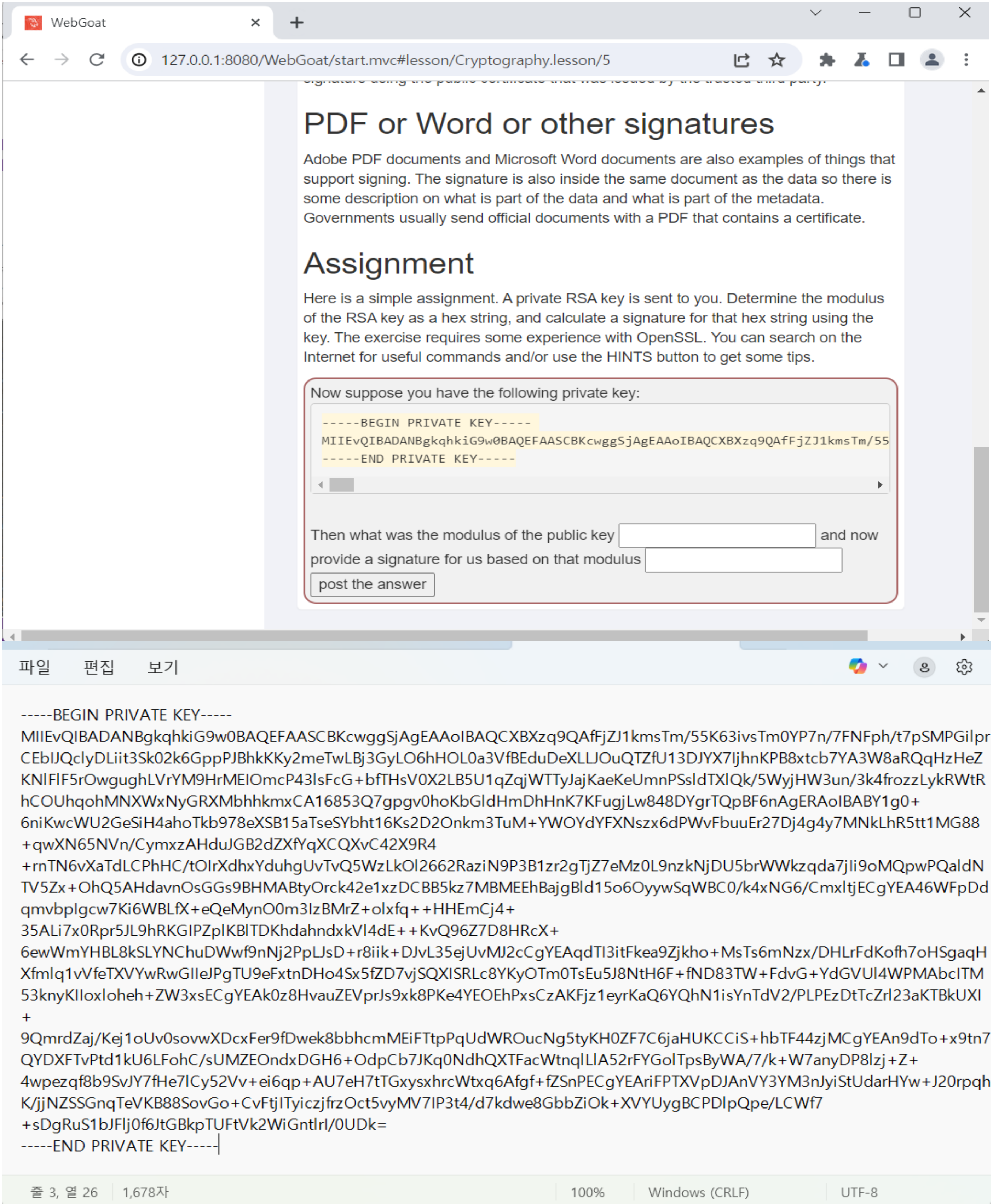
인터넷응용보안 5주차 과제

202121556 곽지현

Cryptographic Failures 6번항.

개인키를 복사하여 메모장에 붙여 넣는다.


WebGoat
127.0.0.1:8080/WebGoat/start.mvc#lesson/Cryptography.lesson/5
파일
편집
보기

PDF or Word or other signatures

Adobe PDF documents and Microsoft Word documents are also examples of things that support signing. The signature is also inside the same document as the data so there is some description on what is part of the data and what is part of the metadata. Governments usually send official documents with a PDF that contains a certificate.

Assignment

Here is a simple assignment. A private RSA key is sent to you. Determine the modulus of the RSA key as a hex string, and calculate a signature for that hex string using the key. The exercise requires some experience with OpenSSL. You can search on the Internet for useful commands and/or use the HINTS button to get some tips.

Now suppose you have the following private key:

```
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQCXBXzq9QAfFjZJ1kmsTm/55
-----END PRIVATE KEY-----
```

Then what was the modulus of the public key ______ and now provide a signature for us based on that modulus ______

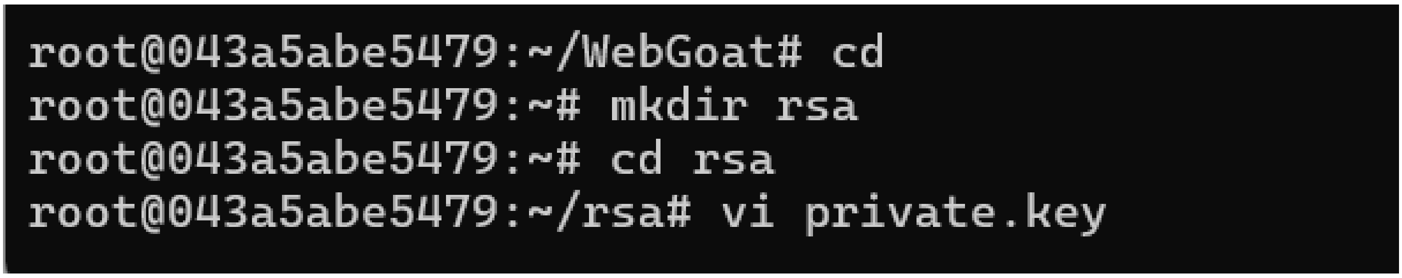
post the answer

```
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQCXBXzq9QAfFjZJ1kmsTm/55K63ivsTm0YP7n/7FNFph/t7pSMPGilpr
CEbIJQclyDLiit3Sk02k6GppPJBhkKKy2meTwLBj3GyLO6hHOL0a3VfBEduDeXLLJOuQTZfU13DJYX7IjhnKPB8xtcb7YA3W8aRQqHzHeZ
KNlFlF5rOwgughLVrYM9HrMElOmcP43lsFcG+bfTHsV0X2LB5U1qZqjWTTyJajKaeKeUmnPSsldTXlQk/5WyjHW3un/3k4frozzLykRWtR
hCOUhqohMNXWxNyGRXMbhhkmxCA16853Q7gpgv0hoKbGldHmDhHnK7KFugjLw848DYgrTQpBF6nAgERAoIBABY1g0+
6niKwcWU2GeSiH4ahoTkb978eXSB15aTseSYbht16Ks2D2Onkm3TuM+YWOYdYFXNszx6dPWvFbuuEr27Dj4g4y7MNkLhR5tt1MG88
+qwXN65NVn/CymxzAHduJGB2dZXfYqXCQXvC42X9R4
+rnTN6vXaTdLCPhHC/tOIrXdhxYduhgUvTvQ5WzLkOl2662RaziN9P3B1zr2gTjZ7eMz0L9nzkNjDU5brWWkzqda7jIi9oMQpwPQaldN
TV5Zx+OhQ5AHdavnOsGGs9BHMABtyOrck42e1xzDCBB5kz7MBMEEhBajgBld15o6OyywSqWBC0/k4xNG6/CmxltjECgYEA46WFpDd
qmvbplgcw7Ki6WBLfX+eQeMynO0m3IzBMrZ+olxfq++HHEmCj4+
35ALi7x0Rpr5JL9hRKGIPZplKBlTDKhdahndxkVl4dE++KvQ96Z7D8HRcX+
6ewWmYHBL8kSLYNChuDWwf9nNj2PpLJsD+r8iik+DJvL35ejUvMJ2cCgYEAqdTI3itFkea9Zjkho+MsTs6mNzx/DHLrFdKofh7oHSgaqH
Xfmlq1vVfeTXVYwRwGIleJPgTU9eFxtnDHo4Sx5fZD7vjSQXISRLc8YKyOTm0TsEu5J8NtH6F+fND83TW+FdvG+YdGVUl4WPMAbcITM
53knyKIloxloheh+ZW3xsECgYEAk0z8HvauZEVprJs9xk8PKe4YEOEhPxsCzAKFjz1eyrKaQ6YQhN1isYnTdV2/PLPEzDtTcZrl23aKTBkUXI
+
9QmrdZaj/Kej1oUv0sovwXDcxFer9fDwek8bbhcmMEiFTtpPqUdWROucNg5tyKH0ZF7C6jaHUKCCiS+hbTF44zjMCgYEAn9dTo+x9tn7
QYDXFTvPtd1kU6LFohC/sUMZEOndxDGH6+OdpCb7JKq0NdhQXTFacWtnqlLlA52rFYGolTpsByWA/7/k+W7anyDP8lzj+Z+
4wpezqf8b9SvJY7fHe7lCy52Vv+ei6qp+AU7eH7tTGxysxhrcWtxq6Afgf+fZSnPECgYEAriFPTXVpDJAnVY3YM3nJyiStUdarHYw+J20rpqh
K/jjNZSSGnqTeVKB88SovGo+CvFtjITyiczjfrzOct5vyMV7IP3t4/d7kdwe8GbbZiOk+XVYUygBCPDlpQpe/LCWf7
+sDgRuS1bJFlj0f6JtGBkpTUFtVk2WiGntlrl/0UDk=
-----END PRIVATE KEY-----
```

줄 3, 열 26 | 1,678자 | 100% | Windows (CRLF) | UTF-8

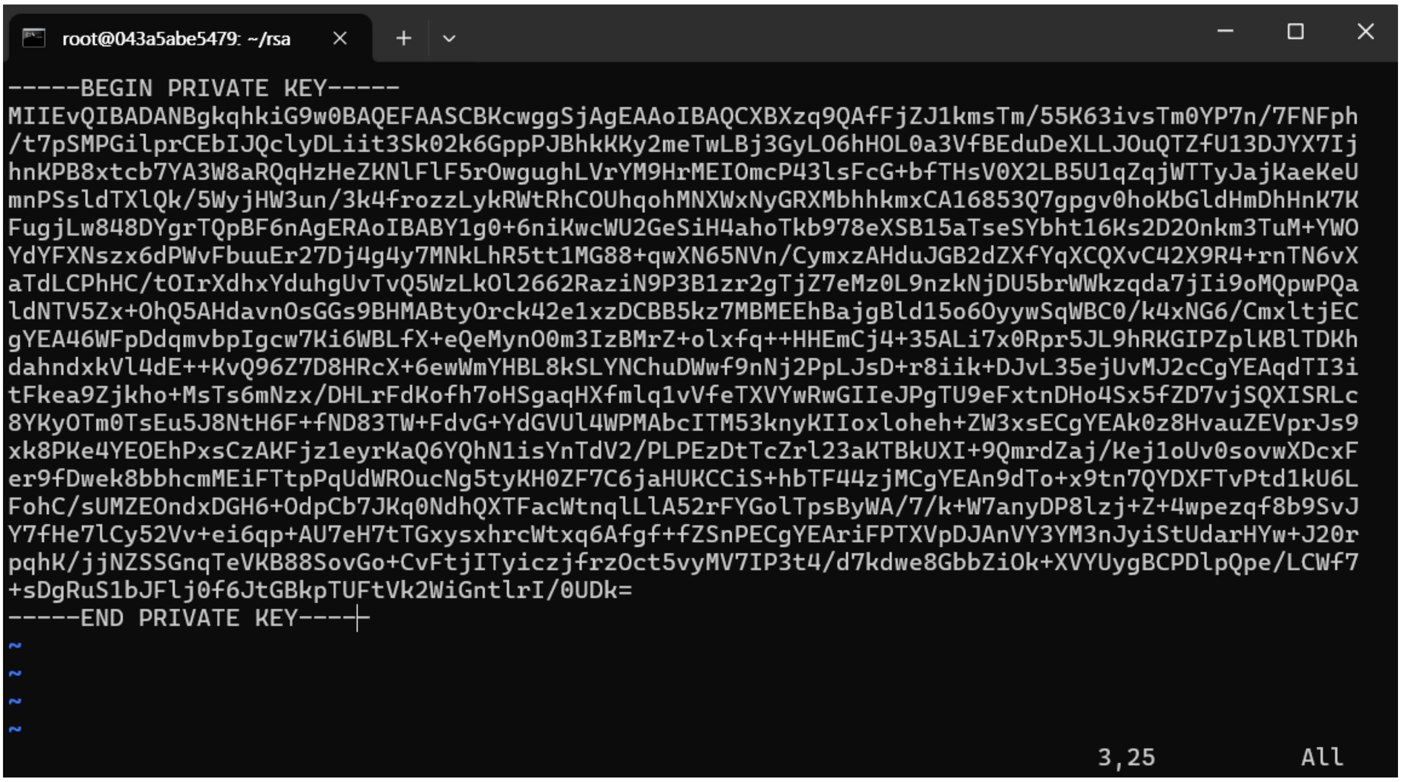
홈 디렉터리로 이동 후 rsa 디렉터리 생성하고 rsa 디렉터리로 이동 vi 에디터로 private.key 생성

cd -> mkdirrsa rsa -> cd rsa -> vi private.key



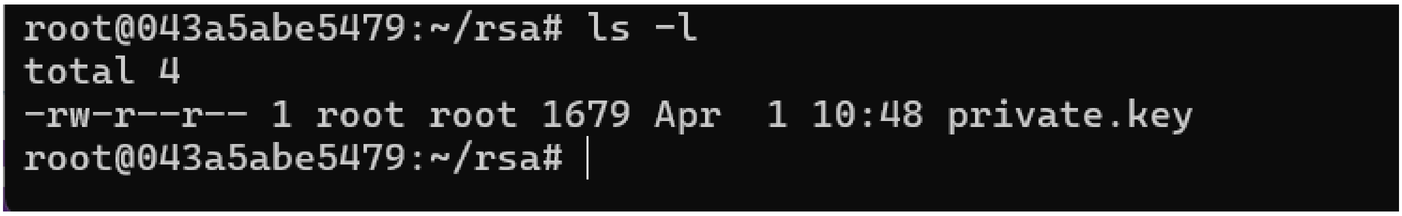
```
root@043a5abe5479:~/WebGoat# cd
root@043a5abe5479:~# mkdir rsa
root@043a5abe5479:~# cd rsa
root@043a5abe5479:~/rsa# vi private.key
```

private.key에 복사한 개인키 데이터를 붙여넣고 저장



```
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQCXBXzq9QAfFjZJ1kmsTm/55K63ivsTm0YP7n/7FNFph
/t7pSMPGilprCEbIJQclyDLiit3Sk02k6GppPJBhkKKy2meTwLBj3GyLO6hHOL0a3VfBEduDeXLLJOuQTZfU13DJYX7Ij
hnKPB8xtcb7YA3W8aRQqHzHeZKNlFlF5rOwgughLVrYM9HrMEIOmcP43lsFcG+bfTHsV0X2LB5U1qZqjWTTyJajKaeKeU
mnPSsldTXlQk/5WyjHW3un/3k4frozzLykRWtRhCOUhqohMNXWxNyGRXMbhhkmxCA16853Q7gpgv0hoKbGldHmDhHnK7K
FugjLw848DYgrTQpBF6nAgERAoIBABY1g0+6niKwcWU2GeSiH4ahoTkb978eXSB15aTseSYbht16Ks2D2Onkm3TuM+YWO
YdYFXNszx6dPWvFbuuEr27Dj4g4y7MNkLhR5tt1MG88+qwXN65NVn/CymxzAHduJGB2dZXfYqXCQXvC42X9R4+rnTN6vX
aTdLCPhHC/tOIrXdhxYduhgUvTvQ5WzLkOl2662RaziN9P3B1zr2gTjZ7eMz0L9nzkNjDU5brWWkzqda7jIi9oMQpwPQa
ldNTV5Zx+OhQ5AHdavnOsGGs9BHMABtyOrck42e1xzDCBB5kz7MBMEEhBajgBld15o6OyywSqWBC0/k4xNG6/CmxltjEC
gYEA46WFpDdqmvbpIgcw7Ki6WBLfX+eQeMynO0m3IzBMrZ+olxfq++HHEmCj4+35ALi7x0Rpr5JL9hRKGIPZplKBlTDKh
dahndxkVl4dE++KvQ96Z7D8HRcX+6ewWmYHBL8kSLYNChuDWwf9nNj2PpLJsD+r8iik+DJvL35ejUvMJ2cCgYEAqdTI3i
tFkea9Zjkho+MsTs6mNzx/DHLrFdKofh7oHSgaqHXfmlq1vVfeTXVYwRwGIIeJPgTU9eFxtnDHo4Sx5fZD7vjSQXISRLc
8YKyOTm0TsEu5J8NtH6F+fND83TW+FdvG+YdGVUl4WPMAbcITM53knyKIIoxloheh+ZW3xsECgYEAk0z8HvauZEVprJs9
xk8PKe4YEOEhPxsCzAKFjz1eyrKaQ6YQhN1isYnTdV2/PLPEzDtTcZrl23aKTBkUXI+9QmrdZaj/Kej1oUv0sovwXDcxF
er9fDwek8bbhcmMEiFTtpPqUdWROucNg5tyKH0ZF7C6jaHUKCCiS+hbTF44zjMCgYEAn9dTo+x9tn7QYDXFTvPtd1kU6L
FohC/sUMZEOndxDGH6+OdpCb7JKq0NdhQXTFacWtnqlLlA52rFYGolTpsByWA/7/k+W7anyDP8lzj+Z+4wpezqf8b9SvJ
Y7fHe7lCy52Vv+ei6qp+AU7eH7tTGxysxhrcWtxq6Afgf+fZSnPECgYEAriFPTXVpDJAnVY3YM3nJyiStUdarHYw+J20r
pqhK/jjNZSSGnqTeVKB88SovGo+CvFtjITyiczjfrzOct5vyMV7IP3t4/d7kdwe8GbbZiOk+XVYUygBCPDlpQpe/LCWf7
+sDgRuS1bJFlj0f6JtGBkpTUFtVk2WiGntlrI/0UDk=
-----END PRIVATE KEY-----
~
~
~
~
                                                                  3,25          All
```

ls -l 명령어로 파일 정보 확인하기



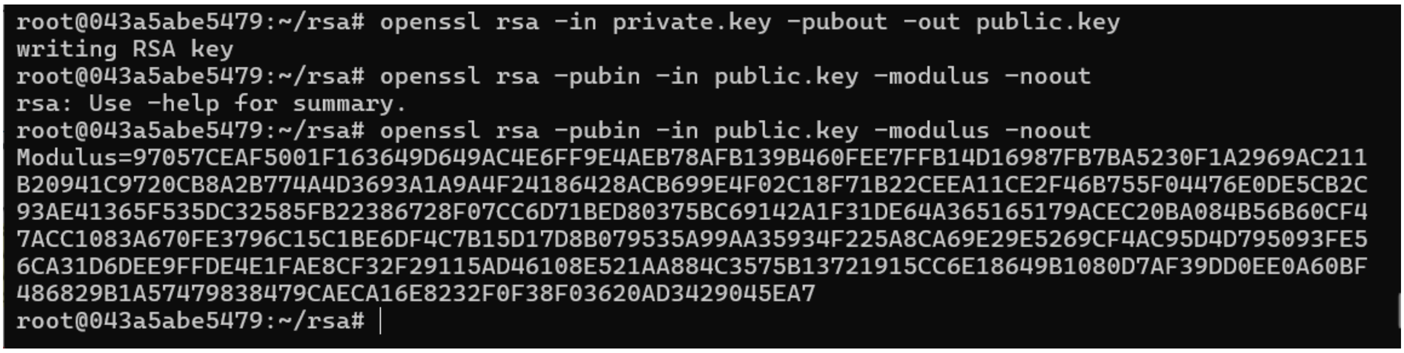
```
root@043a5abe5479:~/rsa# ls -l
total 4
-rw-r--r-- 1 root root 1679 Apr  1 10:48 private.key
root@043a5abe5479:~/rsa#
```

공개키 추출 -> 공개키에서 modulus 추출

openssl rsa -in private.key -pubout -out public.key ->

openssl rsa –pubin –in public.key –modulus –noout

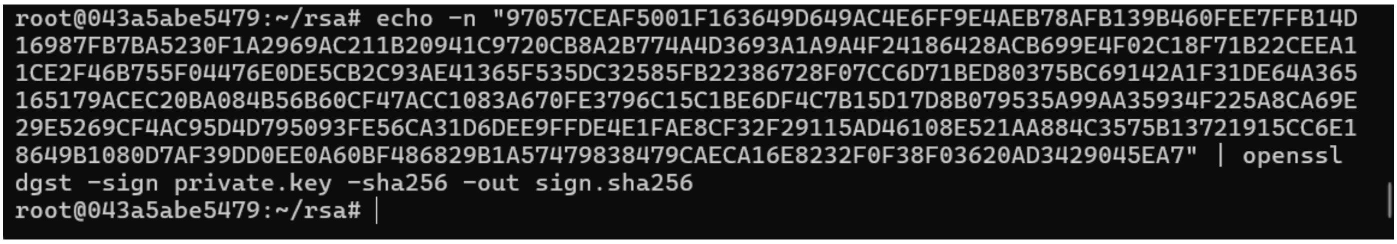
추출된 modulus 복사해두기



```
root@043a5abe5479:~/rsa# openssl rsa -in private.key -pubout -out public.key
writing RSA key
root@043a5abe5479:~/rsa# openssl rsa -pubin -in public.key -modulus -noout
rsa: Use -help for summary.
root@043a5abe5479:~/rsa# openssl rsa -pubin -in public.key -modulus -noout
Modulus=97057CEAF5001F163649D649AC4E6FF9E4AEB78AFB139B460FEE7FFB14D16987FB7BA5230F1A2969AC211
B20941C9720CB8A2B774A4D3693A1A9A4F24186428ACB699E4F02C18F71B22CEEA11CE2F46B755F04476E0DE5CB2C
93AE41365F535DC32585FB22386728F07CC6D71BED80375BC69142A1F31DE64A365165179ACEC20BA084B56B60CF4
7ACC1083A670FE3796C15C1BE6DF4C7B15D17D8B079535A99AA35934F225A8CA69E29E5269CF4AC95D4D795093FE5
6CA31D6DEE9FFDE4E1FAE8CF32F29115AD46108E521AA884C3575B13721915CC6E18649B1080D7AF39DD0EE0A60BF
486829B1A57479838479CAECA16E8232F0F38F03620AD3429045EA7
root@043a5abe5479:~/rsa# |
```

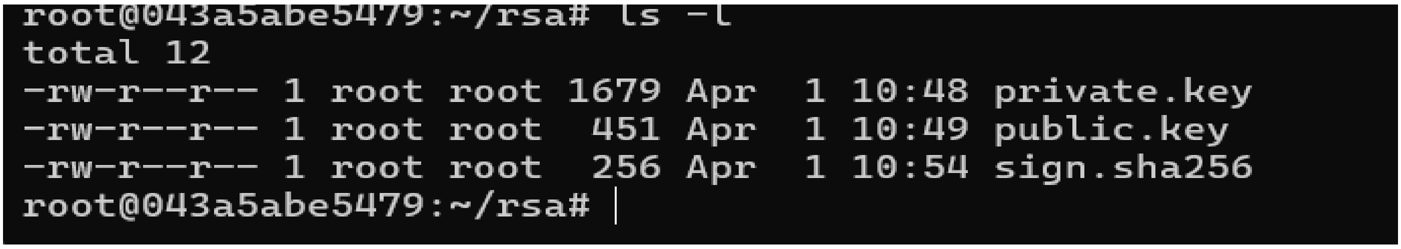
서명용 openssl 명령을 작성후, 수행 [modulus 값을 개인키로 서명하는 과정]

echo -n "복사한 modulus값" | openssl dgst -sign private.key -sha256 -out sign.sha256



```
root@043a5abe5479:~/rsa# echo -n "97057CEAF5001F163649D649AC4E6FF9E4AEB78AFB139B460FEE7FFB14D
16987FB7BA5230F1A2969AC211B20941C9720CB8A2B774A4D3693A1A9A4F24186428ACB699E4F02C18F71B22CEEA1
1CE2F46B755F04476E0DE5CB2C93AE41365F535DC32585FB22386728F07CC6D71BED80375BC69142A1F31DE64A365
165179ACEC20BA084B56B60CF47ACC1083A670FE3796C15C1BE6DF4C7B15D17D8B079535A99AA35934F225A8CA69E
29E5269CF4AC95D4D795093FE56CA31D6DEE9FFDE4E1FAE8CF32F29115AD46108E521AA884C3575B13721915CC6E1
8649B1080D7AF39DD0EE0A60BF486829B1A57479838479CAECA16E8232F0F38F03620AD3429045EA7" | openssl
dgst -sign private.key -sha256 -out sign.sha256
root@043a5abe5479:~/rsa# |
```

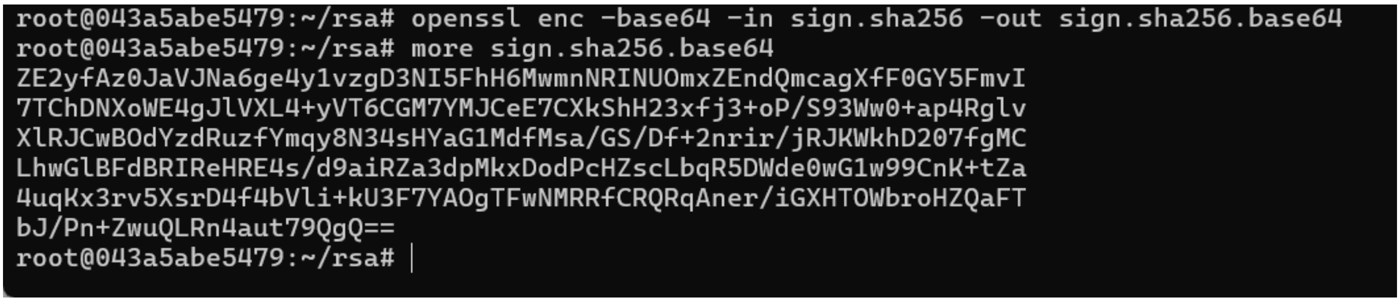
ls -l 명령어로 파일 정보 확인하기



```
root@043a5abe5479:~/rsa# ls -l
total 12
-rw-r--r-- 1 root root 1679 Apr  1 10:48 private.key
-rw-r--r-- 1 root root  451 Apr  1 10:49 public.key
-rw-r--r-- 1 root root  256 Apr  1 10:54 sign.sha256
root@043a5abe5479:~/rsa# |
```

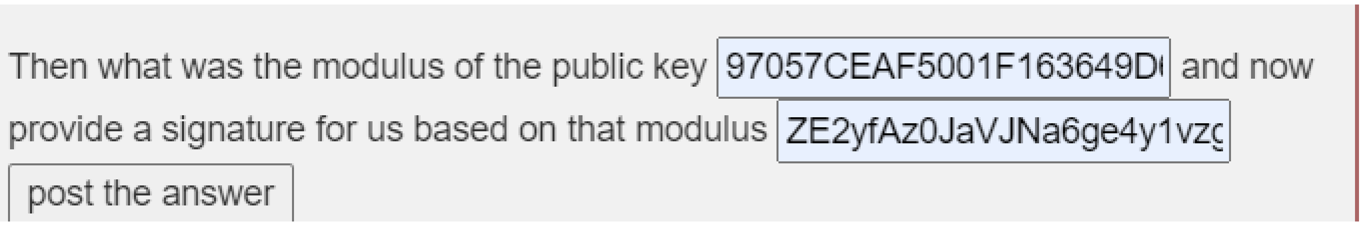
서명값을 base64로 인코딩하고, 그결과를 출력

openssl enc -base64 -in sign.sha256 -out sign.sha256.base64 -> moresign.sha256.base64



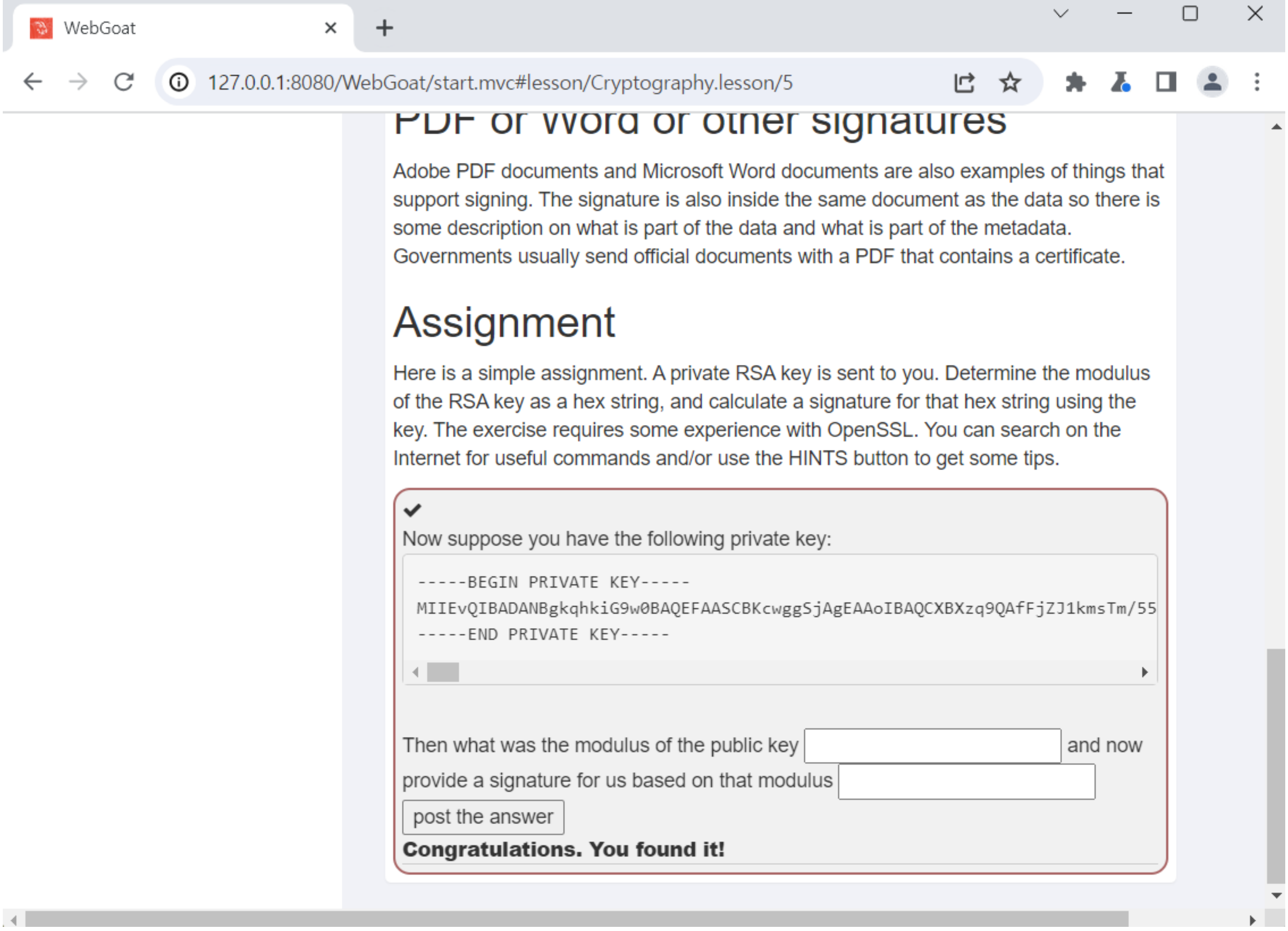
```
root@043a5abe5479:~/rsa# openssl enc -base64 -in sign.sha256 -out sign.sha256.base64
root@043a5abe5479:~/rsa# more sign.sha256.base64
ZE2yfAz0JaVJNa6ge4y1vzgD3NI5FhH6MwmnNRINUOmxZEndQmcagXfF0GY5FmvI
7TChDNXoWE4gJlVXL4+yVT6CGM7YMJCeE7CXkShH23xfj3+oP/S93Ww0+ap4Rglv
XlRJCwBOdYzdRuzfYmqy8N34sHYaG1MdfMsa/GS/Df+2nrir/jRJKWkhD207fgMC
LhwGlBFdBRIReHRE4s/d9aiRZa3dpMkxDodPcHZscLbqR5DWde0wG1w99CnK+tZa
4uqKx3rv5XsrD4f4bVli+kU3F7YAOgTFwNMRRfCRQRqAner/iGXHTOWbroHZQaFT
bJ/Pn+ZwuQLRn4aut79QgQ==
root@043a5abe5479:~/rsa# |
```

위쪽 빈칸에는 Modulus 값을 넣고 밑쪽 빈칸에는 Bae64 인코딩된 서명 값을 넣는다



Then what was the modulus of the public key 97057CEAF5001F163649D and now provide a signature for us based on that modulus ZE2yfAz0JaVJNa6ge4y1vzg

문제 성공!



WebGoat
127.0.0.1:8080/WebGoat/start.mvc#lesson/Cryptography.lesson/5

PDF or Word or other signatures

Adobe PDF documents and Microsoft Word documents are also examples of things that support signing. The signature is also inside the same document as the data so there is some description on what is part of the data and what is part of the metadata. Governments usually send official documents with a PDF that contains a certificate.

Assignment

Here is a simple assignment. A private RSA key is sent to you. Determine the modulus of the RSA key as a hex string, and calculate a signature for that hex string using the key. The exercise requires some experience with OpenSSL. You can search on the Internet for useful commands and/or use the HINTS button to get some tips.

✓

Now suppose you have the following private key:

```
-----BEGIN PRIVATE KEY-----
MIIEvQIBADANBgkqhkiG9w0BAQEFAASCBKcwggSjAgEAAoIBAQCXBXzq9QAfFjZJ1kmsTm/55
-----END PRIVATE KEY-----
```

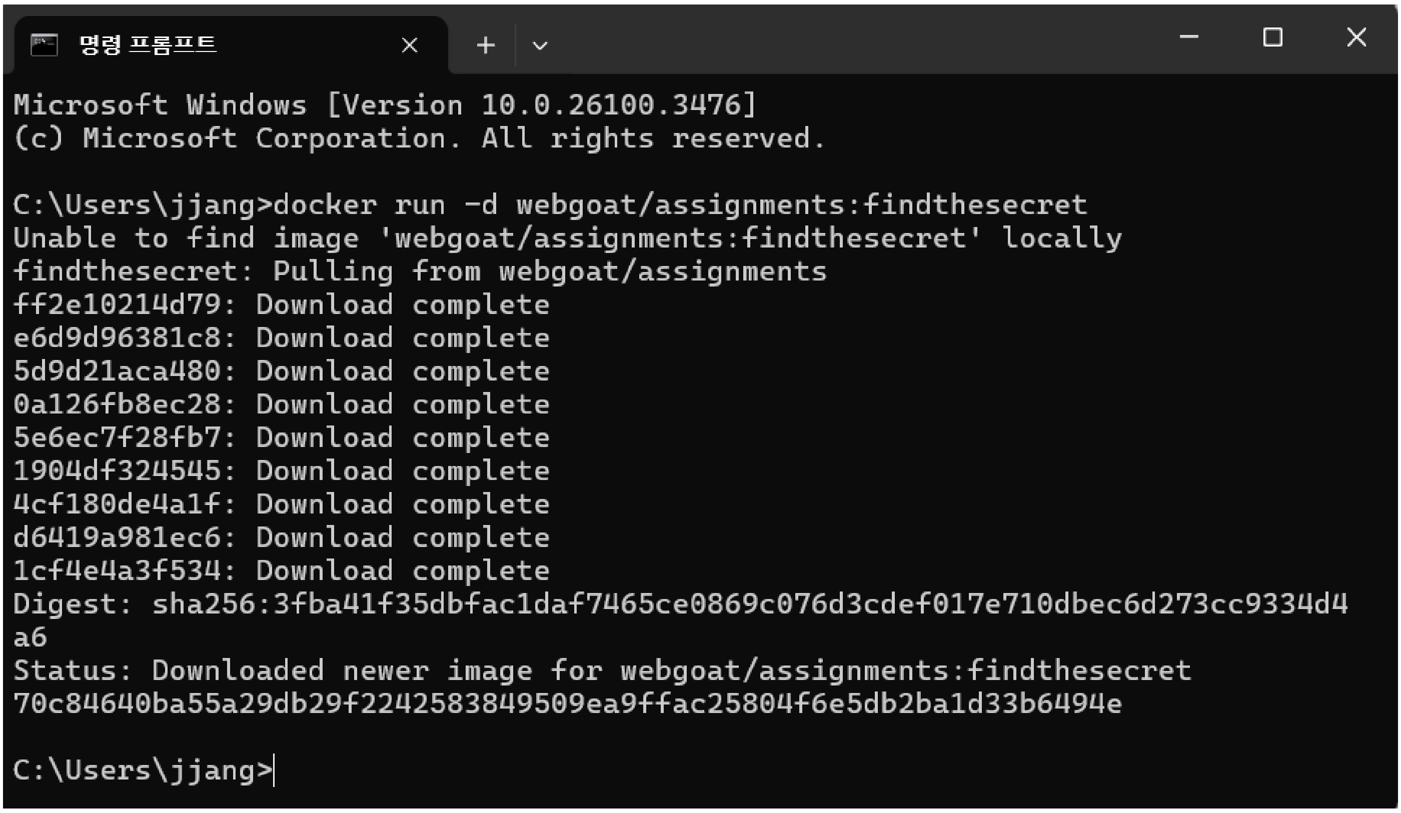
Then what was the modulus of the public key ▭ and now provide a signature for us based on that modulus ▭

Congratulations. You found it!

Cryptographic Failures 8번항.

새 명령창에서 새로운 이미지 다운

docker run -d webgoat/assignments:findthesecret

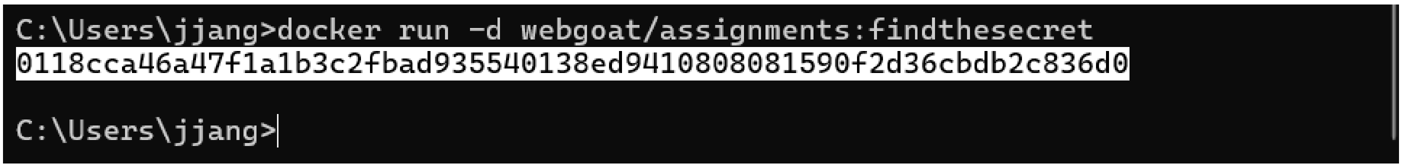


```
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jjang>docker run -d webgoat/assignments:findthesecret
Unable to find image 'webgoat/assignments:findthesecret' locally
findthesecret: Pulling from webgoat/assignments
ff2e10214d79: Download complete
e6d9d96381c8: Download complete
5d9d21aca480: Download complete
0a126fb8ec28: Download complete
5e6ec7f28fb7: Download complete
1904df324545: Download complete
4cf180de4a1f: Download complete
d6419a981ec6: Download complete
1cf4e4a3f534: Download complete
Digest: sha256:3fba41f35dbfac1daf7465ce0869c076d3cdef017e710dbec6d273cc9334d4
a6
Status: Downloaded newer image for webgoat/assignments:findthesecret
70c84640ba55a29db29f2242583849509ea9ffac25804f6e5db2ba1d33b6494e

C:\Users\jjang>
```

동일한 명령어 수행 후, 나오는 컨테이너 ID를 복사

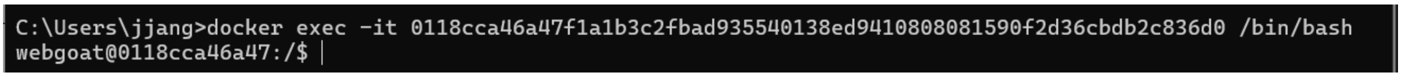


```
C:\Users\jjang>docker run -d webgoat/assignments:findthesecret
0118cca46a47f1a1b3c2fbad935540138ed9410808081590f2d36cbdb2c836d0

C:\Users\jjang>
```

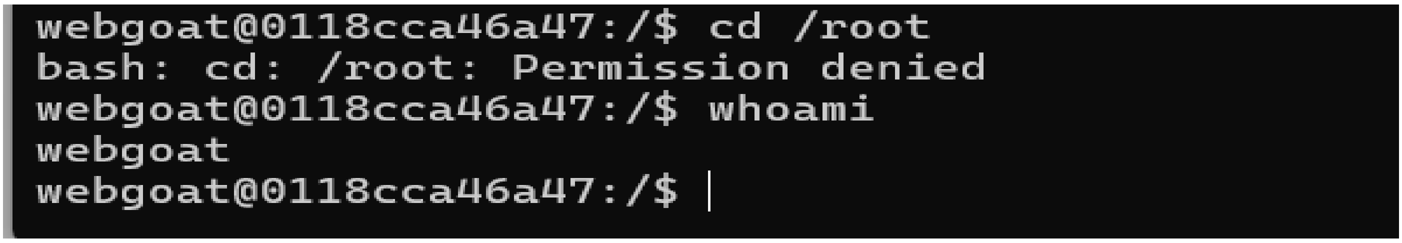
컨테이너에 접속

dockerexec –it [복사한 ID] /bin/bash



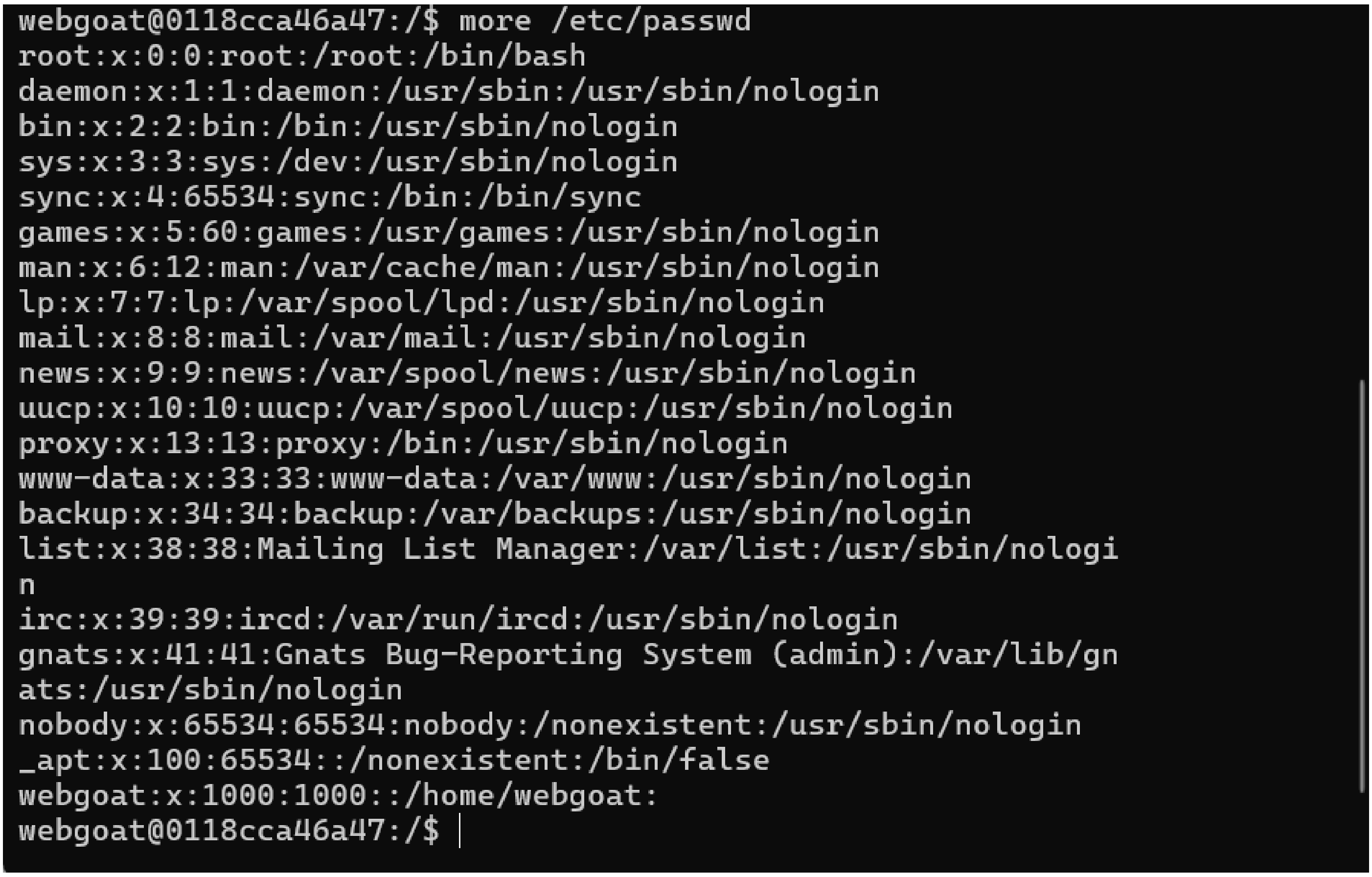
```
C:\Users\jjang>docker exec -it 0118cca46a47f1a1b3c2fbad935540138ed9410808081590f2d36cbdb2c836d0 /bin/bash
webgoat@0118cca46a47:/$
```

/root 디렉터리에 대한 권한이 없음을 확인



```
webgoat@0118cca46a47:/$ cd /root
bash: cd: /root: Permission denied
webgoat@0118cca46a47:/$ whoami
webgoat
webgoat@0118cca46a47:/$
```

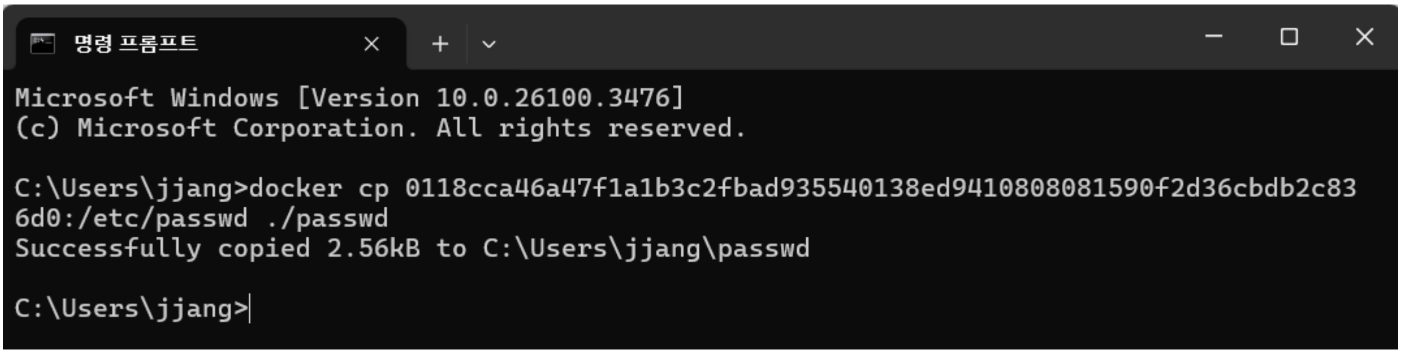
more 명령어를 사용하여 /etc/passwd 파일 열어보고 소유자 확인



```
webgoat@0118cca46a47:/$ more /etc/passwd
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologi
n
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gn
ats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/bin/false
webgoat:x:1000:1000::/home/webgoat:
webgoat@0118cca46a47:/$
```

새로운 명령창에서 도커 내부의 파일을 윈도우로 복사

docker cp [복사한 ID]:/etc/passwd ./passwd

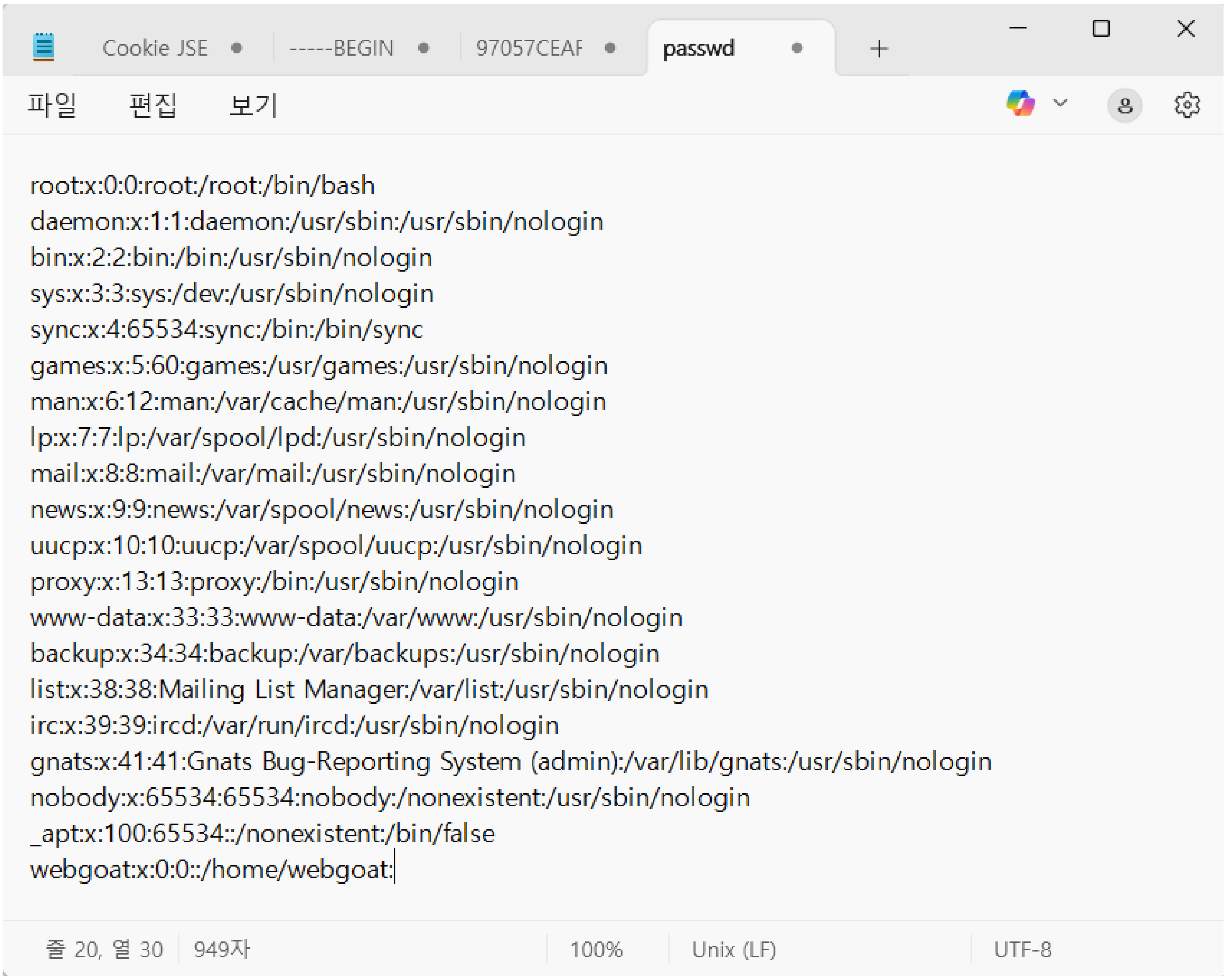


```
Microsoft Windows [Version 10.0.26100.3476]
(c) Microsoft Corporation. All rights reserved.

C:\Users\jjang>docker cp 0118cca46a47f1a1b3c2fbad935540138ed9410808081590f2d36cbdb2c83
6d0:/etc/passwd ./passwd
Successfully copied 2.56kB to C:\Users\jjang\passwd

C:\Users\jjang>
```

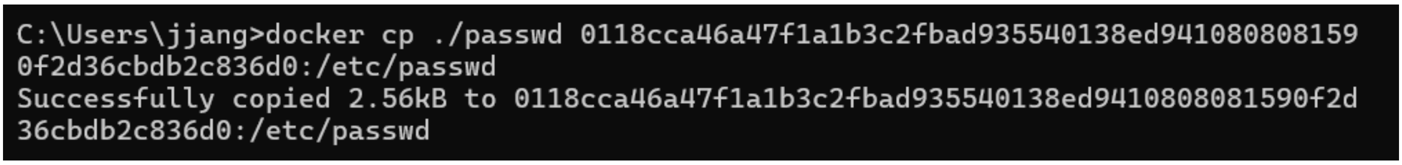
복사해 온 파일을 메모장으로 열어서 webgoat 계정의 uid/gid를 root와 동일한 0으로 변경 후 저장



```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/usr/sbin/nologin
bin:x:2:2:bin:/bin:/usr/sbin/nologin
sys:x:3:3:sys:/dev:/usr/sbin/nologin
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/usr/sbin/nologin
man:x:6:12:man:/var/cache/man:/usr/sbin/nologin
lp:x:7:7:lp:/var/spool/lpd:/usr/sbin/nologin
mail:x:8:8:mail:/var/mail:/usr/sbin/nologin
news:x:9:9:news:/var/spool/news:/usr/sbin/nologin
uucp:x:10:10:uucp:/var/spool/uucp:/usr/sbin/nologin
proxy:x:13:13:proxy:/bin:/usr/sbin/nologin
www-data:x:33:33:www-data:/var/www:/usr/sbin/nologin
backup:x:34:34:backup:/var/backups:/usr/sbin/nologin
list:x:38:38:Mailing List Manager:/var/list:/usr/sbin/nologin
irc:x:39:39:ircd:/var/run/ircd:/usr/sbin/nologin
gnats:x:41:41:Gnats Bug-Reporting System (admin):/var/lib/gnats:/usr/sbin/nologin
nobody:x:65534:65534:nobody:/nonexistent:/usr/sbin/nologin
_apt:x:100:65534::/nonexistent:/bin/false
webgoat:x:0:0::/home/webgoat:
```

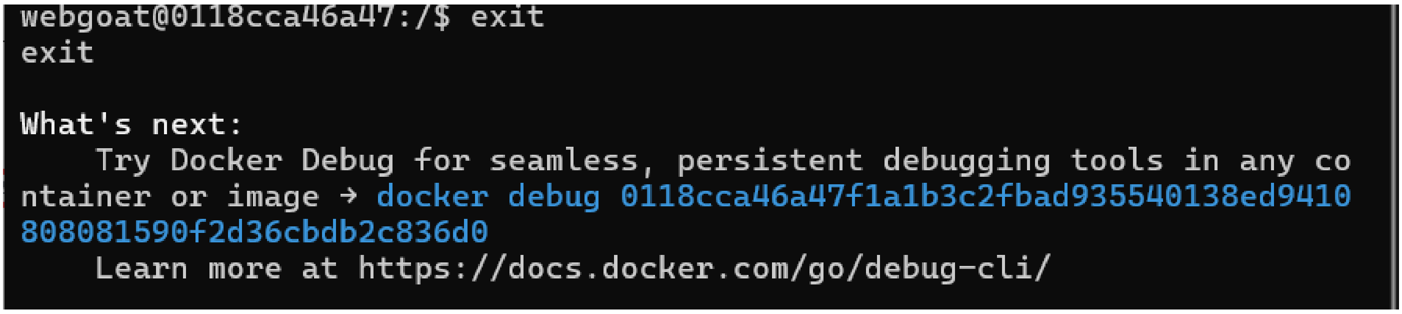
세번째 명령창에서 passwd 파일을 도커 내부로 복사

docker cp ./passwd [복사한 ID]:/etc/passwd



```
C:\Users\jjang>docker cp ./passwd 0118cca46a47f1a1b3c2fbad935540138ed941080808159
0f2d36cbdb2c836d0:/etc/passwd
Successfully copied 2.56kB to 0118cca46a47f1a1b3c2fbad935540138ed9410808081590f2d
36cbdb2c836d0:/etc/passwd
```

현재 접속 중인 컨테이너(두번째 명령창)에서 exit를 입력하여 로그아웃



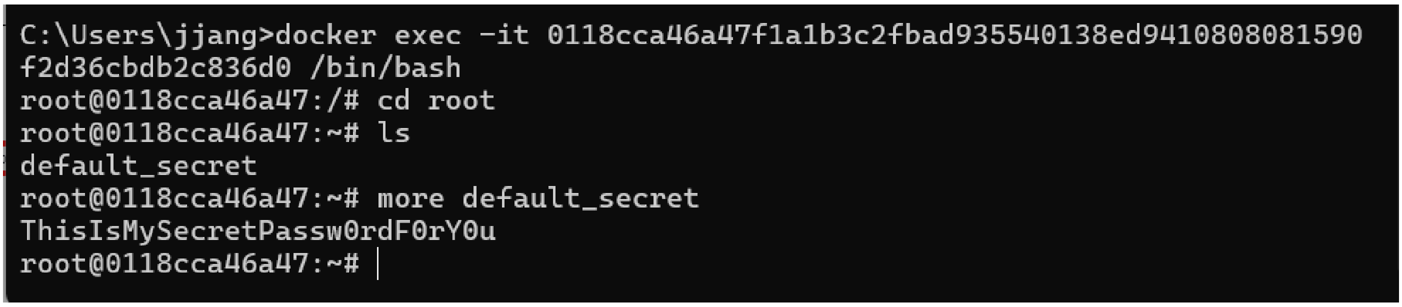
```
webgoat@0118cca46a47:/$ exit
exit

What's next:
    Try Docker Debug for seamless, persistent debugging tools in any co
ntainer or image → docker debug 0118cca46a47f1a1b3c2fbad935540138ed9410
808081590f2d36cbdb2c836d0
    Learn more at https://docs.docker.com/go/debug-cli/
```

다시 컨테이너에 접속하면 root 디렉터리에 들어갈 수 있다

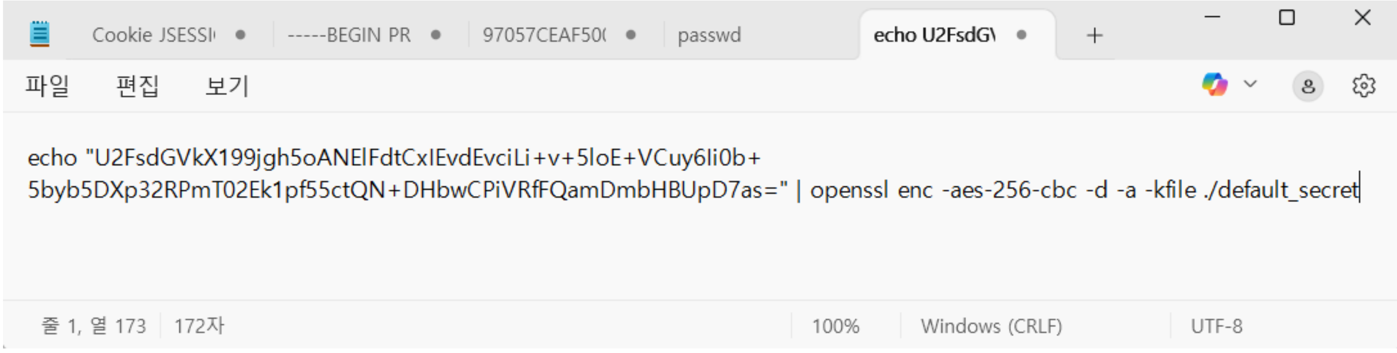
root 디렉터리에 default_secret 파일이 있고

more 명령어를 사용하여 파일 내용 확인 (ThisIsMySecretPassw0rdF0rY0u)



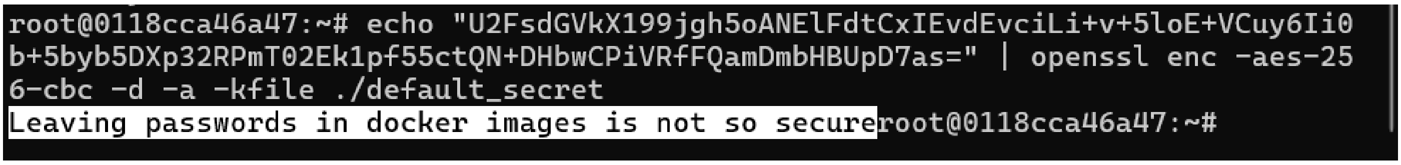
```
C:\Users\jjang>docker exec -it 0118cca46a47f1a1b3c2fbad935540138ed9410808081590
f2d36cbdb2c836d0 /bin/bash
root@0118cca46a47:/# cd root
root@0118cca46a47:~# ls
default_secret
root@0118cca46a47:~# more default_secret
ThisIsMySecretPassw0rdF0rY0u
root@0118cca46a47:~# |
```

메모장에 홈페이지 내용을 복사하여 붙여넣고 ... 부분을 ./default_secret 파일로 수정



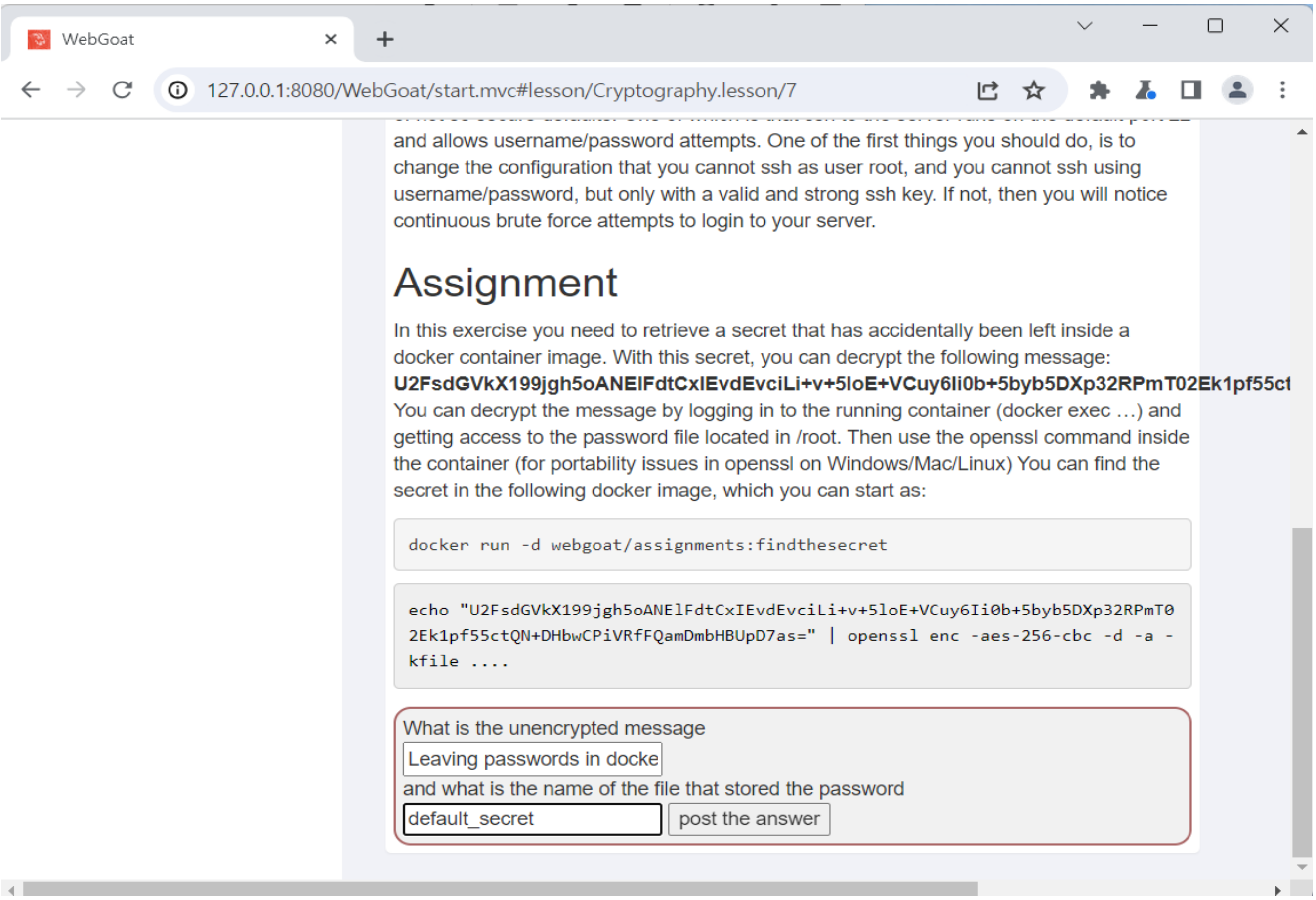
```
echo "U2FsdGVkX199jgh5oANElFdtCxIEvdEvciLi+v+5loE+VCuy6Ii0b+
5byb5DXp32RPmT02Ek1pf55ctQN+DHbwCPiVRfFQamDmbHBUpD7as=" | openssl enc -aes-256-cbc -d -a -kfile ./default_secret
```

명령창에 그대로 붙어넣으면 복호화된 평문을 확인 가능



```
root@0118cca46a47:~# echo "U2FsdGVkX199jgh5oANElFdtCxIEvdEvciLi+v+5loE+VCuy6Ii0
b+5byb5DXp32RPmT02Ek1pf55ctQN+DHbwCPiVRfFQamDmbHBUpD7as=" | openssl enc -aes-25
6-cbc -d -a -kfile ./default_secret
Leaving passwords in docker images is not so secureroot@0118cca46a47:~#
```

첫 번째 빈칸에는 복호화된 평문을 넣고, 두 번째 빈칸에는 키 파일 이름을 넣는다


WebGoat
127.0.0.1:8080/WebGoat/start.mvc#lesson/Cryptography.lesson/7

and allows username/password attempts. One of the first things you should do, is to change the configuration that you cannot ssh as user root, and you cannot ssh using username/password, but only with a valid and strong ssh key. If not, then you will notice continuous brute force attempts to login to your server.

Assignment

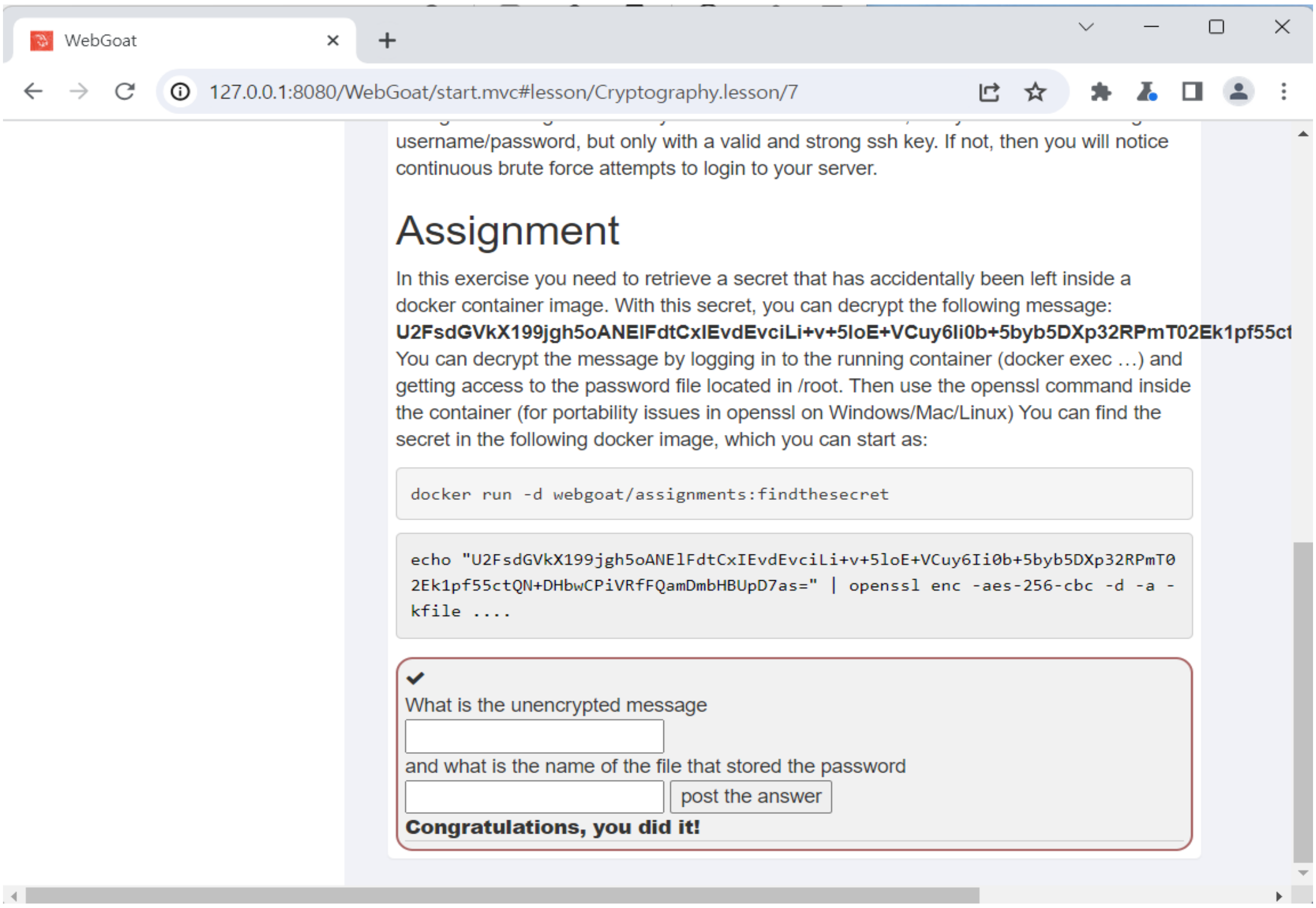
In this exercise you need to retrieve a secret that has accidentally been left inside a docker container image. With this secret, you can decrypt the following message:
U2FsdGVkX199jgh5oANElFdtCxIEvdEvciLi+v+5loE+VCuy6li0b+5byb5DXp32RPmT02Ek1pf55ct
You can decrypt the message by logging in to the running container (docker exec …) and getting access to the password file located in /root. Then use the openssl command inside the container (for portability issues in openssl on Windows/Mac/Linux) You can find the secret in the following docker image, which you can start as:

```
docker run -d webgoat/assignments:findthesecret
```

```
echo "U2FsdGVkX199jgh5oANElFdtCxIEvdEvciLi+v+5loE+VCuy6Ii0b+5byb5DXp32RPmT0
2Ek1pf55ctQN+DHbwCPiVRfFQamDmbHBUpD7as=" | openssl enc -aes-256-cbc -d -a -
kfile ....
```

What is the unencrypted message
Leaving passwords in docke
and what is the name of the file that stored the password
default_secret

문제 성공!


WebGoat
127.0.0.1:8080/WebGoat/start.mvc#lesson/Cryptography.lesson/7

username/password, but only with a valid and strong ssh key. If not, then you will notice continuous brute force attempts to login to your server.

Assignment

In this exercise you need to retrieve a secret that has accidentally been left inside a docker container image. With this secret, you can decrypt the following message:
U2FsdGVkX199jgh5oANElFdtCxIEvdEvciLi+v+5loE+VCuy6li0b+5byb5DXp32RPmT02Ek1pf55ct
You can decrypt the message by logging in to the running container (docker exec …) and getting access to the password file located in /root. Then use the openssl command inside the container (for portability issues in openssl on Windows/Mac/Linux) You can find the secret in the following docker image, which you can start as:

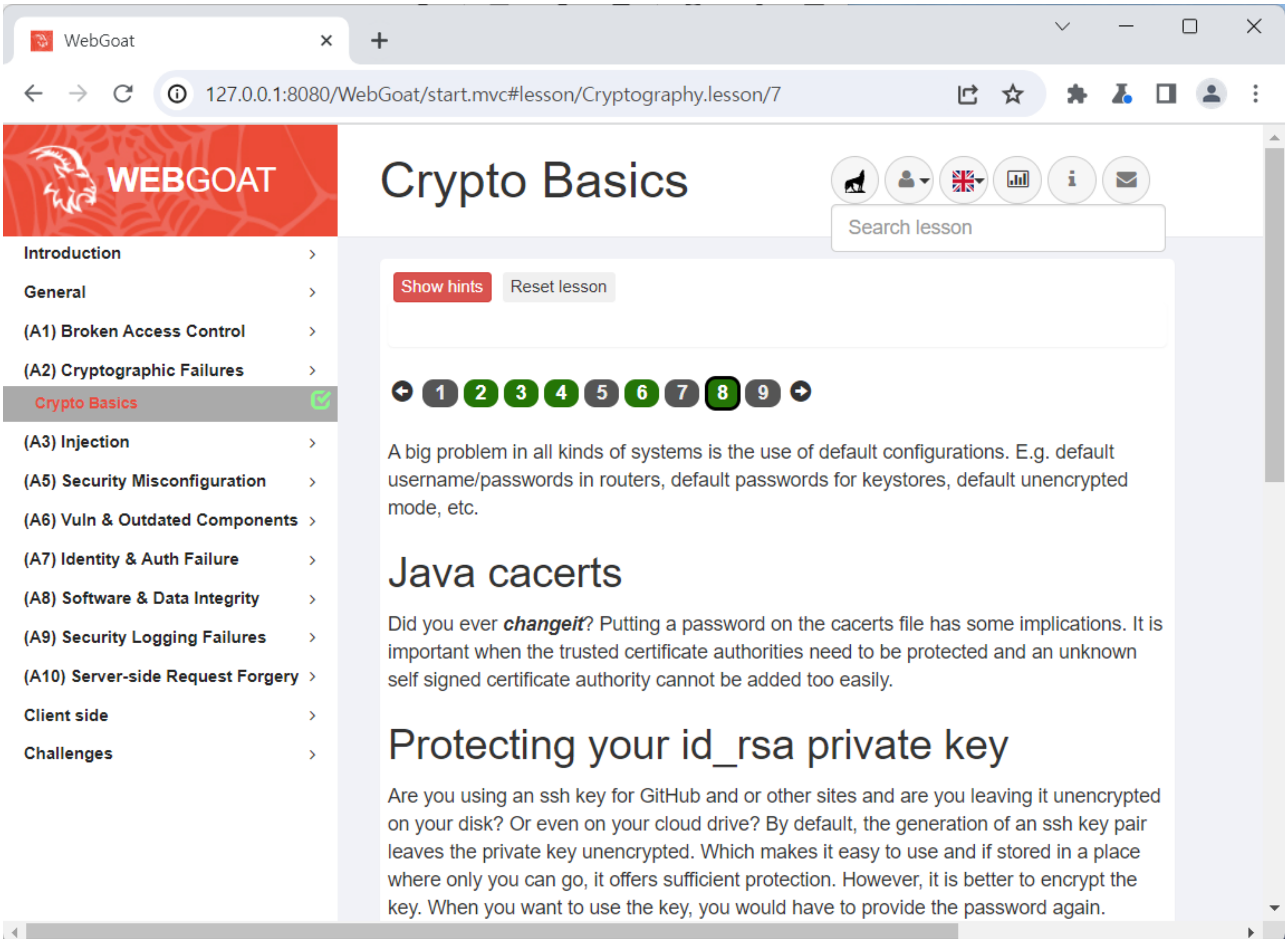
```
docker run -d webgoat/assignments:findthesecret
```

```
echo "U2FsdGVkX199jgh5oANElFdtCxIEvdEvciLi+v+5loE+VCuy6Ii0b+5byb5DXp32RPmT0
2Ek1pf55ctQN+DHbwCPiVRfFQamDmbHBUpD7as=" | openssl enc -aes-256-cbc -d -a -
kfile ....
```

✓
What is the unencrypted message
and what is the name of the file that stored the password

Congratulations, you did it!

모든 문제 성공!



WebGoat
127.0.0.1:8080/WebGoat/start.mvc#lesson/Cryptography.lesson/7
WEBGOAT
Search lesson
Introduction
General
(A1) Broken Access Control
(A2) Cryptographic Failures
Crypto Basics
(A3) Injection
(A5) Security Misconfiguration
(A6) Vuln & Outdated Components
(A7) Identity & Auth Failure
(A8) Software & Data Integrity
(A9) Security Logging Failures
(A10) Server-side Request Forgery
Client side
Challenges
Show hints
Reset lesson
1 2 3 4 5 6 7 8 9

Crypto Basics

A big problem in all kinds of systems is the use of default configurations. E.g. default username/passwords in routers, default passwords for keystores, default unencrypted mode, etc.

Java cacerts

Did you ever ***changeit***? Putting a password on the cacerts file has some implications. It is important when the trusted certificate authorities need to be protected and an unknown self signed certificate authority cannot be added too easily.

Protecting your id_rsa private key

Are you using an ssh key for GitHub and or other sites and are you leaving it unencrypted on your disk? Or even on your cloud drive? By default, the generation of an ssh key pair leaves the private key unencrypted. Which makes it easy to use and if stored in a place where only you can go, it offers sufficient protection. However, it is better to encrypt the key. When you want to use the key, you would have to provide the password again.