**110.1 Выполнение административных задач**

Студент должен уметь проверять систему на соответствие политикам безопасности.

**Изучаем**:

* поиск файлов с установленными suid/sgid битами;
* установку срока действия паролей;
* поиска открытых сетевых портов;
* установку ограничения потребления аппаратных ресурсов;
* определение пользователей, работающих в системе;
* настройку утилиты sudo.

Установка завышенных прав доступа к файлам (биты suid и sgid) может привести в серьезным нарушениям системы безопасности. Так, стороннее ПО или скрипты, запускаемые от имени суперпользователя могут собирать конфиденциальную информацию или менять настройки системы без ведома пользователя.

Для поиска всех файлов в системе, на которые установлен завышенные права можно воспользоваться командами:

***sudo find / -perm –u+s*** *(поиск по всей файловой системе файлов с установленным SUID);*

***sudo find / -perm –g+s*** *(поиск по всей файловой системе файлов с установленным SGID);*

***sudo find / -perm –u+s,g+s*** *(поиск файлов, на который установлен и SUID и SGID);*

***sudo find / -perm /u+s,g+s*** *(поиск файлов, на который установлен SUID или SGID);*

\_\_

Частой задачей политики безопасности является установка срока действия учетной записи и параметров пароля. Для задания пароля учетной записи используется команда **passwd**, например:

***sudo passwd petr*** *(установить пароль на учетную запись petr);*

Для управления блокировкой учетной записью используется команда **usermod**, например:

***sudo usermod –L petr*** *(заблокировать учетную запись petr);*

***sudo usermod –U petr*** *(разблокировать учетную запись petr).*

Установка срока действия параметров учетной записи осуществляется при помощи команды **chage**, например:

***sudo chage –l petr*** *(вывести параметры учетной записи petr);*

***sudo chage petr*** *(задать параметры учетной записи petr).*

При помощи утилиты **chage** можно указать следующие значения:

* *дата последней смены пароля;*
* *срок действия пароля (актуальность);*
* *срок деактивации пароля (время работы с неактуальным паролем);*
* *срок действия учетной записи (вне зависимости от параметров пароля);*
* *минимальный срок действия пароля;*
* *максимальный срок действия пароля;*
* *количество дней с предупреждением перед деактивацией пароля.*

Каждый из указанных параметров можно изменить по отдельности, запуская утилиту **chage** с различными ключами.

Глобальные политики безопасности всех учетных записей системы можно указать в файле **/etc/login.defs**

\_\_

Зачастую политикой безопасности предусмотрена установка ограничений ресурсов на процессы, запускаемые пользователями или скриптами. Для этого используется команда **ulimit**, например:

***ulimit –a*** *(показать существующие ограничения);*

***ulimit –f 0*** *(установить ограничение на размер создаваемого файла в 0 блоков);*

При этом установленные ограничения справедливы действуют в рамках открытой консоли, или скрипта, запущенного с соответствующими параметрами. Для установки правил ограничений используется файл **/etc/security/limits.conf**.

Каждая строчка этого файла представляет собой правило вида «кто, вид ограничения, параметр, значение», например:

***\* hard maxlogins 1*** *(для всех пользователей установить жесткое ограничение на параметр «количество одновременных сеансов» в размере «1»)*

\_\_

В традиционной ОС Linux существует один суперпользователь **root** с максимальными правами доступа к системе. Часто вход в систему от имени root запрещен, поэтому для работы от его имени используются утилиты, временно дающие максимальные привилегии.

Для того чтобы начать работу от имени root можно выполнить команду **su –** (будет запрошен пароль root и вызвана оболочка регистрации в системе. Домашним каталогом станет домашняя папка root).

Для завершения сеанса root необходимо выполнить команду **exit**.

Для того чтобы выполнить с правами root одну команду можно воспользоваться командой **su** с ключом **–c**, например:

***su –c ‘vi /etc/resolv.conf’*** *(запустить текстовый редактор vi с правами root);*

В большинстве современных дистрибутивов Linux используется утилита **sudo**, вызов которой запрашивает пароль текущего пользователя и дает разовые полномочия root, например:

***sudo vi /etc/resolv.conf*** *(запустить текстовый редактор vi с правами root);*

Для установки разрешений повышения прав до root при помощи утилиты sudo используется файл ***/etc/sudoers***, редактирование которого необходимо осуществлять при помощи утилиты **visudo**.

Для создания разрешения доступа к sudo можно создать в файле ***/etc/sudoers*** правило вида «кто откуда=(от имени пользователя:от имени группы) команда», например:

***semaev ALL=(ALL) ALL*** *(пользователю semaev разрешено из любого расположения выполнять любые команды от имени любого пользователя).*

При этом в указанном файле уже созданы необходимые правила с максимальными привилегиями для группы **admin** (в Debian) и группы **wheel** (в RedHat). То есть для разрешения доступа пользователя к sudo достаточно добавить его в эти группы.

\_\_

Частой административной задачей является проверка системы на наличие открытых сетевых портов, что повышает вероятность сетевой атаки и уязвимость системы в целом.

Для проверки запущенных сетевых служб и открытых портов можно воспользоваться командой **netstat**. Например:

***netstat –tuna*** *(показать в цифровом формате открытые TCP и UDP порты);*

Для сканирования хоста на наличие открытых портов можно воспользоваться командой **nmap**, например:

***nmap 192.168.1.50*** *(просканировать адрес 192.168.1.50 на наличие открытых портов);*

Для поиска портов занятых конкретным демоном можно воспользоваться командой **lsof**, например:

***sudo lsof –I*** *(показать текущие сетевые соединения);*

***sudo lsof –c:dnsmasq*** *(какие порты использует демон dnsmasq);*

Для поиска демонов, использующих конкретный порт можно воспользоваться командой **fuser**, например:

***sudo fuser 53/tcp*** *(показать демоны, использующие порт 53).*

\_\_

Иногда бывает нужно определить пользователей, работающих в системе в данный момент, или просмотреть историю сеансов.

Команда **w** показывает активных в данный момент пользователей и их процессы, например:

***w*** *(вывести информацию об активных сеансах всех пользователей);*

***w petr*** *(вывести информацию об активных сеансах пользователя* ***petr****).*

Команда **who** показывает информацию о текущих сеансах активных пользователей, например:

***who –b*** *(время последней загрузки);*

***who –qH*** *(все активные сеансы пользователей).*

Также для работы с бинарными файлами, хранящими сведения о сеансах пользователей, доступны команды:

**users** *(вывод пользователей, работающих в системе из файла /var/run/utmp*);

**last** *(вывод истории пользователей, входивших в систему из файла /var/run/wtmp);*

**lastb** *(вывод истории неудачных попыток входа /var/run/btmp).*