КОНТРОЛЬНА РОБОТА № 2

Безпека в мережЕВій операційній системі FreeBSD

Мета роботи: Одержати практичні навички по забезпеченню безпеки в мережевій операційній системі FreeBSD4.2

Теоретичні положення. Принципи захисту

Оскільки ОС FreeBSD з самого свого зародження задумувалась як багатокористувацька операційна система, у ній завжди була актуальною проблема авторизації доступу різних користувачів до файлів файлової системи. Під авторизацією доступу ми розуміємо дії системи, які дозволяють або не дозволяють доступ певного користувача до певного файлу в залежності від прав доступу користувача і обмежень доступу встановлених для файла. Схема авторизації доступу, яка застосована в ОС FreeBSD, настільки проста і зручна й одночасно настільки потужна, що стала фактично стандартом сучасних операційних систем (які не претендують бути системами з багаторівневим захистом).

Ідентифікатори користувача та групи користувачів

З кожним виконуваним процесом в ОС FreeBSD зв’язується реальний ідентифікатор користувача (real user ID), діючий ідентифікатор користувача (effective user ID) і збережений ідентифікатор користувача (saved user ID). Всі ці ідентифікатори встановлюються за допомогою системного виклику setuid, який можна виконувати тільки в режимі суперкористувача. Аналогічно, з кожним процесом зв’язуються три ідентифікатори групи користувачів - real group ID, effective group ID і saved group ID. Ці ідентифікатори встановлюються привілейованим системним викликом setgid. При вході користувача в систему програма login провіряє, чи користувач зареєстрований в системі і знає правильний пароль (якщо він встановлений), створює новий процес і запускає в ньому потрібний для даного користувача shell. Але перед цим login установлює для знов створеного процесу ідентифікатори користувача і групи, використовуючи для цього інформацію, збережену в файлах /etc/passwd і /etc/group. Після того, як з процесом зв’язані ідентифікатори користувача і групи, для цього процесу починають діяти обмеження для доступу до файлів. Процес може отримати доступ до файлу або виконати його (якщо файл містить виконувану програму) тільки в тому разі, якщо збережені обмеження доступу, які відносяться до файлу, дозволяють це зробити. Зв'язані з процесом ідентифікатори передаються створюваним їм процесам, поширюючи на них ті ж обмеження. Однак у деяких випадках процес може змінити свої права за допомогою системних викликів setuid і setgid, а іноді система може змінити права доступу процесу автоматично.

Розглянемо, наприклад, наступну ситуацію. У файл /etc/passwd заборонений запис усім, крім суперкористувача (суперкористувач може писати в будь-який файл). Цей файл, крім іншого, містить паролі користувачів і кожному користувачу дозволяється змінювати свій пароль. Мається спеціальна програма /bin/passwd, що змінює паролі. Однак користувач не може зробити це навіть за допомогою цієї програми, оскільки запис у файл /etc/passwd заборонений. У системі FreeBSD ця проблема розв’язується в такий спосіб. При виконуваному файлі може бути зазначено, що при його запуску повинні встановлюватися ідентифікатори користувача і/або групи. Якщо користувач запитує виконання такої програми (за допомогою системного виклику exec), то для відповідного процесу встановлюються ідентифікатор користувача, що відповідає ідентифікатору власника виконуваного файлу і/або ідентифікатор групи цього власника. Зокрема, при запуску програми /bin/passwd процес одержить ідентифікатор суперкористувача, і програма зможе зробити запис у файл /etc/passwd. І для ідентифікатора користувача, і для ідентифікатора групи реальний ID є істинним ідентифікатором, а діючий ID - ідентифікатором поточного виконання. Якщо поточний ідентифікатор користувача відповідає суперкористувачу, то цей ідентифікатор і ідентифікатор групи можуть бути перевстановлені в будь-яке значення системними викликами setuid і setgid. Якщо ж поточний ідентифікатор користувача відрізняється від ідентифікатора суперкористувача, то виконання системних викликів setuid і setgid приводить до заміни поточного ідентифікатора істинним ідентифікатором ( користувача або групи відповідно).

Захист файлів

Як прийнято в багатокористувацькій операційній системі, у FreeBSD підтримується однаковий механізм контролю доступу до файлів і довідників файлової системи. Кожен процес може одержати доступ до деякого файлу в тому і тільки в тому випадку, якщо права доступу, описані при файлі, відповідають можливостям даного процесу.

Захист файлів від несанкціонованого доступу в ОС FreeBSD грунтується на трьох фактах. По-перше, з будь-яким процесом, що створює файл, асоційований деякий унікальний у системі ідентифікатор користувача (UID - User Identifier), що надалі можна трактувати як ідентифікатор власника знов створеного файлу. По-друге, з кожен процесом, що намагається одержати деякий доступ до файлу, зв’язана пара ідентифікаторів поточні ідентифікатори користувача і його групи. По-третє, кожному файлу однозначно відповідає його описувач - i-вузол.

На останньому факті варто зупинитися більш докладно. Важливо розуміти, що імена файлів і файли як такі - це не одне і теж . Зокрема, при наявності декількох твердих зв'язків з одним файлом кілька імен файлу реально представляють один і той же файл і асоційовані з одним і тим ж i-вузлом. Будь-якому використовуваному у файловій системі i-вузлу завжди однозначно відповідає один і тільки один файл. I-вузол містить досить багато різноманітної інформації (велика її частина доступна користувачам через системні виклики stat і fstat), і серед цієї інформації знаходиться частина, що дозволяє файловій системі оцінити правомірність доступу даного процесу до даного файлу в потрібному режимі.

Загальні принципи захисту однакові для всіх існуючих варіантів системи:

Інформація i-вузла включає UID і GID поточного власника файлу (негайно після створення файлу ідентифікатори його поточного власника установлюються відповідними діючим ідентифікатором процесу-творця, але надалі можуть бути змінені системними викликами chown і chgrp). Крім того, в i-вузлі файла зберігається шкала, у якій відзначено, що може робити з файлом користувач - його власник, що можуть робити з файлом користувачі, що входять у ту ж групу користувачів, що і власник, і що можуть робити з файлом інші користувачі. Дрібні деталі реалізації в різних варіантах системи розрізняються. Володіння файлом визначає той набір операцій, що користувач може зробити з файлом. Частина з них, такі як зміна прав чи доступу власника файлу (табл. 1.1), може здійснювати тільки власник (чи суперкористувач), інші операції, такі як читання, запис і запуск на виконання (для файлів, що виконуються) додатково контролюються правами доступу.

Таблиця 1.1. Операції зміни власників файлу

|  |  |  |  |
| --- | --- | --- | --- |
| Операція | Команда | Має право | Виконувати |
|  |  | у системі BSD 4.х | У системі SVR4 |
| Зміна власника-користувача  Зміна власника-групи | chown(1)  chgrp(1) | Суперкористувач  Суперкористувач | Власник файлу  Власник файлу тільки для групи, до якої сам належить (відповідно до POSIX) |

Права доступу до файлу

В операційній системі FreeBSD існують три базових класи доступу до файлу, у кожнім з який установлені відповідні права доступу:

User access (u) Для власника-користувача файлу

Group access (g) Для членів групи, що є власником файлу

Other access (о) Для інших користувачів (крім суперкористувача)

FreeBSD підтримує три типи прав доступу для кожного класу: на читання (read, позначається символом г), на запис (write, позначається символом w) і на виконання (execute, позначається символом х).

За допомогою команди Is -l можна одержати список прав доступу до файлу:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| -rw-r--r-- | 1 | andy | group | 36482 | Dec | 22 | 19:13 | report.txt.1 |
| drwxr-xr-- | 2 | andy | group | 64 | Aug | 15 | 11:03 | temp |
| -rwxr-xr-- | 1 | andy | group | 4889 | Dec | 22 | 15:13 | а. out |
| -rw-r--r-- | 1 | andy | group | 7622 | Feb | 11 | 09:13 | cont.с |

Права доступу лістинга відображаються в першому стовпчику (за винятком першого символу, що позначає тип файлу). Наявність права доступу позначається відповідним символом, а відсутність — символом '-'. Розглянемо, наприклад, права доступу до файлу a.out:

|  |  |  |  |
| --- | --- | --- | --- |
| Тип файлу | Права власника- користувача | Права власника- групи | Права інших користувачів |
| - | rwx | r-x | r-- |
| Звичайний файл | Читання, запис, виконання | Читання і виконання | Тільки читання |

Права доступу можуть бути змінені тільки власником файлу або суперкористувачем (superuser) — адміністратором системи. Для цього використовується команда chmod(l). Нижче приведений загальний формат цієї команди.

chmod [ u g o a ] [ + - = ] [ r w x ] file1 file2 …

Як аргументи команда приймає вказівку класів доступу ('u' — власник-користувач, g' — власник-група, 'о' — інші користувачі, 'а' — усі класи користувачів), права доступу ('г' — читання, 'w' — запис і 'х' — виконання) і операцію, яку необхідно зробити ('+' — додати, '-' — видалити і '=' — привласнити) для списку файлів filel, file2 і т.д. Наприклад, команда

$ chmod g-wx ownfile

позбавить членів групи-власника файлу ownfile права на запис і виконання цього файлу.

В одній команді можна задавати різні права для декількох класів Доступу, розділивши їх комами.

можна установити відразу всі дев'ять прав доступу, використовуючи числову форму команди chmod(l):

$ chmod 754 \*

Число визначається в такий спосіб: потрібно представити права доступу в двійковому виді (0 — відсутність відповідного права, 1 — його наявність) і кожну тріаду, що відповідає класу доступу, у свою чергу перетворити в десяткове число.

|  |  |  |
| --- | --- | --- |
| Власник | Група | Інші |
| Г W Х | 27 X | г - - |
| 111 | 101 | 100 |
| 7 | 5 | 4 |

Таким чином, приведений приклад еквівалентний наступній символьній формі chmod(l)'.

$ chmod u=rwx, g=rx, o=r \*

Значення прав доступу різне для різних типів файлів. Для файлів операції, які можна робити, випливають із самих назв прав доступу. Наприклад, щоб переглянути вміст файлу командою cat(l), користувач повинний мати право на читання (г). Редагування файлу, тобто його зміна, передбачає наявність права на запис (w). Нарешті, для того щоб запустити деяку програму на виконання, потрібно мати відповідне право (х). Файл, що виконується, може бути як скомпільованою програмою, так і скриптом командного інтерпретатора shell. В останньому випадку також знадобиться право на читання, оскільки при виконанні скрипта командний інтерпретатор повинний мати можливість зчитувати команди з файлу. Усе сказане, за винятком, мабуть, права на виконання, що має зміст лише для звичайних файлів і каталогів, справедливо і для інших типів файлів: спеціальних файлів пристроїв, іменованих каналів, і сокетов. Наприклад, щоб мати можливість роздрукувати документ, потрібно мати право на запис у спеціальний файл пристрою, зв'язаний із принтером. Для каталогів ці права мають інший зміст, а для символічних зв'язків вони взагалі не використовується, оскільки контролюються цільовим файлом.

Права доступу для каталогів не настільки очевидні. Це, у першу чергу, зв'язане з тим, що система трактує операції читання і запису для каталогів відмінно від інших файлів. Право читання каталогу дозволяє одержати імена (і тільки імена) файлів, що знаходяться в даному каталозі. Щоб одержати додаткову інформацію про файли каталогу (наприклад, докладний лістинг команди Is -/), системі прийдеться "заглянути" у метадані файлів, що вимагає права на виконання для каталогу. Право на виконання також буде потрібно для каталогу, у який ви захочете перейти (тобто зробити його поточним) за допомогою команди cd (1). Це ж право потрібно мати для доступу до всіх каталогів на шляху до зазначеного. Наприклад, якщо ви установите право на виконання для всіх користувачів в одному зі своїх підкаталогів, він усе рівно залишиться недоступним, поки ваш домашній каталог не буде мати такого ж права.

Права г и х діють незалежно, право х для каталогу не вимагає наявності права м, і навпаки. Комбінацією цих двох прав можна домогтися цікавих ефектів, наприклад, створення "темних" каталогів, файли яких доступні тільки у випадку, якщо користувач заздалегідь знає їхні імена, оскільки одержання списку файлів таких каталогів заборонено. Даний прийом, до речі, використовується при створенні загальнодоступних архівів у мережі (FTP-серверів), коли деякі розділи архіву можуть використовуватися тільки тими, хто знає про наявність того чи іншого файлу в каталозі. Приведемо приклад створення "темного" каталогу.

$ pwd Де ми знаходимося?

home/andrei

$ mkdir darkroom Створимо каталог

$ Is — 1 Одержимо його атрибути

-rwxr--r-- 2 andy group 65 Dec 22 19:13 darkroom

$ chaod a-r+x darkroom Перетворимо його в "темний"каталог

$ Is - 1 Одержимо його атрибути

--wx--x--x 2 andy group 65 Dec 22 19:13 darkroom

$ cp filel darkroom Помістимо в каталог darkroom деякий файл

$ cd darkroom Перейдемо в цей каталог

$ Is -I darkroom Спробуємо одержати лістинг каталогу

-#permi3sion denied На жаль...

$ cat filel Проте, заздалегідь знаючи ім'я файлу (filel), можна працювати з ним (наприклад, прочитати, якщо є відповідне право доступу)

Особливої уваги вимагає право на запис для каталогу. Створення і видалення файлів у каталозі вимагають зміни його вмісту, і, отже, права на запис у цей каталог. Найважливіше, що при цьому не враховуються права доступу для самого файлу. Тобто для того, щоб видалити деякий файл із каталогу, не обов'язково мати які-небудь права доступу до цього файлу, важливо лише мати право на запис для каталогу, у якому знаходиться цей файл. Майте на увазі, що право на запис у каталог дає великі повноваження, і надавати це право потрібно з обережністю. Правда, існує спосіб трохи убезпечити себе у випадку, коли необхідно надати право на запис іншим користувачам, — установка прапора Sticky bit на каталог.

У табл. 1.2 приведені приклади деяких дій над файлами і мінімальні права доступу, необхідні для виконання цих операцій.

Таблиця 1.2. Приклади прав доступу

|  |  |  |  |
| --- | --- | --- | --- |
| Команда | Зміст дії | Мінімальні права доступу | |
|  |  |  | |
|  |  | для звичайного файлу | для каталогу, що містить файл |
| cd /u/andrei | Перейти в каталог /u/andrei | — | x |
| Is /u/andrei/\* .с | Вивести усі файли із суффиксом з цього каталогу | — | г |
| Is -s /u/andrei/\*.с | Вивести додаткову інформацію про ці файли (розмір) | —— | гх |
| cat report.txt | Вивести на екран вміст файлу report.txt | г | х |
| cat » report.txt | Додати дані у файл report.txt | W | х |
| runme.sh | Виконати програму runme | х | х |
| runme | Виконати скрипт командного інтерпретатора runme. Sh | гх | х |
| rm runme | Видалити файл runme у поточному каталозі | — | XW |

Отже, для виконання операції над файлом мають значення клас доступу, до якого належить користувач, і права доступу, установлені для цього класу. Оскільки для кожного класу встановлюються окремі права доступу, усього визначено 9 прав доступу, по 3 на кожен клас.

Операційна система робить перевірку прав доступу при створенні, відкритті (для читання або запису), запуску на виконання або видаленні файлу. При цьому виконуються наступні перевірки:

1. Якщо операція запитується суперкористувачем, доступ дозволяється. Ніяких додаткових перевірок не робиться. Це дозволяє адміністратору мати необмежений доступ до усієї файлової системи.

2. Якщо операція запитується власником файлу, то:

а) якщо необхідне право доступу визначене (наприклад, при операції читання файлу встановлене право на читання для власника-користувача даного файлу), доступ дозволяється,

б) в противному випадку доступ забороняється.

3. Якщо операція запитується користувачем, що є членом групи, що є власником файлу, то:

а) якщо необхідне право доступу визначене, доступ дозволяється,

б) в противному випадку доступ забороняється.

4. Якщо необхідне право доступу для інших користувачів (other) установлено, доступ дозволяється, у противному випадку доступ забороняється.

Система проводить перевірки в зазначеній послідовності. Наприклад, якщо користувач є власником файлу, то доступ визначається винятково з прав власника-користувача, права власники-групи не перевіряються, навіть якщо користувач є членом власника-групи. Щоб проілюструвати це, розглянемо наступне:

--—rw-r-- 2 andy group 65 Dec 22 19:13 filel

Навіть якщо користувач andy є членом групи group, він не зможе ні прочитати, ні змінити уміст файлу filel. У той же час всі інші члени цієї групи мають таку можливість. У даному випадку, власник файлу має найменші права доступу до нього. Зрозуміло, розглянута ситуація носить гіпотетичний характер, оскільки користувач andy у будь-який момент може змінити права доступу до даного файлу як для себе (власника), так і для групи, і всіх інших користувачів у системі.

Додаткові атрибути файлу

Ми розглянули основні атрибути, що керують доступом до файлу. Існує ще кілька атрибутів, що змінюють стандартне виконання різних операцій. Як і у випадку прав доступу, ці атрибути по-різному інтерпретуються для каталогів і інших типів файлів.

Додаткові атрибути також встановлюються утилітою chmod(l), але замість кодів 'г', 'w' чи 'х' використовуються коди з табл. 1.3. Наприклад, для установки атрибута SGID для файлу filel необхідно виконати команду

$ chmod g+s filel'.

У табл. 1.3 приведені додаткові атрибути для файлів, і показано, як вони інтерпретуються операційною системою.

|  |  |  |
| --- | --- | --- |
| Таблиця 1.3. Додаткові атрибути для звичайних файлів | | |
| Код | Назва | Значення |
| t | Sticky bit | Зберегти образ виконуваного файлу в пам'яті після завершення виконання |
| S | Set UID, SUID | Установити UID процесу при виконанні |
| S | SetGID, SGID | Установити GID процесу при виконанні |
| 1 | Блокування | Установити обов'язкове блокування файлу |

Установка атрибута Sticky bit (справжня назва — save text mode) рідко використовується в сучасних версіях FREEBSD для файлів. У ранніх версіях цей атрибут застосовувався з метою зменшити час завантаження програм, що часто запускаються, (наприклад, редактора або командного інтерпретатора). Після завершення виконання задачі її образ (тобто код і дані) залишалися в пам'яті, тому наступні запуски цієї програми займали значно менше часу.

Атрибути (чи прапори) SUID і SGID дозволяють змінити права користувача при запуску на виконання файлу, що має ці атрибути. При цьому привілеї будуть змінені (зазвичай розширені) лише на час виконання і тільки у відношенні цієї програми.

Як правило програма, що запускається, дістає права доступу до системних ресурсів на основі прав доступу користувача, що запустив програму. Установка прапорів SUID і SGID змінює це правило, призначаючи права доступу виходячи з прав доступу власника файлу. Таким чином, запущений виконуваний файл, яким володіє суперкористувач, дістає необмежені права доступу до системних ресурсів, незалежно від того, хто його запустив. При цьому установка SUID приведе до спадкування прав власника-користувача файлу, а установка SGID — власника-групи.

Як приклад використання цієї властивості розглянемо утиліту passwd(l), що дозволяє користувачу змінити свій пароль. Очевидно, що зміна пароля повинна привести до зміни зміни певних системних файлів (файлу пароля /etc/passwd чи /etc/shadow,або бази даних користувачів, якщо використовується додатковий захист системи). Зрозуміло, що надання права на запис у ці файли всім користувачам системи є аж ніяк не кращим рішенням. Установка SUID для програми passwd(l) (точніше, на файл /usr/bin/passwd — файл утиліти, що виконується, passwd(l)) дозволяє добірно дозволити це протиріччя. Оскільки власником файлу /usr/bin/passwd є суперкористувач (його ім'я в системі — root), то хто б ні запустив утиліту passwd(l) на виконання, під час роботи даної програми він тимчасово дістає права суперкористувача, тобто може робити запис у системні файли, захищені від інших користувачів.

$ Is -IPa /usr/bin/passwd

-r-sr-sr-x 3 root sys 15688 Oct 25 1995 /usr/bin/passwd\*

Зрозуміло, що вимоги по безпеці для такої програми повинні бути підвищені. Утиліта passwd(J) повинна робити зміну пароля тільки користувача, що запустив її, і не дозволяти ніякі інші операції (наприклад, виклик інших програм).

Блокування файлів дозволяє усунути можливість конфлікту, коли дві чи більш задачі одночасно працюють з тим самим файлом.

Однак повернемося до обговорення додаткових атрибутів для каталогів (табл. 1.4).

Таблиця 1.4. Додаткові атрибути для каталогів

|  |  |  |
| --- | --- | --- |
| Код | Назва | Значення |
| t | Sticky bit | Дозволяє користувачу видаляти тільки файли, якими він чи володіє має права на запис |
| S | Set GID, SGID | Дозволяє змінити правило установки власника групи створюваних файлів, аналогічно реалізованому в BSD FreeBSD |

Під час обговорення прав доступу відзначалося, що надання права на запис у каталог дає досить великі повноваження. Маючи таке право, користувач може видалити з каталогу будь-який файл, навіть той, власником якого він не є й у відношенні якого не має ніяких прав. Установка атрибута Sticky bit для каталогу дозволяє установити додатковий захист файлів, що знаходяться в каталозі. З такого каталогу користувач може видалити тільки файли, якими він володіє, чи на який він має явне право доступу на запис, навіть при наявності права на запис у каталог. Прикладом може служити каталог /tmp, що є відкритим на запис для всіх користувачів, але в який може виявитися небажаної можливість видалення користувачем чужих тимчасових файлів.

Атрибут SGID також має інше значення для каталогів. При установці цього атрибута для каталогу знову створені файли цього каталогу будуть успадковувати власника-групу по власнику-групі каталогу. У такий спосіб для FREEBSD версії System V вдається імітувати поводження систем версії BSD, для яких таке правило спадкування діє за замовчуванням.

Подивитися наявність додаткових атрибутів можна за допомогою докладного списку файлів:

$ Is –1

drwxrwxrwt -r-sr-sr-x 5 sys 3 root sys 367 sys 15688 Dec 19 Oct 25 20:29 /tmp 1995 /usr/bin/passwd

Таблиця 1.5. Операції зміни атрибутів файлу

|  |  |  |
| --- | --- | --- |
| Операція | Команда/системний виклик | Кому дозволено |
| Зміна прав доступу | chmod(1) | Власник |
| Зміна додаткового атрибута Sticky bit | chmod(1) | Суперкористувач |
| Зміна додаткового атрибута SGID | chmod(1) | власник, причому його GID також повинний збігатися з ідентифікатором групи файлу |

Користувачі системи

Зараз же наша задача — розібратися, яку інформацію зберігає система про користувача. Для цього розглянемо фрагмент файлу /etc/passwd:

root :x:0:l:0000-Adinin( 0000) : /: /bin/bash daemon:х:1:1:0000-Admin(0000):/:

bin:x:2:2:OOOO-Admin(OOOO):/usr/bin:

sys:x:3:3:0000-Admin(0000) :/:

adm:x:4:4:0000-Admin(0000):/var/adm:

lp:x:71:8:0000-lp(0000):/usr/spool/lp:

uucp:x:5:5:0000-uucp(0000):/usr/lib/uucp:

nobody:x:60001:60001:uid no body:/:

andy:x:206:101:Andrei Robachevsky:/home/andy:/bin/bash

Кожен рядок файлу є записом конкретного користувача і має наступний формат:

name-.passwd-encod:UID:GID:comments:home-dir:shell — всього сім полів (атрибутів), розділених двокрапками.

Розглянемо докладніше кожний з атрибутів:

|  |  |
| --- | --- |
| name | Реєстраційне ім'я користувача. Це ім'я користувач вводить у відповідь на запрошення системи login. Для невеликих систем ім'я користувача досить довільне. У великих системах, у яких зареєстровані сотні користувачів, вимоги унікальності змушують застосовувати визначені правила вибору імен. |
| Password-encod | Пароль користувача в закодованому виді. Алгоритми кодування відомі, але вони не дозволяють декодировать пароль. При вході в систему пароль, що набирається, кодується, і результат порівнюється з полем Password-encod. У випадку збігу користувачу дозволяється ввійти в систему.  Навіть у закодованому виді доступність пароля представляє деяку небез-пеку, тому часто пароль зберігають в окремому файлі.  Користувач, у даному полі якого є символ \* , ніколи не зможе потрапити в систему. Як правило такий пароль мають псевдокористувачі. |
| UID | Ідентифікатор користувача є внутрішнім представленням користувача в системі. Цей ідентифікатор успадковується задачами, що запускає користувач, і файлами, що він створює. По цьому ідентифікаторі система перевіряє користувальницькі права (наприклад, при запуску програми чи читанні файлу). Суперкористувач має UID == 0, що дає йому необмежені права в системі. |
| GID | Визначає ідентифікатор первинної групи користувача. Цей ідентифікатор відповідає ідентифікатору у файлі /etc/group, що містить ім'я групи і повний список користувачів, що є її членами. Приналежність користувача до групи визначає додаткові права в системі. Група визначає загальні для всіх членів права доступу і тим самим забезпечує можливість спільної роботи (наприклад, спільного використання файлів). |
| comments | Як правило, це повне "реальне" ім'я користувача. Це поле може містити додаткову інформацію, наприклад, телефон або адрес електронної пошти. Деякі програми (наприклад, flnger(l) і поштові системи) використовують це поле. |
| home-dir | Домашній каталог користувача. При вході в систему користувач виявляється в цьому каталозі. Як правило, користувач має обмежені права в інших частинах файлової системи, але домашній каталог і його підкаталоги визначають область файлової системи, де він є повноправним хазяїном. |
| shell | Ім'я програми, що FREEBSD використовує як командного інтерпретатора. При вході користувача в систему FREEBSD автоматично запустить зазначену програму. Звичайно це один зі стандартних командних інтерпретаторів /bin/sh (Bourne shell), /bin/csh (C shell) чи /bin/ksh (Kom shell), що дозволяють користувачу вводити команди і запускати задачі. В принципі, у цьому полі може бути зазначена будь-яка програма, наприклад, командний інтерпретатор з обмеженими функціями (restricted shell), клієнт системи керування базою даних чи навіть редактор. Важливо те, що, завершивши виконання цієї задачі, користувач автоматично вийде із системи. Деякі системи мають файл /etc/shells, що містить список програм, що можуть бути використані як командний інтерпретатор. |

Паролі

Наявність пароля дозволяє захистити ваші дані, а можливо (якщо суперкористувач) і всю систему в цілому. Уточнимо: наявність хорошого пароля, тому що невірно обраний пароль — серйозний недолік у безпеці системи.

Призначити або змінити пароль можна командою passwd(l). Звичайний користувач може змінити свій пароль, адміністратор може призначити пароль будь-якому користувачу.

Перед запуском програми passwd(l) варто тримати в голові загальне правило вибору пароля: пароль повинний добре запам'ятовуватися і бути важким для підбору.

Не рекомендується записувати пароль, його необхідно запам'ятати. Власне прізвище, кличка улюбленого собаки, рік і місяць народження, безумовно легкі для запам'ятовування, але такі паролі неважко підібрати. Багато систем пропонують пароль, згенерований самою системою. Передбачається, що він зовсім позбавлений якого-небудь змісту, тобто не містить імен, і звань і взагалі яких-небудь вимовних слів. Хоча система пропонує його у вигляді, зручному для запам'ятовування, це не завжди допомагає.

Багато систем вимагають, щоб пароль задовольняв наступним вимогам:

1. довжина пароля не повинна бути менше шести символів;
2. пароль повинний включати принаймні 2 алфавітних символи й одну цифру або спеціальний символ;
3. пароль повинний містити хоча б 3 символи, що не зустрічалися у вашому попередньому паролі.

Паролі відіграють значну роль у забезпеченні безпеки системи. Загальні рекомендації, адресовані насамперед адміністраторам, можна звести до наступного:

1. У системі не повинне існувати незахищених користувальницьких входів. Це відноситься як до користувальницьких входів без пароля, так і до входів користувачів, що залишили систему. Якщо користувач тривалий час не працює в системі, видалите його чи запис хоча б захистите його вхід символом '\*' у поле пароля.

2. Якщо система допускає, установіть мінімальну довжину пароля. У залежності від вимог безпеки в системі це число може варіюватися від 8 до 12.

3. Завжди змінюйте пароль у наступних випадках:

• якщо хто-небудь довідався ваш пароль.

• якщо користувач більше не працює у вашій системі, усі паролі, що він знав, повинний бути змінений.

• якщо міняється адміністратор системи, повинні бути змінені всі системні паролі.

• якщо у вас з'явилася підозра, що файл паролів був зчитаний по мережі, буде розумним перемінити всі паролі в системі.

4. Пароль адміністратора повинний періодично мінятися, незалежно від обставин.

5. Це може показатися дивним, але не варто змушувати користувачів змінювати паролі надто часто. Швидше за все, у цьому випадку користувач вибере не кращий пароль. Але змінювати паролі усе-таки випливає. Частота зміни залежить від ступеня приступності вашої системи (ізольована станція, сервер з мережним доступом, наявність мережних екранів).

Не применшуйте роль паролів у системі.

Стандартні користувачі і групи

Після установки FreeBSD звичайно вже містить кілька зареєстрованих користувачів. Перелічимо основні з них (у різних версіях системи UID цих користувачів можуть незначно відрізнятися):

|  |  |
| --- | --- |
| Ім'я | Користувач |
| Root | Суперкористувач, адміністратор системи, UID=0. Користувач з цим ім'ям має необмежені повноваження в системі. Для нього не перевіряються права доступу, і в такий спосіб він має всі "важелі" для керування системою. Для виконання більшості функцій адміністрування потрібно вхід саме з цим ім'ям. Слід зазначити, що root — це тільки ім'я. Насправді значення має UID. Любою користувач з UID=0 має повноваження суперкористувача |
| Adm | Псевдокористувач, що володіє файлами системи ведення журналів |
| Bin | Звичайно це власник усіх файлів, що виконуються, що є командами FreeBSD |
| Cron | Псевдокористувач, що володіє відповідними файлами, від імені якого виконуються процеси підсистеми запуску програм за розкладом |
| ip чи lpd | Псевдокористувач, від імені якого виконуються процеси системи друкування, що володіє відповідними файлами |
| News | Псевдокористувач, від імені якого виконуються процеси системи телеконференцій |
| Nobody | Псевдокористувач, використовуваний у роботі NFS |
| Uucp | Псевдокористувач підсистеми FreeBSD -to- FreeBSD copy (uucp), що дозволяє передавати поштові повідомлення і файли між FreeBSD -хостами |

Нова система також містить ряд передвстановлених груп. Оскільки групи, як правило, менш важливі, приведемо лише дві категорії:

|  |  |
| --- | --- |
| Ім'я | Група |
| root чи wheel | Адміністративна група, GID=0 |
| User чи users чи staff | Група, у яку за замовчуванням включаються всі звичайні користувачі FreeBSD |

Завдання на роботу

1. Ввійти в мережу під ім'ям root

2. Створити свій особистий об'єкт myname і домашню директорію

2.1 Створимо користувача myname

2.2 Створити домашні директорії

2.3 Переконатися в створенні директорій

2.4 Перевірити їх права доступу

3. Призначити користувачу myname і групі myname права, необхідні для зміни інформації

3.1 Забрати усі права доступу

3.2 Призначити права доступу, для того щоб користувач і група myname одержали право на можливість зміни інформації в об'єкті

4. Призначити користувачу myname і групі myname права, необхідні для того, щоб вони могли запускати файли, що виконуються

5. Призначити користувачу myname і групі myname права, необхідні для того, щоб вони могли проглядати директорію

6. Вийти з мережі

7. Ввійти в мережу під ім'ям myname

8.

8.1 Перевірити права доступу директорій folder і mydir

8.2 Переконатися, що в користувача myname є права тільки на директорію folder

9. Видалити всі створені об'єкти

9.1 Видалити користувача myname