

Защита хоста



Алексей
Федин




Алексей Федин

**Ведущий инженер
по информационной безопасности**

План занятия

1. [Информационная безопасность](#)
2. [Защита хоста](#)
3. [SELinux и AppArmor](#)
4. [РАМ](#)
5. [Шифрование](#)
6. [Итоги](#)
7. [Домашнее задание](#)



Информационная безопасность

ИБ: определения

Информационная безопасность (Information Security, InfoSec) — методы защиты информации путём уменьшения информационных рисков.

Кибербезопасность (Computer security, cybersecurity) — методы защиты компьютерных систем, их программного и аппаратного обеспечения, хранимых и передаваемых данных от компьютерных атак

В РФ термин **кибербезопасность** не используется на официальном уровне

ИБ: определения

Субъект — пытается получить **доступ** к **объекту**

Объект — то, к чему ограничивается или отслеживается **доступ**

Доступ — операция чтения, записи, создания, удаления, модификации и т. д.

ИБ: свойства информации

Конфиденциальность — состояние информации, при котором доступ к ней возможен только со стороны авторизованных на это субъектов.

Доступность — возможность беспрепятственного доступа к информации для субъекта, имеющего необходимые права.

Целостность — отсутствие неправомерного искажения, добавления, удаления информации.

Эти термины относятся к **бумажной безопасности** и больше нужны **безопасникам**

ИБ: угроза безопасности информации

Угроза безопасности информации — совокупность условий и факторов, создающих потенциальную или реально существующую опасность нарушения информационной безопасности (ГОСТ Р 50922-96).

Угроза = вероятность + опасность

Наличие угрозы не подразумевает атаку на систему, но говорит о её вероятности.

Угроза ➡ (вероятность) ➡ ущерб

ИБ: источник угрозы безопасности информации

Источник угрозы безопасности информации — субъект, являющийся непосредственной причиной возникновения угрозы безопасности информации: авторизованный пользователь, злоумышленник, насекомое.

Уязвимость информационной системы — свойство системы, обуславливающее возможность реализации угроз безопасности информации



Защита хоста

Защита хоста: антивирусы

Антивирус — средство защиты, предназначенное для определения и блокирования вредоносного ПО: вирусов, троянов, червей и т. д.

Основные подходы к обнаружению вредоносного ПО:

- сигнатурный — проверка по БД индикаторов
- поиск аномалий — анализ поведения

Варианты установки:

- на хосте
- на сервере, обычно подключаются как отдельные модули

Защита хоста: HIDS

Хостовая система обнаружения вторжений (Host-based intrusion detection system, HIDS) — средство защиты, предназначенное для анализа событий в динамике, происходящих на хосте.

Основные подходы к обнаружению вредоносного ПО:

- наблюдение за участками памяти
- наблюдение за выбранными файлами
- наблюдение за поведением системы
- наблюдение за логами

Защита хоста: песочница

Песочница (sandbox) — тестовая среда для запуска ПО, изолированная от основной системы.

Дистрибутивы:

- seccomp (средство ядра Linux, ограничивает системные вызовы)
- Sandboxie
- Shade sandbox



SELinux и AppArmor

Защита хоста: SELinux

SELinux — это система принудительного контроля доступа, реализованная на уровне ядра.

SELinux применяет систему **дополнительной маркировки** объектов ОС и определяет правила доступа каждого процесса к объектам на основе присвоенных меток.

Такие правила объединяются и называются **политикой**

Защита хоста: SELinux

Режимы работы:

- Disabled
- Enforcing — блокировка в соответствии с заданными политикам
- Permissive — логирование в соответствии с заданными политикам.



AppArmor: установка

AppArmor — модуль ядра Linux, обеспечивающий управление доступом на основе имён.

Т. е. для каждой программы создаётся свой профиль, в котором указываются уровни доступа к каталогам, программам и системным ресурсам.

```
user@user:~$ sudo apt install apparmor-profiles apparmor-utils  
apparmor-profiles-extra
```

Ссылка: gitlab.com/apparmor/apparmor

AppArmor: статyc

user@user:~\$ sudo apparmor_status

```
user@user-VirtualBox:~$ sudo apparmor_status
[sudo] password for user:
apparmor module is loaded.
77 profiles are loaded.
38 profiles are in enforce mode.
/sbin/dhclient
/snap/core/10185/usr/lib/snapd/snap-confine
/snap/core/10185/usr/lib/snapd/snap-confine//mount-namespace-capture-helper
/usr/bin/evince
/usr/bin/evince-previewer
/usr/bin/evince-previewer//sanitized_helper
/usr/bin/evince-thumbnailer
/usr/bin/evince//sanitized_helper
/usr/bin/man
/usr/lib/NetworkManager/nm-dhcp-client.action
/usr/lib/NetworkManager/nm-dhcp-helper
/usr/lib/chromium-browser/chromium-browser//browser_java
/usr/lib/chromium-browser/chromium-browser//browser_openjdk
/usr/lib/chromium-browser/chromium-browser//sanitized_helper
/usr/lib/connman/scripts/dhclient-script
/usr/lib/cups/backend/cups-pdf
/usr/lib/snapd/snap-confine
/usr/lib/snapd/snap-confine//mount-namespace-capture-helper
/usr/sbin/cups-browsed
```



AppArmor: режимы работы

Режим обучения (сообщения о действиях приложения):

```
user@user:~$ sudo aa-complain <путь к файлу>
```

Режим ограничения согласно профилю:

```
user@user:~$ sudo aa-enforce <путь к файлу>
```

AppArmor: режимы работы

Рассмотрим подробнее вывод `sudo apparmor_status`:

```
5 processes have profiles defined.  
3 processes are in enforce mode.  
  /sbin/dhclient (948)  
  /usr/sbin/cups-browsed (876)  
  /usr/sbin/cupsd (828)  
2 processes are in complain mode.  
  /usr/sbin/avahi-daemon (789)  
  /usr/sbin/avahi-daemon (798)  
0 processes are unconfined but have a profile defined.
```

Здесь нам показано, сколько процессов имеют профили и в каком режиме защиты они сейчас запущены

AppArmor: пример работы

```
user@user:~$ ls /etc/apparmor.d/
```

```
user@user:~$ sudo cp /usr/bin/man /usr/bin/man1
```

```
user@user:~$ sudo cp /bin/ping /usr/bin/man
```

```
user@user:~$ sudo man 127.0.0.1
```

```
PING 127.0.0.1 (127.0.0.1) 56(84) bytes of data.
```

```
64 bytes from 127.0.0.1: icmp_seq=1 ttl=64 time=0.126 ms
```

```
user@user:~$ sudo aa-enforce man
```

```
user@user:~$ sudo man 127.0.0.1
```

```
ping: socket: Permission denied
```

AppArmor: файл приложения

nano /etc/apparmor.d/bin.ping

```
user@user-VirtualBox:~$ cat /etc/apparmor.d/bin.ping
# -----
#
# Copyright (C) 2002-2009 Novell/SUSE
# Copyright (C) 2010 Canonical Ltd.
#
# This program is free software; you can redistribute it and/or
# modify it under the terms of version 2 of the GNU General Public
# License published by the Free Software Foundation.
# -----
#include <tunables/global>
profile ping [{usr/,}bin/ping flags=(complain) {
  #include <abstractions/base>
  #include <abstractions/consoles>
  #include <abstractions/nameservice>

  capability net_raw,
  capability setuid,
  network inet raw,
  network inet6 raw,

  [{,usr/}bin/ping mixr,
  /etc/modules.conf r,

  # Site-specific additions and overrides. See local/README for details
  #include <local/bin.ping>
}
```



AppArmor: отключение

Остановка службы:

```
user@user:~$ sudo service apparmor stop
```

Выгрузка профилей:

```
user@user:~$ sudo service apparmor teardown
```



PAM

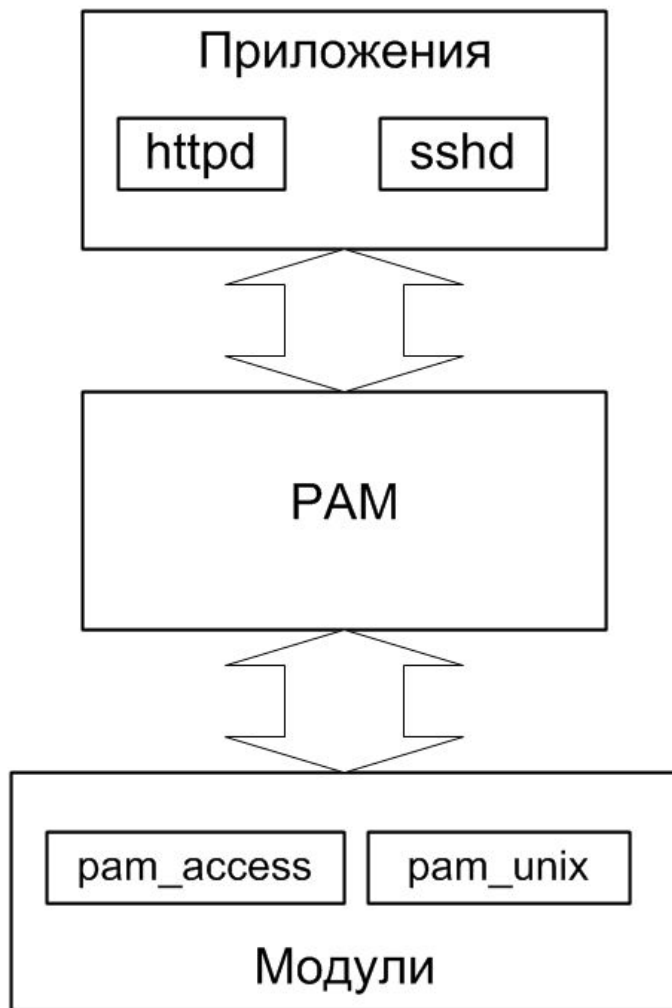
PAM

PAM (Pluggable Authentication Modules, подключаемые модули аутентификации) — набор библиотек, с помощью которых можно настроить методы аутентификации пользователей.

Типы модулей PAM:

- модули учётных записей
- модули аутентификации
- модули паролей
- модули сессий

PAM



Ограничение попыток входа

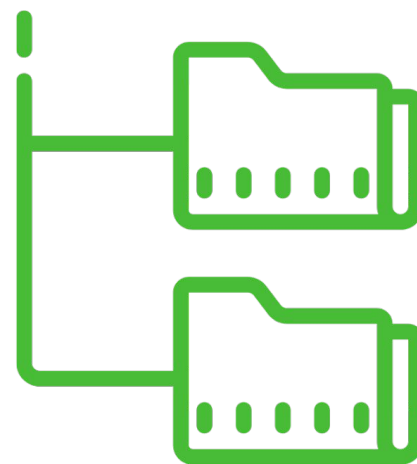
Структура каталогов PAM:

`/etc/pam.d/` — файлы конфигураций приложений

`/lib/security/` — модули PAM

`/etc/security/` — файлы конфигураций для PAM-окружений

`/usr/share/doc/pam-*/` — документация



Ограничение попыток входа

user@user:~\$ sudo nano /etc/pam.d/common-auth

```
GNU nano 4.8 /etc/pam.d/common-auth

# here are the per-package modules (the "Primary" block)
auth [success=1 default=ignore] pam_unix.so nullok_secure
# here's the fallback if no module succeeds

#auth requisite pam_deny.so
auth required pam_deny.so
# prime the stack with a positive return value if there isn't one already;
# this avoids us returning an error just because nothing sets a success code
# since the modules above will each just jump around

auth required pam_tally2.so onerr=fail deny=3 unlock_time=1500

auth required pam_permit.so
# and here are more per-package modules (the "Additional" block)
auth optional pam_cap.so
# end of pam-auth-update config
```



Шифрование

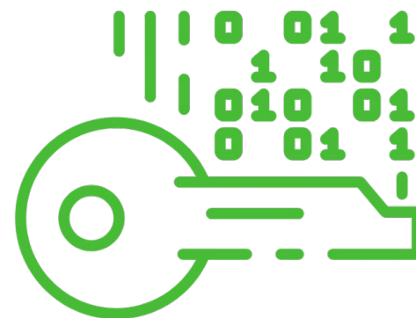
eCryptfs

eCryptfs — это POSIX-совместимая файловая система вложенного (stacked) шифрования.

eCryptfs защищает файлы для любой файловой системы, раздела и т. д.

Установка:

```
user@user:~$ sudo apt install ecryptfs-utils
```



Шифрование домашнего каталога

Создание нового пользователя:

```
user@user:~$ sudo adduser --encrypt-home user2
```

Проверка шифрования:

```
user@user:~$ su - user2
```

```
user2@user:~$ touch 123,456
```

```
user2@user:~$ exit
```

```
user@user:~$ sudo ls /home/user2/
```

```
Access-Your-Private-Data.desktop README.txt
```

Шифрование каталогов

Миграция домашнего каталога пользователя:

```
user@user:~$ sudo ecryptfs-migrate-home -u user1
```

Шифрование раздела swap:

```
user@user:~$ sudo ecryptfs-setup-swap
```

Информация для восстановления:

```
user@user:~$ ecryptfs-unwrap-passphrase
```

LUKS

LUKS (Linux Unified Key Setup) — спецификация формата шифрования дисков, используемая в ОС Linux.

При помощи LUKS могут быть зашифрованы диски, работающие в ОС Linux как в настольных компьютерах, так и в разнообразных устройствах, например, сетевых накопителях



Установка LUKS

Подготовка диска:

```
user@user:~$ sudo apt install gparted
```

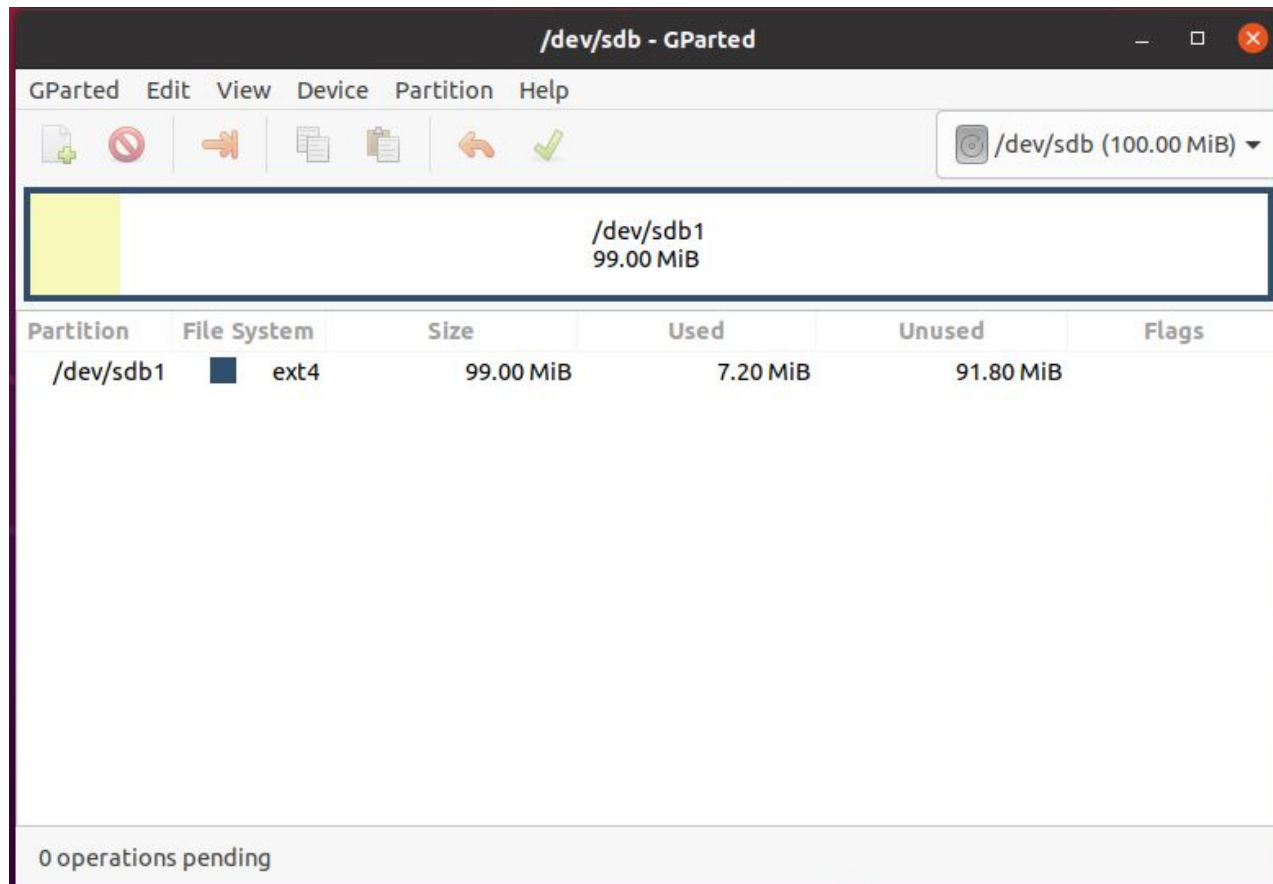
Установка LUKS (должна быть установлено по умолчанию):

```
user@user:~$ sudo apt-get install cryptsetup
```

Проверка установки:

```
user@user:~$ cryptsetup --version
```

LUKS: подготовка раздела



Шифрование раздела LUKS

Подготовка раздела (luksFormat):

```
user@user:~$ sudo cryptsetup -y -v --type luks2 luksFormat /dev/sdb1
```

Монтирование раздела:

```
user@user:~$ sudo cryptsetup luksOpen /dev/sdb1 disk
```

```
user@user:~$ ls /dev/mapper/disk
```

Форматирование раздела:

```
user@user:~$ sudo dd if=/dev/zero of=/dev/mapper/disk
```

```
user@user:~$ sudo mkfs.ext4 /dev/mapper/disk
```

Шифрование раздела LUKS

Монтирование «открытого» раздела:

```
user@user:~$ mkdir .secret
```

```
user@user:~$ sudo mount /dev/mapper/disk .secret/
```

Завершение работы:

```
user@user:~$ sudo umount .secret
```

```
user@user:~$ sudo cryptsetup luksClose disk
```



Итоги

Итоги

Сегодня мы:

- получили представление о защите рабочей станции от атак
- познакомились с дополнительными системами защиты:
SELinux, AppArmor, PAM
- настроили шифрование каталогов и разделов с помощью
eCryptfs и LUKS

Домашнее задание

Давайте посмотрим ваше [домашнее задание](#).

- Вопросы по домашней работе задавайте **в чате** мессенджера
- Задачи можно сдавать **по частям**
- Зачёт по домашней работе проставляется после того, как **приняты все задачи**

**Задавайте вопросы и
пишите отзыв о лекции!**

Алексей Федин