Настройка stunnel и немного магии

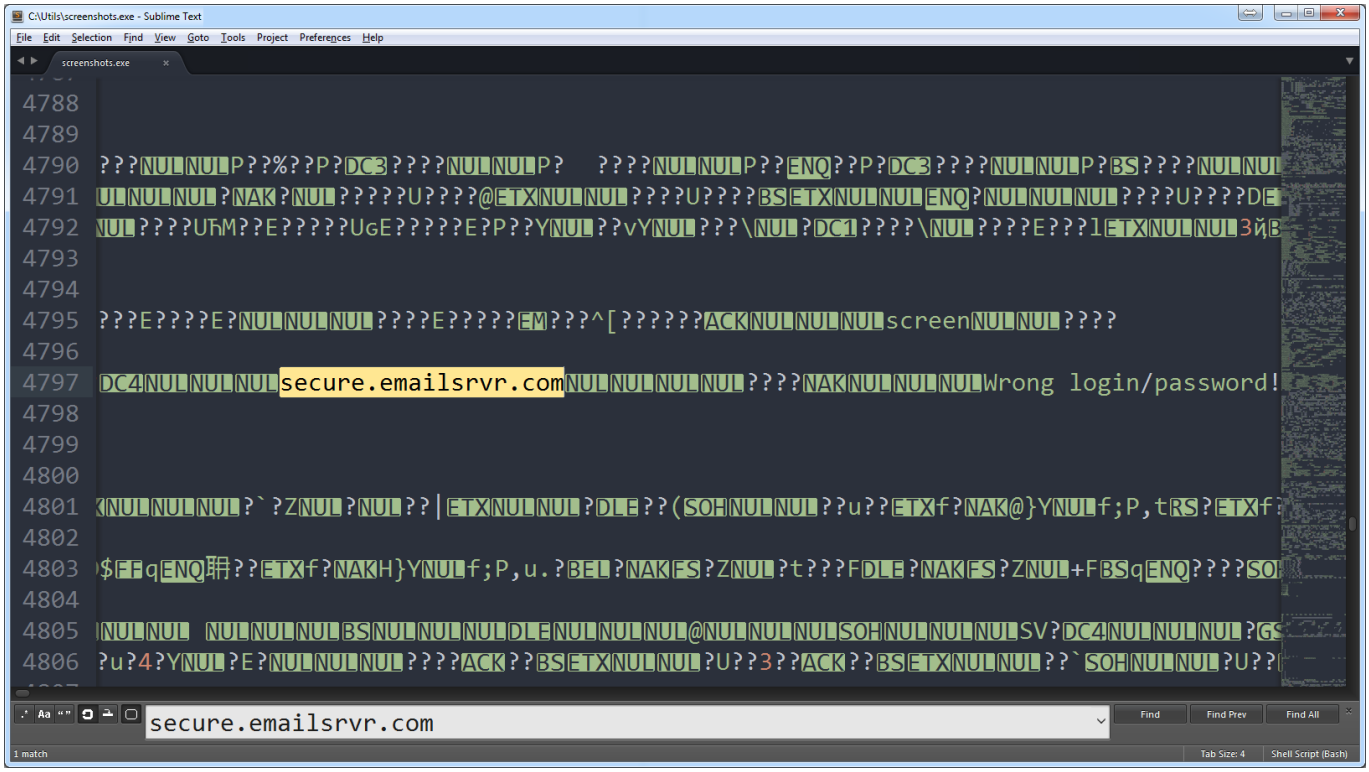
Введение

На днях столкнулся с интересной задачей. В компании несколько лет назад для создания скриншотов была написана программа, написана на делфи, программа очень удобная, но есть ряд минусов - работает только под windows и аутентификация в программе происходит путем подключения к почтовому серверу компании и проверки корректности данных через pop3 протокол. Причем имя почтового сервера жестко вшито в сам exe файл. Собственно сама программа представляет из себя один монолитный exe файл без каких либо настроек и dll, забегая немного вперед, скажу что это помогло нам решить проблему. Сам разработчик уже уволился, а так как программу он писал в свое свободное время и никто ему за это не платил, то и исходники программы при увольнении он не захотел оставлять. В принципе тут я его понимаю.

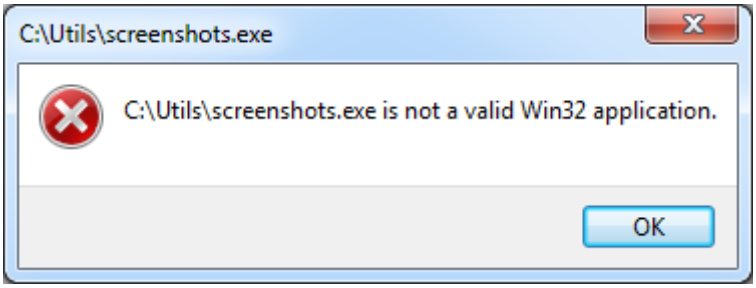
И все было хорошо, до того момента, пока мы не приняли решение о переходе на gmail. А об этой программе как то особо никто и не вспоминал, а точнее о том, что после изменения MX записей и перевода почты на gmail, аутентификация перестанет работать. Можно конечно найти альтернативы, например, тот же monosnap, но во-первых monosnap не такой удобный и более медлительный, во-вторых все уже привыкли к этой программе. А в третьих это не наш метод. Мы ведь не ищем легких путей?

Первый взгляд

Итак, первое, что пришло в голову это попробовать поискать в самом exe файле имя почтового сервера, на который происходит подключение при аутентификации пользователя. И как ни странно, само имя сервера нашлось без проблем



Пробуем поменять имя сервера на pop.gmail.com. Но тут возникает первая проблема. Новое имя сервера должно совпадать по длине с текущим, иначе изменится размер самого exe файла. Ну, если быть точным, то так как у каждой переменной есть адрес, и при изменении длины имени самой переменной, изменятся и все адреса. Именно поэтому при запуске нам и выдается подобная ошибка.




```
> dir screenshots.exe -c
 Volume in drive C is SSD
 Volume Serial Number is 4CA3-E2F5

 Directory of c:\Utils

05.12.2014  21:54         2 593 280 screenshots.exe

 Directory of c:\Utils

               1 File(s)      2 593 280 bytes
               0 Dir(s)  65 791 164 416 bytes free
```

```
> dir screenshots.exe -c
 Volume in drive C is SSD
 Volume Serial Number is 4CA3-E2F5
```

```
 Directory of c:\Utils

14.02.2015  20:38         2 593 273 screenshots.exe

 Directory of c:\Utils

               1 File(s)      2 593 273 bytes
               0 Dir(s)  65 790 615 552 bytes free
```

Для этого создаем в домене CNAME запись на pop.gmail.com, например scrshots.example.net, которая по длине совпадает с secure.emailsrvr.com. Смотрим размер exe файла

```
> dir screenshots.exe -c
 Volume in drive C is SSD
 Volume Serial Number is 4CA3-E2F5

 Directory of c:\Utils

14.02.2015  20:56         2 593 280 screenshots.exe

 Directory of c:\Utils

               1 File(s)      2 593 280 bytes
               0 Dir(s)  65 792 364 544 bytes free
```

Как видим, размер совпадает с оригинальным exe, но md5 сумма при этом отличается, но это вполне нормально и ожидаемо.

original screenshots.exe

```
> md5sum screenshots.exe
542c00e88ddaa13bd90e67447e3e02c5 *screenshots.exe
```

modified screenshots.exe

```
> md5sum screenshots.exe
ff676855b4374ef0cc7746fb8bfcd8ee *screenshots.exe
```

Теперь у нас exe запускается без проблем. И казалось бы, счастье уже у нас в руках, но не тут то было. После длительной паузы получаем ошибку Wrong login/password

Но мы так просто не здаемся, запускаем tcpdump и смотрим, что же там происходит на самом

деле. По логам видно, что наша программа обращается на 110 порт! В то время, как gmail работает только по pop3s (995 порт).

```
# nmap -v -A pop.gmail.com

Starting Nmap 6.40 ( http://nmap.org ) at 2015-02-14 19:28 UTC
NSE: Loaded 110 scripts for scanning.
NSE: Script Pre-scanning.
Initiating Ping Scan at 19:28
Scanning pop.gmail.com (74.125.136.108) [4 ports]
Completed Ping Scan at 19:28, 0.03s elapsed (1 total hosts)
Initiating Parallel DNS resolution of 1 host. at 19:28
Completed Parallel DNS resolution of 1 host. at 19:28, 0.00s elapsed
Initiating SYN Stealth Scan at 19:28
Scanning pop.gmail.com (74.125.136.108) [1000 ports]
Discovered open port 993/tcp on 74.125.136.108
Discovered open port 587/tcp on 74.125.136.108
Discovered open port 25/tcp on 74.125.136.108
Discovered open port 995/tcp on 74.125.136.108
Discovered open port 465/tcp on 74.125.136.108
...
...
...
995/tcp open  ssl/pop3 Google Gmail pop3d (m17mb115231066wlg)
|_pop3-capabilities: RESP-CODES LOGIN-DELAY(300) TOP USER EXPIRE(0) X-
GOOGLE-RICO UIDL SASL(PLAIN XOAUTH2)
| ssl-cert: Subject: commonName=pop.gmail.com/organizationName=Google
Inc/stateOrProvinceName=California/countryName=US
| Issuer: commonName=Google Internet Authority G2/organizationName=Google
Inc/countryName=US
| Public Key type: rsa
| Public Key bits: 2048
...
...
```

Даже если перенаправить все запросы со 110 порта на 995, то программа работать не будет, так как она не понимает ssl. И тут нам на помощь приходит stunnel. Мы делаем небольшой финт и направляем программу не напрямую к gmail.com, а на наш сервер, где на 110 порту как раз и будет запущен stunnel.

Установка и настройка stunnel

Установка тривиальная

```
# yum install stunnel
```

После этого создаем конфигурационный файл gmail-pop3s-client.conf в папке /etc/stunnel/ с таким содержимым

```
# cat /etc/stunnel/gmail-pop3s-client.conf
[gmail-pop3s]
client = yes
accept = 176.198.xxx.xxx:110
connect = pop.gmail.com:995
```

Тестирование

Так как stunnel будет слушать порт ниже 1024, то его необходимо запускать от root.

```
# stunnel /etc/stunnel/gmail-pop3s-client.conf
```

На самом деле можно запустить и от обычного пользователя, кому интересно могут посмотреть в конце статьи

При запуске в консоль ничего не выводится, но в /var/log/secure должны появится следующие строки.

```
Feb 14 19:46:18 web-srv01 stunnel: LOG5[799:140339650004928]: stunnel is in
FIPS mode
Feb 14 19:46:18 web-srv01 stunnel: LOG5[799:140339650004928]: stunnel 4.29
on x86_64-redhat-linux-gnu with OpenSSL 1.0.1e-fips 11 Feb 2013
Feb 14 19:46:18 web-srv01 stunnel: LOG5[799:140339650004928]:
Threading:PTHREAD SSL:ENGINE,FIPS Sockets:POLL,IPv6 Auth:LIBWRAP
Feb 14 19:46:18 web-srv01 stunnel: LOG5[799:140339650004928]: 500 clients
allowed
```

На всякий случай делаем пару базовых проверок, чтобы убедится, что туннель поднялся

```
# netstat -lanp | grep 110
tcp        0      0 176.198.xxx.xxx:110             0.0.0.0:*
LISTEN      12918/stunnel
```

```
# lsof -n -i tcp:110
COMMAND   PID USER   FD   TYPE   DEVICE SIZE/OFF NODE NAME
stunnel 12918 root   12u  IPv4 13882741      0t0  TCP 176.198.xxx.xxx:pop3
(LISTEN)
```

Отлично, теперь проверяем через telnet с удаленного сервера

```
# telnet 176.198.xxx.xxx 110
Trying 176.198.xxx.xxx...
Connected to 176.198.xxx.xxx.
Escape character is '^]'.
+OK Gpop ready for requests from 176.198.xxx.xxx r195mb57795961wlb
quit
+OK Bye r195mb57795961wlb
Connection closed by foreign host.
```

Отлично, это как раз то, что мы и хотели получить. Т.е. наша программа, как и прежде, обращается по «открытому» протоколу pop3 на 110й порт, а уже stunnel делает всю черную работу за нас.

Вот так вот, применив немного смекалки с linux, мы заставили программу продолжить свою работу.

Запуск stunnel от непривилегированного пользователя

Как мы все знаем, привязку к портам < 1024 может делать только суперпользователь root. Если мы попробуем запустить наш туннель от обычного пользователя, то в логах увидим подобное сообщение

```
Feb 15 13:32:55 web-srv01 stunnel: LOG5[25450:140050890143680]:
Threading:PTHREAD SSL:ENGINE,FIPS Sockets:POLL,IPv6 Auth:LIBWRAP
Feb 15 13:32:55 web-srv01 stunnel: LOG5[25450:140050890143680]: 500 clients
allowed
Feb 15 13:32:55 web-srv01 stunnel: LOG3[25450:140050890143680]: Error
binding gmail-pop3s to 176.198.xxx.xxx:110
Feb 15 13:32:55 web-srv01 stunnel: LOG3[25450:140050890143680]: bind:
Permission denied (13)
```

Но на самом деле это ограничение можно и обойти. Для этого достаточно из под root выполнить следующую команду

```
# setcap cap_net_bind_service=ep /usr/bin/stunnel
```

После этого снова пробуем запустить наш туннель. Но перед этим необходимо указать путь к pid файлу, так как на дефолтный файл /var/run/stunnel.pid права есть только у root

```
# ls -la /var/run/stunnel.pid
-rw-r--r-- 1 root root 6 Feb 12 14:46 /var/run/stunnel.pid
```

Итак, добавляем следующую строку в файл /etc/stunnel/gmail-pop3s-client.conf

```
pid = /home/gmail-stunnel/gmail.pid
```

И снова пробуем запустить

```
# lsof -n -i tcp:110
COMMAND   PID          USER   FD   TYPE   DEVICE SIZE/OFF NODE NAME
stunnel 25834 gmail-stunnel   12u  IPv4 16489394      0t0  TCP
176.198.xxx.xxx:pop3 (LISTEN)
```

При это в логах увидим привычные для нас строчки

```
Feb 15 13:44:11 web-srv01 stunnel: LOG5[25828:139677113530304]: stunnel is
in FIPS mode
```

```
Feb 15 13:44:11 web-srv01 stunnel: LOG5[25828:139677113530304]: stunnel 4.29
on x86_64-redhat-linux-gnu with OpenSSL 1.0.1e-fips 11 Feb 2013
Feb 15 13:44:11 web-srv01 stunnel: LOG5[25828:139677113530304]:
Threading:PTHREAD SSL:ENGINE,FIPS Sockets:POLL,IPv6 Auth:LIBWRAP
Feb 15 13:44:11 web-srv01 stunnel: LOG5[25828:139677113530304]: 500 clients
allowed
```

Следует учитывать, что при использовании setcap, программу сможет запускать любой непривилегированный пользователь! Более того, у setcap есть свои ограничения:

- ядро должно быть как минимум 2.6.24
- не будет работать со скриптами, т.е. если ваш файл является bash/sh скриптом, например. В таком случае setcap необходимо устанавливаться для bash интерпретатора, что очень опасно
- не будет работать LD_LIBRARY_PATH. Например, если ваша программа использует собственные библиотеки .../lib/.

- http://stackoverflow.com/questions/413807/is-there-a-way-for-non-root-processes-to-bind-to-privileged-ports-1024-on-l
- http://man7.org/linux/man-pages/man8/setcap.8.html