Bezpieczeństwo systemów informatycznych

ĆWICZENIE SSO

1. Mechanizm jednokrotnego uwierzytelniania (*single sign-on*)

1.1 Usługi zdalnego dostępu

1.1.1 rlogin – remote login

Komenda `rlogin` pozwala na zdalny dostęp do konta systemu operacyjnego. Podejmuje próbę zalogowania bieżącego użytkownika lokalnego systemu operacyjnego na wskazanym systemie zdalnym. Jeśli nie wyspecyfikowano inaczej, wybrane zostaje zdalne konto użytkownika o takiej samej nazwie jak bieżąca nazwa użytkownika w systemie lokalnym. Po zalogowaniu, uruchamiana jest w trybie interaktywnym domyślna powłoka zdefiniowana dla konta zdalnego, np. sh, csh, tcsh itp.

```
localhost> rlogin remotehost
```

`remotehost` – nazwa (domenowa) lub adres (IP) komputera zdalnego

Dalsza praca z systemem zdalnym odbywa się podobnie jak w przypadku usługi sieciowej TELNET. Jeśli wymagane jest podanie hasła dla konta zdalnego, `rlogin` poprosi użytkownika o podanie go przed zalogowaniem:

```
localhost> rlogin remotehost
Password:
remotehost>
```

Polecenie `rlogin` umożliwia również dostęp do konta użytkownika o innej nazwie niż bieżący użytkownik. Wówczas nazwę użytkownika zdalnego konta należy podać po opcji `-l` (log as):

```
localhost> rlogin -l username remotehost
username's Password:
remotehost>
```

Hasło jest transmitowane tekstem jawnym, podobnie jak w protokole TELNET, co stanowi poważne zagrożenie poufności zdalnego konta. Jednak istnieje możliwość wykorzystania procedury jednokrotnego uwierzytelniania (ang. *single sign-on*) zalogowania użytkownika bez potrzeby podawania hasła dostępu do konta. Umożliwia to *mechanizm zaufania* do systemów, z których nawiązywane są sesje `rlogin`. W systemowym pliku /etc/hosts.equiv umieszczona jest lista nazw komputerów, z których dozwolony jest zdalny dostęp na lokalne konto użytkownika bez pytania o hasło. Lista ta dotyczy wszystkich kont lokalnych. Oto zawartość przykładowego pliku /etc/hosts.equiv na komputerze uran:

```
saturn
mars
neptun
```

Tak zdefiniowana lista zaufanych systemów pozwala każdemu użytkownikowi posiadającemu konto na dowolnym z wymienionych na niej systemów, a więc na saturnie, marsie lub neptunie (w tym w szczególności na wszystkich trzech), zalogować się na własne konto w systemie uran, bez wymogu podania hasła. Istotne jest, żeby konto, na które użytkownik uzyskuje dostęp nazywało się identycznie jak zdalne konto, z którego ten użytkownik nawiązuje połączenie.

Każdy z użytkowników może również zdefiniować w pliku ~/.rhosts własną dodatkową listę obejmującą nazwy komputerów (i ewentualnie nazwy użytkowników na tych komputerach), które on obdarza zaufaniem i umożliwia im zdalny dostęp na swoje konto bez podawania hasła. Jest to użyteczne, gdy ten sam użytkownik posiada różne konta (być może o różnych nazwach użytkownika) na kilku systemach i :

1. nie chce wystawiać swojego hasła na transmisje tekstem jawnym; lub
2. chce umożliwić zdalne wywoływania poleceń na swoim koncie.

Oto zawartość przykładowego pliku ~/.rhosts na koncie michal w systemie uran:

```
jowisz
unixlab szychowiak
```

Tak lista pozwala zalogować się bez wymogu podania hasła użytkownikowi systemu jowisz posiadającemu konto o identycznej nazwie (czyli michal) oraz użytkownikowi szychowiak z systemu unixlab.

Mechanizm zaufania jest jednak bardzo niebezpieczny, gdyż wystawia konto ufające na najprostsze nawet ataki poprzez podszycie się (DNS spoofing, IP spoofing).

1.1.2 rsh – remote shell

Komenda `rsh` pozwala na wywołanie polecenia systemowego na zdalnym koncie systemu operacyjnego. Podejmuje próbę zalogowania bieżącego użytkownika lokalnego systemu operacyjnego na wskazanym systemie zdalnym i wykonania podanego polecenia. Jeśli nie wyspecyfikowano inaczej, wybrane zostaje zdalne konto użytkownika o takiej samej nazwie jak bieżąca nazwa użytkownika w systemie lokalnym. Dla wykonania polecenia uruchamiana jest powłoka domyślna w trybie nieinteraktywnym.

```
local> rsh remotehost ls
```

`remotehost` – nazwa (domenowa) lub adres (IP) komputera zdalnego
`ls` – przykładowe polecenie do wykonania na zdalnym koncie.

Jeśli nie wyspecyfikowano polecenia, uruchomiona zostanie powłoka i wówczas polecenie to działa identycznie jak `rlogin` (powłoka w trybie interaktywnym).

1.1.3 Bezpieczeństwo

Polecenia z rodziny r*, jakkolwiek użyteczne w wielu zastosowaniach, wprowadzają istotne zagrożenie wystawiając konto zdalne na ataki poprzez podszycie się oraz podsłuchanie hasła. Korzystanie z nich wymaga więc na ogół zastosowania dodatkowych mechanizmów kontroli dostępu do systemu.

Zadania:

1. Zdefiniuj listę zaufania pozwalającą tobie zalogować się bez uwierzytelniania na twoim własnym koncie w systemie `unixlab` z dowolnego komputera w laboratorium.

2. Mechanizmy kontroli zdalnego dostępu do usług systemu

2.1 inetd – demon usług internetowych

Podstawowe usługi sieciowe są udostępniane za pośrednictwem demona inetd, którego zadaniem jest odbieranie nadchodzących z sieci zgłoszeń klientów usług i kierowanie ich do właściwych procesów obsługujących te zgłoszenia (serwerów usług). Część najprostszych usług jest obsługiwana przez samego inetd, a dla obsługi pozostałych inetd uruchamia procesy przypisane odpowiednim usługom zgodnie z zawartością pliku /etc/inetd.conf. Przykładowo:

```
#<service> tli <proto> <flags> <user>  <server_pathname>    <args>

echo     stream  tcp   nowait  root    internal
echo     dgram   udp   wait    root    internal
daytime  stream  tcp   nowait  root    internal
daytime  dgram   udp   wait    root    internal

ftp      stream  tcp   nowait  root    /usr/sbin/in.ftpd    in.ftpd
telnet   stream  tcp   nowait  root    /usr/sbin/in.telnetd in.telnetd
shell    stream  tcp   nowait  root    /usr/sbin/in.rshd    in.rshd
login    stream  tcp   nowait  root    /usr/sbin/in.rlogind in.rlogind
talk     dgram   udp   wait    root    /usr/sbin/in.talkd   in.talkd
finger   stream  tcp   nowait  nobody  /usr/sbin/in.fingerd in.fingerd
```

Konfiguracja ta udostępnia m.in. usługę daytime na protokołach transportowych TCP i UDP obsługiwaną przez samego inetd oraz usługę o nazwie login (*de facto* jest to znany nam już rlogin) osiągalną jak widać tylko poprzez TCP i obsługiwaną przez zewnętrznego demona `in.rlogind`, uruchamianego przez inetd oddzielnie dla każdego klienta z programu /usr/sbin/in.rlogind z uprawnieniami użytkownika root. Usuwając odpowiednią linię z pliku konfiguracyjnego dezaktywuje się wybraną usługę (nie jest ona dostępna z zewnątrz, gdyż inetd nie będzie odbierał zgłoszeń kierowanych do tej usługi). Dezaktywacja usług zbędnych bądź wręcz zagrażających utrzymaniu bezpieczeństwa systemu operacyjnego jest elementem podnoszenia poziomu bezpieczeństwa całej sieci.

Należy pamiętać, że niezależnie od pracy demona inetd, użytkownicy mogą uruchamiać procesy obsługujące usługi sieciowe (swoje własne serwery usług), które – niekontrolowane – mogą naruszać stan bezpieczeństwa. Na skromne pocieszenie pozostaje jedynie fakt, iż system nie pozwoli zwykłemu użytkownikowi na uruchomienie procesu obsługującego zarezerwowane usługi systemowe oraz dodatkowo wszystkie te usługi, które aktualnie są już obsługiwane np. przez inetd (również te kierowane przez inetd do innych procesów).

Niestety, nawet jeśli dana usługa sieciowa jest obsługiwana przez inetd, demon ten nie jest wyposażony w żadne mechanizmy kontroli połączeń z tymi usługami, w szczególności inetd nie potrafi zezwalać na dostęp do usług tylko wybranym systemom zdalnym, np. tylko tym komputerom, które należą do tej samej sieci lokalnej lub które administrator uzna za godne zaufania. Tym problemem może się więc zająć co najwyżej sam proces serwera danej usługi, co po pierwsze wymaga posiadania odpowiednio dostosowanego do tego celu serwera dla każdej usługi oddzielnie, a po wtóre przy dużej liczbie usług jest niezwykle trudno utrzymać spójną politykę kontroli dostępu do nich. Ostatecznie więc stosuje się inne rozwiązanie, wprowadzając proces pośredniczący pomiędzy inetd a serwerem konkretnej usługi. Proces ten – tcpd – nazywany popularnie TCP_Wrapper – zajmuje się właśnie kontrolą dostępu do usług obsługiwanych przez inetd. Pobiera on dwie listy: jawnych zezwoleń na dostęp do usług (z pliku /etc/hosts.allow) i jawnych zakazów dostępu do usług (z pliku /etc/hosts.deny), a następnie konfrontuje każde nowe zgłoszenie z danymi zapisanymi najpierw na tej pierwszej, a o ile to nie przynosi rozstrzygnięcia – na drugiej z nich.

TCP_Wrapper jest wywoływany przez inetd, stąd wymaga on dość oczywistej modyfikacji pliku konfiguracyjnego /etc/inetd.conf, np.:

```
#<service> tli <proto> <flags> <user>  <server_pathname>    <args>

echo     stream  tcp   nowait  root    internal
echo     dgram   udp   wait    root    internal
daytime  stream  tcp   nowait  root    internal
daytime  dgram   udp   wait    root    internal

ftp      stream  tcp   nowait  root   /usr/sbin/tcpd /usr/sbin/ftpd
telnet   stream  tcp   nowait  root   /usr/sbin/tcpd /usr/sbin/in.telnetd
login    stream  tcp   nowait  root   /usr/sbin/tcpd /usr/sbin/in.rlogind
finger   stream  tcp   nowait  nobody /usr/sbin/tcpd /usr/sbin/cfingerd
```

Przykładowa konfiguracja uprawnień dostępu do usług sieciowych może wyglądać następująco:

plik /etc/hosts.allow:

```
ftpd:     ALL
cfingerd: LOCAL
ALL:      localhost   spica.cs.put.poznan.pl   150.254.27.45 \
          omikron@gumis.sc.cs.put.poznan.pl \
          150.254.32.0/255.255.254.0
```

plik /etc/hosts.deny:

```
ALL: PARANOID
ALL: ALL : twist (/bin/echo "Sorry. Service %d is disabled for %u@%h.")
```

Konfiguracja ta zapewnia dostęp do usługi ftp (obsługiwanej przez demona ftpd) z każdego systemu zdalnego, do usługi finger (demon cfingerd) tylko z sieci lokalnej i do każdej innej usługi lokalnie, z komputerów spica.cs.put.poznan.pl oraz 150.254.27.45, użytkownikowi omikron z systemu gumis.sc.cs.put.poznan.pl oraz każdemu komputerowi z sieci 150.254.32.0 i 150.254.33.0 (!). Dostęp zostanie natomiast odrzucony bez względu na żądaną usługę (ALL) jeśli podany w zgłoszeniu adres domenowy nie odpowiada adresowi IP zgłaszającego się klienta (PARANOID) oraz w każdym innym przypadku – z wysłaniem do klienta komunikatu o odmowie dostępu.

Zadania:

1. Korzystając z uprawnień administracyjnych, skonfiguruj lokalny system operacyjny tak aby umożliwiał dostęp jedynie do usługi rlogin.
2. Skonfiguruj lokalny system operacyjny tak aby umożliwiał dostęp jedynie do usługi rlogin i jedynie z sąsiedniego komputera w laboratorium.
3. Skonfiguruj lokalny system operacyjny tak aby umożliwiał dostęp jedynie do usługi rlogin jedynie z bieżącego laboratorium za wyjątkiem sąsiedniego komputera. W przypadku próby dostępu z sąsiedniego komputera użytkownik powinien otrzymać odpowiedni komunikat o zablokowanym dostępie.