Domeny zaufania i zabezpieczanie usług sieciowych w systemie Linux za pomocą programu tcpd

1 Wprowadzenie

W rozbudowanych środowiskach sieciowych w których istnieje wiele usług, wielu użytkowników oraz potencjalnie wiele różnych mechanizmów kontroli dostępu, praca użytkowników obarczona zostaje potrzebą wielokrotnego uwierzytelniania się przed różnymi serwerami usługowymi. Taki stan rzeczy nie jest pożądany, co najmniej z kilku powodów. Po pierwsze zabiera czas przewidziany na efektywną pracę. Po drugie wielokrotne uwierzytelnianie się budzi niechęć użytkowników dla których kwestie bezpieczeństwa w większości sytuacji nie są priorytetowe. Po trzecie wielokrotne uwierzytelnianie może prowadzić do sytuacji w której użytkownicy zamiast pamiętać wszystkie klucze, hasła itd. Zwyczajnie zapisują je na różnych nośnikach informacji i robią to najczęściej bez zachowania należytych środków ostrożności.

Mechanizm domeny zaufania w której wykorzystywana jest idea jednokrotnego uwierzytelniania*(ang. single sign-on*) stanowi receptę na opisane powyżej problemy.

Druga część niniejszego opracowania będzie dotyczyła programu określanego mianem TCP Wrapper. W systemie Linux jest to program wykonywalny `tcpd` oraz dwa pliki konfiguracyjne w których zapisywane są reguły bezpieczeństwa dla programu `tcpd`. Mechanizm TCP Wrapper mimo, że nie jest ani najnowszy ani bardzo skomplikowany opiera się na prostej zasadzie według której ruch sieciowy nadchodzący, który ma trafić do odpowiedniego programu jest najpierw transparentnie dla tego programu sprawdzany na wypadek niezgodności z regułami bezpieczeństwa zapisanymi w plikach konfiguracyjnych. Naruszenie reguł spowoduje zablokowanie ruchu sieciowego skierowanego do danego programu i tym samym niedopuszczenie do potencjalnego nadużycia programu.

Słowa kluczowe: domena zaufania, jednokrotne uwierzytelnianie, tcpd, xinetd, bastion, twierdza

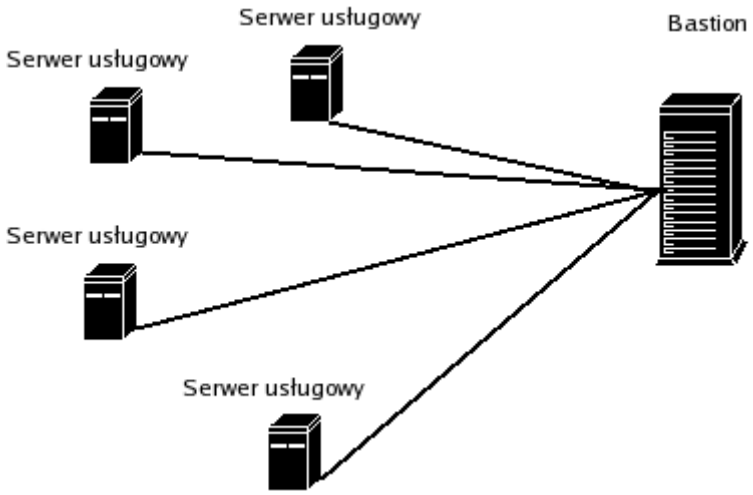
2 Zastosowanie domen zaufania

Domena zaufania to środowisko sieciowe wraz z działającymi w nim usługami w którym występuje jednolity mechanizm kontroli dostępu. W przypadku opisywanej niżej domenie zaufania mechanizm kontroli opiera się na zaufaniu jakim darzą wyszczególnionego hosta pozostałe hosty jeżeli chodzi o uwierzytelnianie użytkowników. W tak zorganizowanej domenie grupa hostów/serwerów/usług ufa wyszczególnionemu hostowi, że ten przeprowadzi silne uwierzytelnianie. Zaufanie grupy hostów opiera się na idei według której, jeśli użytkownik uwierzytelnił się w wystarczający sposób przed wybranym hostem to znaczy, że jest tym za kogo się podaje i nie ma potrzeby ponownie przeprowadzać procedury uwierzytelniania przy dostępie do innych hostów. Wybrany host, który jest odpowiedzialny za silne uwierzytelnianie użytkowników określany jest mianem twierdzy lub bastionu.

Utworzenie domeny zaufania i wprowadzenie mechanizmu jednokrotnego uwierzytelniania pozwoli zmniejszyć uciążliwość korzystania z rozproszonego środowiska sieciowego w którym jest zlokalizowanych wiele usług wymagających kontroli dostępu. Myśląc o wadach należy wziąć pod uwagę, co się stanie, gdy wybrany host przeprowadzający uwierzytelnianie

przestanie być dostępny dla użytkowników. Prawdopodobnie spowoduje to efekt odmowy usługi*(ang. Denial of Service*). Kolejny ważny problem to kompromitacja wybranego hosta, który zapewnia silne uwierzytelnianie. Czy spowoduje to natychmiastową kompromitację całej domeny? Przed wprowadzeniem mechanizmu jednokrotnego uwierzytelniania należy dokładnie rozpatrzyć wszystkie zalety i wady takiego rozwiązania.

3 Przykładowa domena zaufania



Rysunek 1 Przykładowa domena zaufania.

3.1 Polecenie `rlogin` w systemie Linux

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. Poniżej zaprezentowano użycie polecenia `rlogin`:

```
localhost>rlogin remotehost
```

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`:

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

Hasło jest transmitowane tekstem jawnym, podobnie jak w protokole TELNET, co stanowi poważne zagrożenie poufności hasła do 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, aby 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 podania hasła. Jest to użyteczne, gdy ten sam użytkownik posiada różne konta(być może o różnych nazwach użytkownika) na kilku systemach i nie chce wystawiać swojego hasła na transmisje niechronionym kanałem lub chce umożliwić zdalne wywoływanie poleceń na swoim koncie. Poniżej zaprezentowano przykładową konfigurację pliku `~/.rhosts` na koncie `Michal` w systemie `Uran`:

```
jowisz
dcslab iksinski
```

Taka lista pozwala zalogować się bez wymogu podania hasła użytkownikowi systemu `jowisz` posiadającemu konto o identycznej nazwie(czyli `michal`) oraz użytkownikowi `iksinski` z systemu `dcslab`.

Mechanizm tak skonfigurowanego zaufania jest jednak bardzo niebezpieczny, gdyż wystawia konto na ataki podszywania się pod zaufanego hosta.

4 Zabezpieczanie usług sieciowych programem `tcpd`

W systemie Linux istnieje szereg prostych usług sieciowych, które nie posiadają rozbudowanych mechanizmów kontroli dostępu lub w ogóle nie posiadają żadnych. Do grupy takich usług można zaliczyć tzw. small services np. finger, chargen, rlogin, echo czy tftp.
W większości sytuacji usługi te w obecnych systemach Linux/Unix są domyślnie wyłączone więc zagrożenie z ich strony jest nikłe. Czasami zdarzają się jednak sytuacje, gdy niektóre usługi z tej grupy pełnią ważną rolę w systemach komputerowych i należy wdrożyć mechanizm kontroli dostępu dla takiej usługi. Nie tylko proste usługi sieciowe mogą być chronione przez program `tcpd`. Usługa bez której nie byłaby możliwa praca zdalna na maszynach Linux/Unix czyli ssh również może posiadać dodatkowe zabezpieczenie w formie reguł bezpieczeństwa dla programu `tcpd`. W większości sytuacji jednak program tcpd stosuje się jako mechanizm kontroli dostępu usług sieciowych, które nie posiadają wystarczających wbudowanych mechanizmów kontroli dostępu. Dlatego też znajomość obsługi programu `tcpd` może być konieczna w niektórych systemach komputerowych.

4.1 Podstawy

Program `tcpd` służy do zabezpieczania usług sieciowych poprzez sprawdzanie zgodności przychodzącego ruchu sieciowego z regułami bezpieczeństwa zdefiniowanymi w dwóch plikach konfiguracyjnych: `/etc/hosts.allow` i `/etc/hosts.deny`. W pierwszej kolejności należy zrozumieć w jaki sposób program `tcpd` podejmuje decyzje odnośnie przepuszczania i blokowania ruchu sieciowego skierowanego do danych usług sieciowych:

- dostęp zostanie przyznany gdy w pliku `/etc/hosts.allow` znajdzie się odpowiednia reguła bezpieczeństwa
- w przeciwnym wypadku dostęp zostanie zablokowany, gdy w pliku `/etc/hosts.deny` znajdzie się odpowiednia reguła bezpieczeństwa
- w przeciwnym wypadku dostęp zostanie przyznany.

Należy zauważyć, że tak skonstruowany mechanizm sprawdzania możliwości przyznania dostępu powoduje, że brak dopasowania w obu plikach konfiguracyjnych spowoduje przyznanie dostępu.

Komplet niezbędnych informacji o programie `tcpd` i jego możliwościach można znaleźć w podręczniku systemowym systemu Linux/Unix wykonując polecenia:

- `man tcpd` - ogólne informacje o programie `tcpd`
- `man 5 hosts_access` - informacje o plikach konfiguracyjnych dla programu `tcpd`
- `man 5 hosts_options` - informacje o możliwych opcjach jakie można użyć w plikach konfiguracyjnych

4.2 Przykład zabezpieczania usługi `tftp` programem `tcpd`

Konfiguracja programu `tcpd` zostanie pokazana na przykładzie zabezpieczania usługi `tftp`. Usługa tftp jest prostszą wersją protokołu ftp wykorzystywaną przez wiele urządzeń sieciowych. Dlatego też ta usługa posłuży jak przykład. Usługa tftp oprócz aktywnych urządzeń

sieciowych jest również wykorzystywana do tzw. *provisioningu*. *Provisioning* jest to możliwość dostarczania usług, różnego rodzaju zasobów użytkownikom lub urządzeniem np. modemom kablowym, które w ten sposób pobierają plik z konfiguracją. Dlatego też usługa `tftp` jest nadal aktywnie wykorzystywana w rozbudowanych środowiskach sieciowych np. w sieciach ISP(*ang. Internet Service Provider*).

Usługę tftp można uruchomić jako samodzielną usługę sieciową(tryb *standalone*) lub po przez superdemona inetd lub xinetd. W niniejszym opracowaniu usługa `tftp` będzie uruchomiona za pomocą superdemona xinetd. Poniżej zaprezentowano przykładową zawartość pliku konfiguracyjnego dla demona xinetd do uruchomienia usługi tftp:

```
service tftp
{
      per_source  = 5
      socket_type = dgram
      protcol     = udp
      wait        = yes
      user        = root
      server      = /usr/sbin/tcpd
      server_args = in.tftpd -t 60 -s /var/tftp -u tftpd
      disable     = no
      banner_fail = /etc/xinetd/fail.banner
      cps         = 100 10

}
```

W konfiguracji przedstawionej powyżej należy w szczególności zwrócić uwagę na dwa parametry. Parametr `server` wskazuje na plik wykonywalny programu usługi sieciowej, którą ma uruchomić xinetd np. tftp, finger, rlogin. Z uwagi, że zależy nam aby usługa `tftp` była chroniona przez program `tcpd` w parametrze server podajemy ścieżkę do pliku wykonywalnego `tcpd` a nie faktycznej usługi która ma być uruchomiona przez demona xinetd. Parametr `server_args` zawiera parametry jakie mają być przekazane do programu podanego w parametrze `server`. W tym przypadku są to parametry dla programu `tcpd` oraz zestaw parametrów dla usługi `tftp`, którą uruchomi program `tcpd`.
W przedstawionym powyżej przykładzie parametr `server_args` zawiera nazwę wykonywalnego pliku serwera usługi `tftp`, plik `in.tftpd`. Drugi parametr określa jak długo po uruchomieniu programu `in.tftpd` ma on być aktywny. Trzeci parametr wymusza przejście do katalogu `/var/tftp` z zastosowaniem mechanizmu chroot. Ostatni parametr wymusza zmianę użytkownika z jakim będzie działał program `in.tftpd` na użytkownika tftpd.
Superdemon xinetd posiada szereg przydatnych funkcjonalności odnośnie kwestii bezpieczeństwa uruchamianych usług sieciowy. Autor niniejszego opracowania zaleca zapoznanie się z nimi w celu lepszego zrozumienia możliwości xinetd.
Komplet informacji na temat superdemona xinetd można znaleźć na stronie domowej projektu xinted[1].

Druga etap zabezpieczania usługi `tftp` to przygotowanie reguł bezpieczeństwa dla programu `tcpd`. W pierwszej kolejności należy zastanowić się, komu będzie przyznawany dostęp do usługi `tftp` i czy chcemy aby działanie usługi `tftp` było zapisywanie w plikach logu a jeśli tak

to w jaki sposób. Powinno nam zależeć na możliwe ścisłym określeniu, kto może korzystać z usługi `tftp` oraz powinniśmy posiadać informacje o działaniu samej usługi. Poniżej zaprezentowano przykładowe pliki `/etc/hosts.allow` i `/etc/hosts.deny`, które poruszają wyżej wymienione kwestie:

```
#plik /etc/hosts.allow

in.tftpd: 192.168.1. : spawn (logger -p auth.notice -t %d
access_granted to hostname %c ip %a ) &

#lub prostrza wersja bez logowania
in.tftpd: 192.168.1.
```

```
#plik /etc/hosts.deny

in.tftpd: ALL EXCEPT 192.168.1. : spawn (logger -p auth.notice -t
%d permission denied to hostname %c ip %a ) &


#lub globalne zablokowanie dostepu

ALL: ALL
```

W powyższych przykładach pokazano w jaki sposób można ograniczyć dostęp do usługi `tftp` tylko dla hostów z adresami 192.168.1. Brak ostatniego oktetu w adresie IP 192.168.1 sugeruje, że będzie on pomijany przy sprawdzaniu możliwości dostępu do usługi, liczą się tylko pierwsze trzy oktety.
W powyższych przykładach każda akcja dostępu do usługi `tftp` zakończona otrzymaniem dostępu bądź odmową jest odnotowywana w plikach logu poprzez tworzenie odpowiedniego wpisu. Należy zauważyć, że w środowiskach sieciowych w których usługa `tftp` jest intensywnie używana odnotowywanie każdej udanej próby dostępu do usługi `tftp` będzie prowadzić do tworzenia wielu wpisów w pliku logów systemowych, co z kolei będzie prowadziło do utrudnionej analizy działania usługi `tftp` z uwagi na ilość zgromadzonych danych. Dlatego w uzasadnionych przypadkach zalecane jest ograniczenie odnotowywania udanych prób uzyskania połączenia lub nawet zaprzestanie zbierania takich informacji. Zalecane jest jednak zbieranie informacji o nieudanych próbach uzyskania dostępu do usługi `tftp` z uwagi na potencjalny atak na tą usługę lub na demona xinted prowadzący w skrajnych przypadkach do efektu odmowy dostępu(atak typu DOS) dla uprawnionych hostów z uwagi na wyczerpanie zasobów systemowych przyznanych usłudze `tftp`.

5 Podsumowanie

Domeny zaufania mogą w wydatny sposób pomóc w ułatwieniu korzystania z usług sieciowych w rozbudowanych środowiskach sieciowych. Duże ułatwienie niesie za sobą również duże ryzyko kompromitacji serwerów usługowych z uwagi na brak niezależnej procedury uwierzytelniania użytkowników. Dlatego też przed wdrożeniem takiego rozwiązania należy szczegółowo rozważyć wszystkie za i przeciw gdyż kompromitacja usług, ujawnienie poufnych danych lub ich kradzież mogą okazać się bardzo kosztowne.

W wielu środowiskach sieciowych nadal działają usługi, które wydawałoby się odeszły w zapomnienie gdyż przeważająca większość użytkowników nie korzysta z nich. Niektóre z tych usług pełnia krytyczną role w swoich środowiskach i należy dbać o ich bezpieczeństwo. Program `tcpd` może okazać się pomocny w zabezpieczaniu takich usług.

6 Zadania

- Zapoznanie się w możliwościami programu `rlogin`
- Utworzenie przykładowej domeny zaufania i wykorzystanie programu `rlogin` do zdalnego dostępu w hosta Bastion do dowolnego hosta w domenie zaufania.
- Zapoznanie się z możliwościami programu xinted.
- Zapoznanie się z możliwościami usługi `tftp`.
- Zapoznanie się z możliwościami programu `tcpd`
- Konfiguracja usługi `tftp` w oparciu o program `tcpd` i pliki konfiguracyjne `hosts.allow` i `hosts.deny`.
- Zapoznanie się z programem `logger`.

7 Problemy do dyskusji

- Jakie zagrożenia i ułatwienia niesie utworzenie domeny zaufania w oparciu o mechanizm jednokrotnego uwierzytelniania i program `rlogin`.
- Czy możliwe jest zablokowanie możliwości samodzielnego dodawania zaufanych hostów w domenie zaufania przez nieuprzywilejowanych użytkowników?
- Jakie wbudowane mechanizmy bezpieczeństwa posiada usługa `tftp`?
- Jakie mechanizmy bezpieczeństwa posiada superdemon xinetd?
- Czy używając programu `tcpd` można odesłać jakąś wiadomość użytkownikowi próbującemu uzyskać dostęp, jeśli tak to w jaki sposób?
- Do czego służy mechanizm chroot?

8 Bibliografia

- [1] Strona domowa projektu xinted: http://www.xinted.org
- Strona podręcznika systemowego dla programu `logger: man logger`
- Strona podręcznika systemowego dla programu `rlogin` i `in.rlogind: man rlogin, man in.rlogind`
- Strona podręcznika systemowego dla programu `tcpd: man tpcd`
- Strona podręcznika systemowego dla programu `chroot: man chroot`