Реализация механизма автоматического выбора сертификата открытого ключа на основании контекста безопасности

Воронин Д.Л., Муравьёв С.К.

8 апреля 2014

Задачи

- Исследовать основные принципы работы системы SELinux
- Разработать способ создания сертификатов с контекстом безопасности
- Разработать средства автоматического выбора сертификата

SELinux

SELinux — мандатная система контроля доступа.

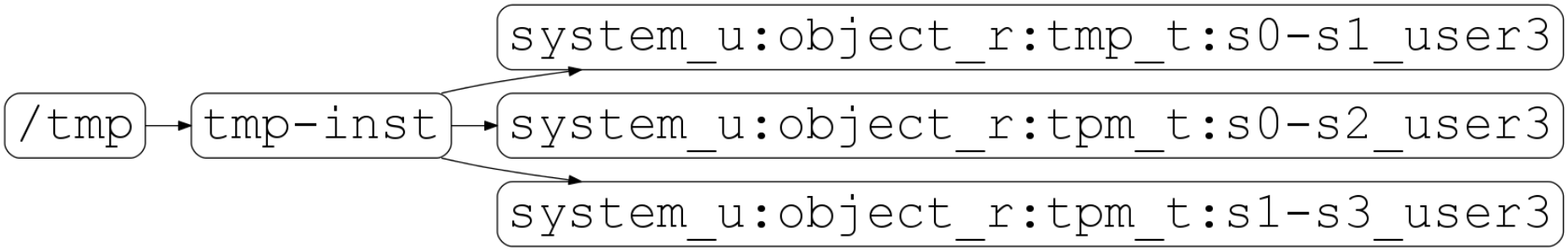
- Режим работы: disabled, permissive, enabled
- Тип политик: target, mls
- Вид контекста безопасности:

```
user:role:type:sensitivity:category
```

Пример контекста безопасности:

```
user_u:user_r:user_t:s0-s3:c0.c10
```

Многоэкземплярность директорий



Реализация: модуль pam_namespace.so
Скрипт инициализации namespace.init
Конфигурационный файл: namespace.conf

Реализована:

- возможность передачи текущего контекста пользователя в скрипт инициализации

OpenSSL

OpenSSL — Криптографичекский пакет для работы с сертификатами.

Дополнения в сертификатах:

- Модификация конфигурационного файла openssl.conf
- Программно:
  - alias на существующее дополнение
  - реализация структуры дополнения

Реализовано:

- Дополнение v3_secon

Разработка утилиты создания сертификатов

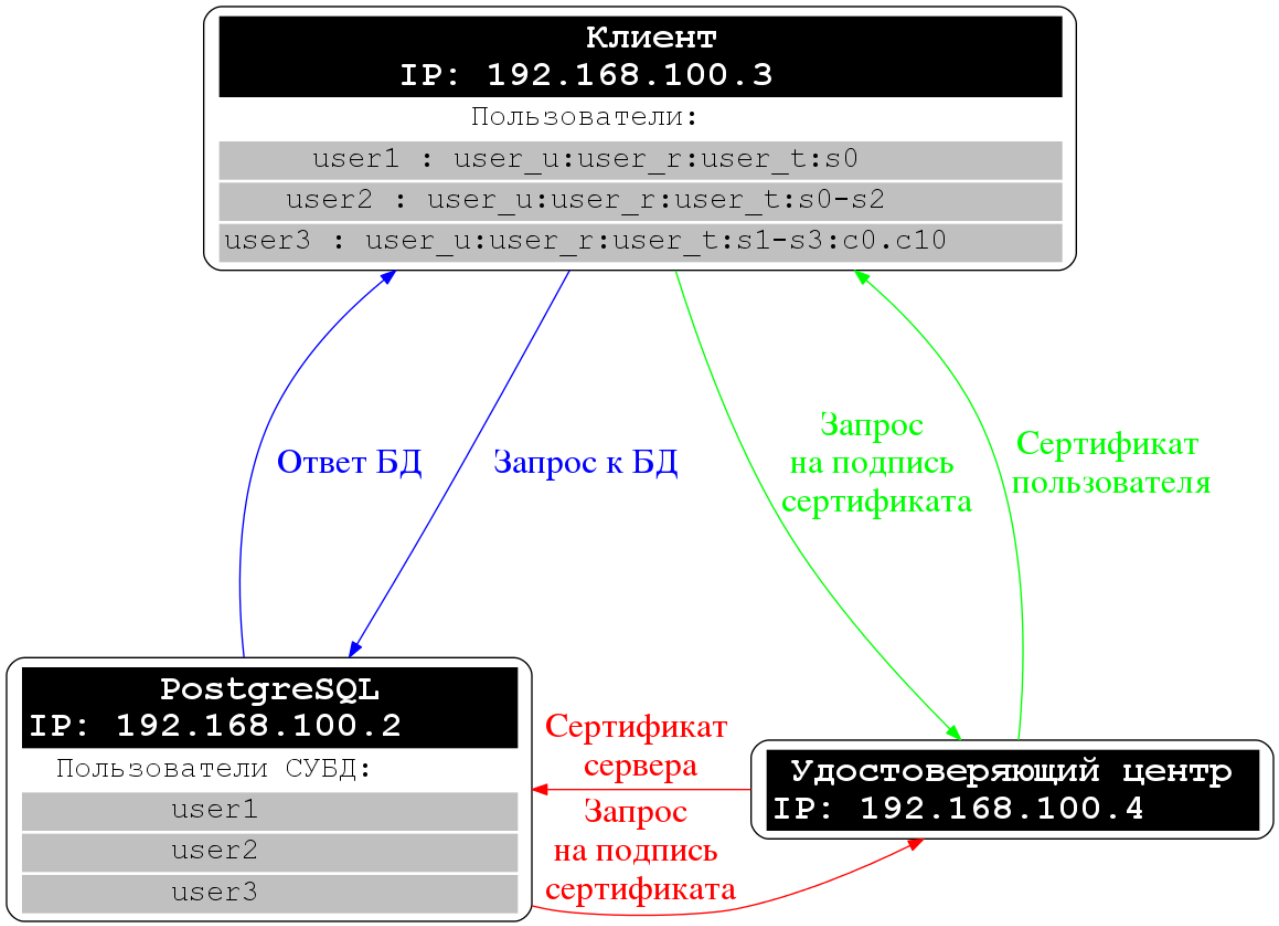
Требования:

1. Возможность создавать закрытый ключ клиента произвольной длины
2. Создавать запросы на подпись сертификата с дополнением selinuxContext
3. Подписывать запрос удостоверяющим центром

Реализовано:

- утилита на языке Python pgcert

Схема стенда



Тестирование работы механизма

```
                    7c:ff
                Exponent: 65537 (0x10001)
        X509v3 extensions:
            Selinux Context:
                user_u:user_r:user_t:s0-s2
            X509v3 Basic Constraints: critical
                CA:FALSE
    Signature Algorithm: sha1WithRSAEncryption
         ce:c1:48:aa:9b:a3:f5:33:0e:3c:03:8e:c4:95:31:39:f7:dc:
         11:ad:fe:e2:95:70:3a:5a:96:cd:a1:32:e6:cf:4f:b9:0b:1b:
         81:26:19:0b:7e:71:ae:0e:e4:3e:a2:34:31:1a:fa:a3:76:66:
         55:92:e5:0b:1b:72:de:7e:70:ee:1c:38:ad:7a:42:6f:85:fd:
         42:02:d0:24:dc:28:d2:fd:30:b6:0d:72:31:fa:85:a8:a1:dd:
         eb:68:b3:a2:68:4c:56:79:a4:a7:3a:d7:2f:68:44:3d:c8:b7:
         a5:2d:0b:f5:ef:75:9d:28:36:1b:aa:64:87:cd:b9:7f:da:c8:
         45:97:70:12:39:3f:00:b0:99:7d:45:4b:33:7a:80:8d:60:06:
         26:a9:b9:1e:2f:f0:c2:c8:e3:ec:d3:56:fd:b2:60:60:89:15:
         6d:6d:ed:f5:57:a5:96:82:26:db:e2:06:29:cb:65:61:fc:e0:
         64:21:f2:07:4d:f2:4f:2d:79:86:f5:4f:cc:48:74:29:df:0b:
         c7:7e:1a:01:24:1f:87:71:fa:2c:41:53:5d:33:42:fc:3d:5c:
         c2:e2:45:61:05:6f:a1:00:f0:53:aa:f7:e2:68:9b:65:f5:de:
         9a:29:9e:61:fd:da:04:bc:7c:d0:73:af:35:58:d6:45:be:11:
         6f:75:39:a6

[user2@pgsslclient ~]$ _
```

Тестирование работы механизма

```
     ¦ обычная
 public ¦ ssl_client_serial          ¦ numeric                ¦
     ¦ обычная
 public ¦ ssl_get_extension_by_name ¦ text                   ¦ text
     ¦ обычная
 public ¦ ssl_get_extensions_count   ¦ text                   ¦
     ¦ обычная
 public ¦ ssl_is_critical_extension ¦ text                   ¦ text
     ¦ обычная
 public ¦ ssl_is_used                ¦ boolean                ¦
     ¦ обычная
 public ¦ ssl_issuer_dn              ¦ text                   ¦
testdb=> select ssl_get_extension_by_name('selinuxContext');
 ssl_get_extension_by_name
---------------------------
 user_u:user_r:user_t:s0-s2
(1 строка)

testdb=> select sepgsql_getcon();
       sepgsql_getcon
---------------------------
 user_u:user_r:user_t:s0-s2
(1 строка)

testdb=> _
```

Заключение

- Реализован механизм выбора сертификата открытого ключа, содержащего метку безопасности
- Расширены возможности pam_namespace, OpenSSL, M2Crypto, sslinfo
- Показано применение разработанного механизма в модуле sepgsql

Спасибо за внимание!