Система обнаружения атак LIDS

Бутько Алексей
Министерство обороны

Введение

- В традиционной модели Linux могущество рута непоколебимо. root обходит запреты файловой системы, может выполнять любые операции. На критических системах такое могущество может быть опасным.

Введение

- Даже если администратор полностью отдает себе отчёт в своих действиях, это ещё не гарантирует безопасность системы: например процесс, запущенный от его имени может иметь уязвимость. А уж если злоумышленник каким-либо способом добудет пароль рута – сделать что-либо будет уже невозможно.

Введение

- LIDS представляет собой патч ядра, который позволяет ограничить могущество рута. Процессам можно назначать разрешенные действия, а доступ к файлам ограничить даже руту.

Функции

- LIDS имеет несколько функций. Первое – встроенный детектор сканирования портов. Эта опция может быть выбрана на этапе сборки ядра. Детектор может быть интегрирован с другими программами (такими как scanlogd), либо просто выдавать информацию через syslog.

Функци

- Важнейшая функция LIDS – это принудительные ограничения доступа (ACL). Они делятся на две категории: действия, выполняемые над файлами (read/write/append) и действия, которые может выполнить процесс

Конфигурация

- После установки патча LIDS и пакета lidsadm создаются файлы:
- /etc/lids/lids.cap файл для назначения возможностей
- /etc/lids/lids.confLIDS файл для хранения ACL
- /etc/lids/lids.pwLIDS файл для хранения пароля
- /etc/lids/lids.netLIDS файл сетевой конфигурации LIDS
- Каталог /etc/lids всегда закрыт для пользователей, даже для рута и может быть отредактирован только администратором LIDS.

«Опечатывание» ядра

- Обычно модули ядра могут подгружаться динамически в процессе работы. Этой возможностью часто пользуются руткиты и, поскольку весь процесс происходит в пространстве ядра, их достаточно трудно обнаружить. LIDS может позволить подгрузку модулей только в процессе запуска системы до момента выполнения команды lidsadm -I. Как только ядро опечатоно никакие модули подгружены быть не могут.

«Опечатывание» ядра

- Команду lidsadm -I обычно помещают в стартовые скрипты на момент отработки всех остальных скриптов. Например, если машина загружается на уровень выполнения 2, можно поместить нечто подобное в /etc/rc2.d/S99sealkernel

```
#!/bin/sh
    case "$1" in
        start)  /sbin/lidsadm -I ;;
        stop)   ;;
        *)      echo "Usage: $0 start" >&2; exit 1 ;;
    esac
    exit 0;
```

LFS

- LFS в данном случае означает LIDS Free Session. Файлы конфигурации LIDS недоступны другим пользователям и процессам. Соответственно для конфигурирования LIDS необходима сессия свободная от этих запретов. Такая сессия вызывается командой lidsadm -S --LIDS. Она запрашивает пароль администратора LIDS и, если он введен корректно, запускает LFS.

Конфигурирование LIDS с помощью lidsadm

- Как только ядро с LIDS загружено единственным способом повлиять на конфигурацию LIDS остается программа lidsadm. Основными ее аргументами являются:
- lidsadm -Z удаление всех ACL
- lidsadm -L вывод текущих ACL
- lidsadm -V вывод текущего набора возможностей
- lidsadm -U обновление таблицы иннодов в /etc/lids/lids.conf

Конфигурирование LIDS с помощью lidsadm

- lidsadm -I опечатывание ядра
- lidsadm -A ... добавление новых ACL
- lidsadm -D ... удаление указанных ACL
- lidsadm -S -- -LIDS запуск LFS
- lidsadm -S -- -LIDS_GLOBAL полное отключение LIDS
- lidsadm -S -- +LIDS_GLOBAL включение LIDS
- lidsadm -S -- +RELOAD_CONF перезагрузка конфигурации
- Эти команды должны выполняться в LFS

Lids.conf

- Все созданные ACL хранятся в файле Lids.conf.Записи в нем модифицируются командами lidsadm -A и lidsadm -D. Править файл вручную нельзя!
- Можно в целях автоматизации организовать добавление ACL скриптом. В его начало следует пометить команду lidsadm -Z во избежании дублирования. После редактирования списка ACL необходимо выполнить команду lidsadm -S -- +RELOAD_CONF чтобы изменения вступили в силу.

Просмотр списка ACL.

- Как было сказано выше, все ACL хранятся в файле /etc/lids/lids.conf. Однако его просмотр даст мало информации, так как все защищаемые объекты хранятся там в виде иннодов. Чтобы просмотреть список существующих ACL достаточно выполнить команду lidsadm -L. Ее выводом будет список файловых и возможностных ACL в том порядке как они записаны в lids.conf.

Перезагрузка конфигурации

- Загрузка конфигурации из /etc/lids производится только на этапе старта системы. Добавление или удаление ACL с помощью lidsadm -A и lidsadm -D влияет только на файл /etc/lids/lids.conf. Чтобы внесенные изменения начали действовать необходимо выполнить команду lidsadm -S --+RELOAD_CONF. Эта операция будет доступна только если на этапе компиляции были выбраны опции «Allow switching LIDS protections» и «Allow reloading config file». В противном случае потребуется перезагрузка системы.

Lids.net

- LIDS включает возможность отсылки предупреждений по сети, если на этапе компиляции была выбрана опция «Send Security alerts through network». В файле /etc/lids/lids.net хранятся параметры smtp, которые позволяют отсылку предупреждений безопасности на e-mail.

Файловые ограничения

- В LINUX существует 2 вида защиты файлов. Первый способ – это стандартные user/group/other разрешения, назначаемые командой chmod. Второй способ – атрбуты файловой системы +a, +i, +s, +d.

Файловые ограничения

- LIDS использует свой метод контроля доступа, он итегрирован в виртуальную файловую систему на уровне ядра и не зависит от конкретной файловой системы. Это значит, что ACL будут действовать на любую, в том числе и удаленно смонтированную файловую систему.

Файловые ограничения

- ACL начинают действовать с момента, когда система стартует и прекращают только если LIDS будет отключен, что порой является причиной головной боли на машине с только что установленной LIDS.

Файловые ограничения

- Существуют 4 типа ACL в LIDS:
- DENY - полный запрет доступа
- READ - доступ только для чтения
- APPEND - доступ только для дописывания в конец файла
- WRITE - доступ для записи

Файловые ограничения

- ACL могут быть назначены, как отдельным файлам, так и директориям, на содержимое которых ACL будут действовать рекурсивно
- ACL добавляются так:
- lidsadm -A -s <субъект> -o <объект> -j <вид доступа>
- Например команда
- lidsadm -A -o /etc -j READ
- делает каталог /etc доступным только для чтения всем субъектам.

Файловые ограничения

- Команда
- lidsadm -A -o /etc/shadow -j DENY
  запрещает всем субъектам доступ к файл /etc/shadow Однко в данном случае для таких процессов как login или sshd следует доступ открыть:
- lidsadm -A -s /usr/sbin/sshd –o /etc/shadow -j READ
- Что разрешит процессу sshd доступ только для чтния

Защита блочных устройств

- Даже когда файловая система смонтирована и в ней назначены ACL, это не может уберечь от прямого чтения с блочного устройства. Это можно предотвратить путем установки возможности -17:CAP_SYS_RAWIO в lids.cap. Возможно, единственный субъект, которому можно позволить прямой доступ к устройствам — это графический сервер (если он используется). Сделать это можно так:
- ```
  lidsadm -A -s /path/to/X_server -o CAP_SYS_RAWIO  -j GRANT
  ```
  ```

Индексные дескрипторы

- Если просмотреть файл lids.conf, то можно увидеть, что ACL связаны не с объектами, а с их инодами. Это имеет как положительные, так и отрицательные стороны. Например, если у файла умется несколько жестких ссылок, то все они защищены одним ACL. Однако это же означает, что при изменении инода ACL разрушаются.
- Иоды изменяются при удалении и создании файла с таким же именем. Так что имейте в виду, после подобного рода операций необходимо выполнить команду lidsadm –U, которая обновит ACL в lids.conf

LINUX возможности

- В своё время была сделана попытка создать стандартный метод категорирования всех проверок доступа в дискретный набор групп, называемых возможностями. Был разработан POSIX 1003.1e где были стандартизованы все возможности *nix-платформ, однако черновик был отклонен и стандартом так и не стал.

LINUX возможности

- LINUX включает в себя один из вариантов этой модели. Вместо проверки UID==0 в ядре используется вызов capable(). Он определяет возможности назначенные процессу. Полный список возможностей можно посмотреть в /usr/include/linux-x.x.xx/capability.h

LINUX возможности

- В стандартной Linux-модели дело ограничивается назначением возможностей процессам, которые имеют uid/euid равный 0. Остальные процессы возможностей не имеют и с ними ядро разбирается по стандартной логике.

LINUX возможности

- LIDS дополняет набор возможностей. Возможность можно удалять так что даже процессы суперпользвателя не смогут их использовать, к тому же возможности не удаляются перманентно и их можно подключить опять если нужно. Также возможности можно будет добавлять любым процессам на уровне от программы к программе

Lids.cap

- Этот файл содержит набор возможностей для ядра. Они записываются в виде
- <+/->:<возможность>
- + означает, что возможность будет оставлена в наборе, - означает, что возможность не будет доступна ни одному из прцессов.

Lids.cap

- Возможности не будут действовать при загрузке машины. Это сделано для того, чтобы программы из /etc/rcX.d могли отработать без каких-либо ограничений. Возможности начинают действовать только после «опечатывания ядра» командой lidsadm –I
- Просмотреть список возможностей можно выполнив команду lidsadm –V Однако опция –V будет доступна только если lidsadm был собран с параметром VIEW=1

Назначение исключений в возможностях

- В традиционной linux-модели возможности, удаленные из набора, более недоступны никаким процессам. LIDS дает возможность назначать возможности каждому процессу.

Назначение исключений в возможностях

- Делается это так
- `lidsadm -A -s <процесс> -o <возможность>  -j GRANT`
- Например:
- `lidsadm -A -s /usr/sbin/ntpdate -o CAP_SYS_TIME -j GRANT`
- разрешает процессу ntpdate менять системное время.

Специальные возможности

- Обычно когда программа имеет возможность CAP_BIND_NET_SERVICE она может закрепится за любым портом младше 1024. LIDS дает возможность определить конкретный порт. Например 80 для httpd
- ```
  lidsadm -A -s /usr/sbin/httpd -o
  CAP_BIND_NET_SERVICE 80-80 -j GRANT
  ```
  ```

Специальные возможности

- Возможность CAP_INIT_KILL защищает демоны от приема сигналов. Если её отключить никакой пользователь или процесс не сможет сигнализировать демону.

Специальные возможности

- Возможность CAP_HIDDEN позволяет прятать процессы. Процесс, неделенный такой возможностью не будет виден в /proc Это однако не 100% гарантия: процесс sshd например можно увидеть через 22 порт. Также можно азослать сигналы по всем PID с 1 по 65535 и, если будет получен отклик, а в /proc соответствующего PID не будет – значит процесс существует. Также определённые выводы можно сделать по содержимому /var/log/NAME.pid