Теоретичні відомості

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.

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. Однак у деяких випадках процес може змінити свої права за допомогою системних викликів ***setuid*** і ***setgid***, а іноді система змінює 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

Захист файлів від несанкціонованого доступу в ОС FreeBSD ґрунтується 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 - і-вузол.

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 (o) Для інших користувачів (крім суперкористувача)

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

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

Розглянемо, наприклад, права доступу до файлу a.out:

Таблиця 1.

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, 'o' — інші користувачі, 'a' — усі класи користувачів), права доступу ('r' — читання, 'w' — запис і 'x' — 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.

Таблиця 2.

Власник	Група	Інші
r w x	27 x	r - -
111	101	100
7	5	4

Таким чином, наведений 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 (r). Редагування файлу, тобто його зміна, передбачає наявність права на запис (w). Нарешті, для того щоб запустити деяку програму на виконання, потрібно мати відповідне право (x).

Для каталогів ці права мають інший зміст, а для символічних зв'язків вони взагалі не використовується, оскільки контролюються цільовим файлом. 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. Права r і x діють незалежно, право x для каталогу не вимагає наявності права м, і навпаки.

Паролі

Наявність пароля дозволяє захистити ваші дані, а можливо (якщо суперкористувач) і всю систему в цілому. 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.