Ograniczone środowisko wykonywania aplikacji, ograniczenie powłoki systemu operacyjnego środowisk serwerowych, delegacja uprawnień administracyjnych (sudo, CAP)

1. Wprowadzenie

Ograniczanie środowiska wykonywania aplikacji i powłoki systemu umożliwia ograniczanie zasobów wykorzystywanych przez aplikację oraz przez konkretnych użytkowników. Wykorzystywany do tego celu jest mechanizm limitów w systemach Linux/Unix.

Delegacja uprawnień administracyjnych wykorzystuje dwa mechanizmy. Pierwszym z nich jest mechanizm bitów SUID i SGID, drugim mechanizm SUDO.

Celem ćwiczenia jest zrozumienie wszystkich wymienionych powyżej mechanizmów i zastosowanie ich w praktyce.

2. Informacje o mechanizmie limitów

Mechanizm limitów wykorzystywany w systemie Linux umożliwia ograniczanie liczby otwartych plików, liczby aktywnych procesów i wiele innych dla każdego z użytkowników. Jednocześnie wiele aplikacji korzysta ze osobnych kont użytkowników i dzięki temu możemy ograniczyć konkretne aplikacje.

Mechanizm ten często nazywamy ulimit (*eng. user limit),*czyli limity użytkownika.

Limity narzucane są przez administratora systemu, aczkolwiek, użytkownik może je troszkę modyfikować, dokładniej może jedynie jeszcze bardziej siebie ograniczyć.

Możliwości mechanizmu limitów

Mechanizm limitów może ograniczać następujące zasoby:

- wielkość pliku core,
- wielkość pamięci zajmowanej przez dane,
- maksymalną wielkość pliku,
- maksymalną ilość zajmowanej pamięci,
- maksymalną liczbę jednocześnie otwartych plików,
- maksymalną wielkość zajmowanych zasobów,
- maksymalną wielkość stosu,
- maksymalny czas procesora, który może wykorzystać,
- maksymalną liczbę równoczesnych procesów,
- ograniczenie przestrzeni adresowej(*),

- maksymalną liczbę jednoczesnego zalogowania się użytkownika do systemu(*),
- maksymalną liczbę zalogowań się do systemu(*),
- priorytet z jakim mają być uruchamiane procesy użytkownika(*),
- maksymalna liczba blokad na pliki,
- maksymalna liczba oczekujących sygnałów do wysłania,
- maksymalna ilość pamięci użyta dla kolejek typu POSIX,
- maksymalny priorytet, do którego może być zwiększony priorytet procesu użytkownika(*),
- maksymalny priorytet czasu rzeczywistego(*).

Ograniczenia z gwiazdkami (*) są jedynie dostępne z poziomu konfiguracji limitów poprzez PAM.

Ograniczenia te można nadawać wykorzystując aplikację **ulimit** z odpowiednimi przełącznikami, jak również można wykorzystać mechanizm PAM, który umożliwia proste skonfigurowanie limitów dla użytkowników oraz grup systemu operacyjnego.

Podsumowanie mechanizmu limitów

Mechanizm limitów może znacznie ograniczyć środowisko dostępne dla użytkownika, można go używać pod warunkiem posiadania wiedzy, którzy użytkownicy wymagają jakich uprawnień. Oczywiście często trudno jest to w prosty sposób zdefiniować, poza tym trzeba brać pod uwagę, że użytkownicy nie lubią żadnych ograniczeń, więc nie można ich się pytać jak bardzo można ich ograniczyć, tylko dochodzić do tego co użytkownik powinien móc robić w systemie. Aplikacje również można w ten sposób ograniczać, na przykład serwer WWW działa na uprawnieniach konta *wwwrun*, w ten sposób można ograniczyć to konto jednocześnie ograniczając serwer WWW.

3. Informacje o mechanizmie SUDO

Mechanizm SUDO ma za zadanie umożliwienie wykonywania aplikacji z innymi uprawnieniami, na przykład z uprawnieniami użytkownika *root*. Mechanizm ten opiera się na aplikacji poprzez którą mogą być uruchamiane inne aplikacje z innymi uprawnieniami. Aplikacja SUDO jest konfigurowalna przez administratora systemu i to on może ustalić jakie aplikacje mogą być uruchamiane z innymi uprawnieniami, a nawet z jakimi uprawnieniami oraz przez jakiego użytkownika.

Możliwości mechanizmu na podstawie pliku konfiguracyjnego

Plik konfiguracyjny znajduje się w */etc/sudoers*i jest dostępny jedynie dla administratora systemu. Dodatkowo powinien być on edytowany za pomocą polecenia *visudo*.

Plik ten składa się z dwóch części:

- aliasy (podstawowe zmienne);

- specyfikację użytkownika (które określają kto może co uruchomić).

Wyróżniamy 4 typy aliasów:

- user_alias
- runas_alias
- host_alias
- cmnd_alias

Każdy z typów może zawierać troszkę inne wartości, jednak w ogólności są one proste do zapamiętania, na przykład user_alias może zawierać nazwy użytkowników, nazwy grup oraz inne. Szczegóły dotyczące aliasy znajdują się z podręczniku użytkownika SUDOERS(5). Poniżej kilka przykładów:

User_alias FULLTIMERS = adam, adrian, ania

User_alias ADVUSER = bartek, piort

Runas_alias OP = root

Runas_alias DB = oracle

Host_alias ORACLE = dblab, oracle, baza

Host_alias CS = 150.254.0.0/16

Cmnd_alias KILL = /usr/bin/kill

Cmnd_alias SHELLS = /usr/bin/sh, /usr/bin/bash

Specyfikacja użytkownika określa jaki użytkownik może wykonać określone aplikacje z określonymi uprawnieniami. Domyślnie każda aplikacja będzie uruchamiana z uprawnieniami administratora, chyba, że podamy inaczej. Również można określić czy użytkownik będzie musiał podawać hasło, aby uzyskać dostęp do danej komendy.

Przykłady:

FULLTIMERS CS = NOPASSWD: SHELLS, KILL

określa, że użytkownicy “FULLTIMERS” mogą na komputerach “CS” bez hasła wykonać komendy “SHELLS” i “KILL” z uprawnieniami *root*.

bartek ALL = (DB) NOPASSWD: SHELLS

użytkownik *bartek* może na wszystkich komputerach wykonać bez hasła komendy “SHELLS” z uprawnieniami “DB”

piotr ALL, !ORACLE = (www) ALL, (OP) KILL, (DB) /usr/bin/ls

użytkownik *piotr* może na wszystkich komputerach za wyjątkiem komputerów “ORACLE” wykonywać wszystkie komendy z uprawnieniami *www*, natomiast komendy “KILL” z uprawnieniami “OP”, a komendę “/usr/bin/ls” z uprawnieniami “DB”

Większą liczbę przykładów i opisy można znaleźć w podręczniku użytkownika SUDOERS(5).

Podsumowanie mechanizmu SUDO

Mechanizm SUDO pozwala dopuszczać użytkowników do wykonywania aplikacji z innymi uprawnieniami, przy czym nie ma znaczenia czy aplikacja na to pozwala czy nie, gdyż jest to on niej nie zależne. Dodatkowo administrator systemu ma pełną kontrolę nad użytkownikami, którzy mogą posiadać możliwości wykorzystywania tego mechanizmu oraz ścisłe ograniczenia w jaki sposób mogą z nich korzystać.

4. Informacje o mechanizmie SUID i SGID

Mechanizm SUID i SGID jest najprostszym z wyżej wymienionych. Jest on także bardzo prosty i opiera się na systemie plików, gdyż z każdym plikiem możemy określić dodatkowy bit uprawnień, określający SUID i SGID. Na podstawie ustawionego tego bitu system operacyjny wykonuje aplikację z innymi uprawnieniami, dokładniej wykonuje ją z uprawnieniami właściciela aplikacji.

Jeśli ustawimy bit SUID to aplikacja zostanie wykonana z uprawnieniami właściciela aplikacji, niezależnie od tego kto ją wykona, pod warunkiem, że posiada uprawnienia do jej wykonania.

Natomiast bit SGID określa, że aplikacja zostanie wykonana z uprawnieniami grupy właściciela aplikacji, która to jest określona jako grupa danego pliku.

Niestety mechanizm ten nie posiada żadnej kontroli, dlatego uważany jest za bardzo niebezpieczny i jedynie niektóre komendy powinny mieć ustawiony bit SUID, przykładem takiej aplikacji jest */bin/ping* lub inne komendy wymagające wysokich uprawnień.

Przypisanie bitu SUID i SGID do pliku wykonywane jest za pomogą programu *chmod*, na przykład *chmod u+s plik* spowoduje dodanie bitu SUID, natomiast *chmod g+s plik* spowoduje przypisanie bitu SGID.

5. Zadania

Wykorzystać zaprezentowane powyżej mechanizmy w celu:

- wykorzystaj mechanizm limitów do ograniczenia użytkownika *secoff* w następujący sposób: ustal możliwość jednokrotnego logowana do systemu, dopisz ograniczenie na liczbę procesów oraz maksymalną wielkość nowo tworzonego pliku. Zweryfikuj poprawność działania wpisanych limitów,
- wykorzystaj mechanizm SUDO, aby umożliwić użytkownikowi *secoff* uruchamiać z uprawnieniami administratorskimi aplikacji *ls* oraz aplikacji find z uprawnieniami użytkownika *wwwrun*,
- wykorzystaj mechanizm SUDO do ustanowienia uprawnień dla *users* do wykonywania polecenia ls z uprawnieniami użytkownika secoff oraz aplikacji minicom z uprawnieniami administratorskimi,
- wykorzystaj mechanizm SUID i SGID do ustanowienia podobnych uprawnień jak w powyższych dwóch punktach, odpowiedz na pytanie na ile jest to możliwe i czy na pewno daliśmy tylko takie uprawnienia?
- wykorzystaj aplikację *find* do znalezienie wszystkich aplikacji posiadających ustawiony jeden z bitów SUID lub SGID, przedstaw listę prowadzącemu wraz z uzasadnieniem dlaczego akurat te aplikacje mają ustawiony ten bit.

6. Problemy do dyskusji

- czy przestawione powyżej mechanizmy posiada wady, jakie?
- jakie są zalety powyższych mechanizmów?
- czy można by rozbudować któryś z mechanizmów, jeśli tak to o jaką funkcjonalność?
- które z mechanizmów powinien wykorzystywać administrator systemu chcący zapewnić bezpieczeństwo systemu, a których powinien się wystrzegać?

7. Bibliografia

[SUDO] Linux SUDO http://www.gratisoft.us/sudo/

[SUID] SUID-SGID http://www.homepage.montana.edu/~unixuser/051602/SUID.html

[Ulimit] podręcznik użytkownika bashbuiltins(1)