu
    - Zainstalować i uruchomić serwis (demona) telnet:
      `$ sudo apt-get install telnetd`
      (Zrestartowć usługę inetd):
      `$ sudo /etc/init.d/openbsd-inetd restart`
    - Połączyć się z serwerem używając standardowego numeru portu (z komputera 2).
    - Zmienić numer portu dla telnet (zmodyfikować odpowiednią linię pliku `/etc/services`)
    - Uruchomić Wireshark, połączyć się z serwerem za pomocą putty (z komputera 2), ocenić poufność przesyłanych danych (login, hasło, komendy)

7. Instalacja i uruchomienie SSH
   - Zainstalować ssh na serwerze Ubuntu (pakiet *openssh-server*). Zmienić domyślny port na port spoza puli *well-known-ports*, zmodyfikować 2 inne wybrane parametry ssh (np. komunikat powitalny).

     Konfiguracja ssh: `/etc/ssh/sshd_config`

     Zrestartować usługę komendą: `systemctl restart sshd.service`
   - Z drugiego komputera połączyć się z serwerem za pomocą putty. W jaki sposób można potwierdzić autentyczność serwera ?
   - Uruchomić Wireshark, ponownie połączyć się z serwerem za pomocą putty, ocenić poufność przesyłanych danych (login, hasło, komendy) – porównać z poufnością protokołu telnet.

   Wykonanie zadania zgłosić prowadzącemu.
8. Tunelowanie SSH

   Komputer 2 podłączyć do sieci 'publicznej' (przełącznik wskazany przez Prowadzącego). Adres w sieci uzyskać z DHCP. Z sieci 'publicznej' jest możliwy dostęp do sieci laboratoryjnej oraz Internetu, jednak blokowany jest port 80 (http). Poza tym, sieć jest monitorowana, w związku z czym transmisja w niej nie jest poufna. Zestawić tunel z komputera 2 do serwera SSH – tunel ma umożliwiać ominięcie blokady portu 80, jak i zapewniać poufność transmisji w niebezpiecznej sieci. Zbadać poufność danych przesyłanych tunelem, potwierdzić fakt ominięcia zapory.

   Idea tunelowania jest następująca. Transmisja http zostanie przesłana szyfrowanym tunelem do serwera SSH, który będzie pośredniczył pomiędzy komputerem 2 a serwerami WWW w Internecie. W celu budowy tunelu należy:
   - Przekierować ruch WWW (protokół SOCKS) na komputerze 2 na 'wysoki' port (np. 6000) tego samego komputera – za pomocą ustawień Proxy przeglądarki.
   - Skonfigurować putty na komputerze 2: połączenie ssh z serwerem oraz w zakładce SSH->Tunnels jako *forwarded port* dodać 6000 (w sekcji *Destination* wybrać *Dynamic*, *Auto*).

   Działający tunel zademonstrować Prowadzącemu.

Przed opuszczeniem sali wyłączyć komputer.

Sprawozdanie - składa się z dwóch części:

1. Sprawozdanie z ćwiczenia - wyłącznie w formacie .pdf. Proszę udokumentować wykonane zadania za pomocą zrzutów ekranu opatrzonych własnymi komentarzami. W przypadku zadań wymagających samodzielnego rozwiązania zamieścić dokładny opis ich realizacji (np. wybrane opcje i wartości). Odpowiedzieć na pytania z instrukcji, zamieścić wymagane analizy (logów Wiresharka oraz utworzonego certyfikatu ssl, zestawów kryptograficznych).

2. Podpisy pod sprawozdaniem. Plik ze sprawozdaniem ma zostać podpisany (oczywiście cyfrowo) przez wszystkich członków grupy. Podpisy mają być osobnymi plikami i muszą być weryfikowalne za pomocą kluczy publicznych z ćwiczenia 2. Plik z podpisem ma mieć nazwę złożoną z czterech pierwszych liter

nazwiska (tylko małe litery, bez polskich znaków), np.: *kowa.sig* (student Kowalski), *zak.asc* (student Żak), nazwa pliku jest bardzo ważna – poprawność podpisów weryfikuje skrypt. Brak lub niepoprawny podpis oznacza obniżenie oceny z ćwiczenia o 20%.