

Backdoor Assignment

Testing Document

Ian Lee and Luke Tao

Assignment 2 for COMP 8505

October 6, 2014

Test Case Table

#	Description	Tool	Expected Output	Pass/Fail
1	Masks Backdoor Process	ps	backdoor renamed	Pass
2	Validates Password	Client	works on matching password; fails on non matching password	Pass
3	Encrypts Payloads	wireshark	unable to read payloads of packets	Pass
4	Commands are executed on backdoor host	touch a file	file is created on backdoor host	Pass
5	Command results are sent back to Client	Client	Results are shown on screen	Pass

Test Case #1

In this test case, we show that the process is masked.

```
[root@DataComm src]# ../exe/server -d root 6165 0.0 0.0 0 0 ? S 00:17 0:00 [kworker/u32:1]
Daemon started root 6278 0.0 0.0 0 0 ? S 00:20 0:00 [kworker/0:0]
Daemon mode enabled. root 6284 0.0 0.0 10824 1912 pts/0 S 00:20 0:00 /sbin/rngd -f -d
Process name masked as: /sbin/rngd -f root 6288 0.0 0.0 123364 2708 pts/1 R+ 00:20 0:00 ps -aux
[root@DataComm src]# [root@DataComm src]#
```

In the screenshot above, the process is renamed to '/sbin/rngd -f -d'

Test Case #2

Here, you can see that when a client enters the wrong password “asdfjkl;”, they get no command results back from the server. However, when the client enters the right password “uest1onQ?”, they should be able to get the results back (see test case #3 for proof).

```
[root@DataComm src]# ../exe/client -a 192.168.0.2
Enter a password:
Enter a command: ls -l
Sending data: asdfjkl; 0 cmd[ls -l]cmd

Enter a command: ^C
[root@DataComm src]# ../exe/client -a 192.168.0.2
Enter a password:
Enter a command: ls -l
Sending data: uest1onQ? 0 cmd[ls -l]cmd
```

In the following screenshot, the server receives an incorrect password and ignores the packet, and a packet with the correct password, and executes it.

```
[root@DataComm src]# ../exe/server
Daemon mode disabled.
Process name masked as: /sbin/rngd -f
Incorrect Password
Password Authenticated. Executing command.
Packet: uest1onQ? 1 cmd[total 148]cmd
Packet: uest1onQ? 1 cmd[-rw-r--r-- 1 root root 5418 Oct  5 23:44 backdoor-client.clcmd]
```

Test Case #3

This screenshot shows after the client decrypts the command line results using the XOR algorithm, it displays properly.

```
[root@DataComm src]# ../exe/client -a 192.168.0.2
Enter a password:
Enter a command: ls -l
Sending data: uestlonQ? 0 cmd[ls -l]cmd
total 148
-rw-r--r-- 1 root root 5124 Oct 5 22:26 backdoor-client.c
-rw-r--r-- 1 root root 480 Oct 4 21:24 backdoor-client.h
-rw-r--r-- 1 root root 12320 Oct 5 23:41 backdoor-client.o
-rw-r--r-- 1 root root 4654 Oct 5 23:37 backdoor-server.c
-rw-r--r-- 1 root root 417 Oct 5 23:06 backdoor-server.h
-rw-r--r-- 1 root root 10648 Oct 5 23:41 backdoor-server.o
-rw-r--r-- 1 root root 7264 Oct 5 23:41 isaac_encryption.o
drwxr-xr-x 2 root root 4096 Oct 5 21:15 lib
-rw-r--r-- 1 root root 991 Oct 4 19:06 main.c
-rw-r--r-- 1 root root 187 Oct 4 19:06 main.h
-rw-r--r-- 1 root root 1016 Oct 4 19:06 Makefile
-rw-r--r-- 1 root root 8547 Oct 5 23:21 pktcap.c
-rw-r--r-- 1 root root 2716 Oct 4 20:17 pktcap.h
-rw-r--r-- 1 root root 19160 Oct 5 23:41 pktcap.o
-rw-r--r-- 1 root root 0 Oct 5 22:59 test
-rw-r--r-- 1 root root 0 Oct 5 22:59 test.txt
-rw-r--r-- 1 root root 10757 Oct 5 23:40 utils.c
-rw-r--r-- 1 root root 1065 Oct 5 22:25 utils.h
-rw-r--r-- 1 root root 19240 Oct 5 23:41 utils.o
```

For encryption, this packet below shows the command result being sent back to the client.

```
Packet: uestlonQ? 1 cmd[-rw-r--r-- 1 root root 1065 Oct 5 22:25 utils.h]cmd
Packet: uestlonQ? 1 cmd[-rw-r--r-- 1 root root 19240 Oct 5 23:41 utils.o]cmd
Sent
```

This screenshot in Wireshark shows the same command result mentioned above being encrypted:

36084	5147.8981680	192.168.0.2	192.168.0.1	TCP	123 6476 > sops [SYN, Reserved] Seq=0 Win=31416 Len=69
36085	5147.8981900	192.168.0.2	192.168.0.1	TCP	123 5872 > sops [SYN, Reserved] Seq=0 Win=31416 Len=69
⊕ Frame 36085: 123 bytes on wire (984 bits), 123 bytes captured (984 bits) on interface 0					
⊕ Ethernet II, Src: Dell_a3:43:25 (78:2b:cb:a3:43:25), Dst: Dell_a3:eb:af (78:2b:cb:a3:eb:af)					
⊕ Internet Protocol Version 4, Src: 192.168.0.2 (192.168.0.2), Dst: 192.168.0.1 (192.168.0.1)					
⊕ Transmission Control Protocol, Src Port: 5872 (5872), Dst Port: sops (3944), Seq: 0, Len: 69					
⊖ Data (69 bytes)					
Data: 423f1718031c0f0667446b145021366961033245391b672a...					
[Length: 69]					
0000	78 2b cb a3 eb af 78 2b cb a3 43 25 08 00 45 00	x+....x+ ..C%.E.			
0010	00 6d 9b bc 00 00 64 06 39 7b c0 a8 00 02 c0 a8	.m....d. 9{.....			
0020	00 01 16 f0 0f 68 00 00 0a 70 fc fc a7 e1 58 02h.. .p....X.			
0030	7a b8 04 a0 d1 8f 42 3f 17 18 03 1c 0f 06 67 44	z.....B?gD			
0040	6b 14 50 21 36 69 61 03 32 45 39 1b 67 2a 40 5a	k.Pi6ia. 2E9.g*@Z			
0050	79 58 57 45 3f 56 41 70 21 2e 31 5b 4a 7b 11 45	yXWE?VAP !.l[j{.E			
0060	03 6a 4a 63 3e 39 48 61 62 16 18 73 7e 08 6e 1a	.jJc>9Ha b..s~.n.			
0070	29 2c 5c 2e 46 56 5e 10 0f 32 2a),\..FV^.. .2*			

Test Case #4

In this test case we will try to create a file using the command 'touch test.txt'. To verify that the file is created on the server, ls will be used.

Client side:

```
[root@DataComm src]# ../exe/client -a 192.168.0.2
Enter a password:
Enter a command: touch test.txt
Sending data: uestlonQ? 0 cmd[touch test.txt]cmd

Enter a command: ls -l
Sending data: uestlonQ? 0 cmd[ls -l]cmd
total 148
-rw-r--r-- 1 root root 5418 Oct 5 23:44 backdoor-client.c
-rw-r--r-- 1 root root 480 Oct 4 21:24 backdoor-client.h
-rw-r--r-- 1 root root 13376 Oct 6 00:20 backdoor-client.o
-rw-r--r-- 1 root root 4656 Oct 6 00:19 backdoor-server.c
-rw-r--r-- 1 root root 417 Oct 5 23:06 backdoor-server.h
-rw-r--r-- 1 root root 10616 Oct 6 00:20 backdoor-server.o
-rw-r--r-- 1 root root 7264 Oct 6 00:20 isaac_encryption.o
drwxr-xr-x 2 root root 4096 Oct 5 21:15 lib
-rw-r--r-- 1 root root 991 Oct 4 19:06 main.c
-rw-r--r-- 1 root root 187 Oct 4 19:06 main.h
-rw-r--r-- 1 root root 1016 Oct 4 19:06 Makefile
-rw-r--r-- 1 root root 8449 Oct 5 23:44 pktcap.c
-rw-r--r-- 1 root root 2716 Oct 4 20:17 pktcap.h
-rw-r--r-- 1 root root 18848 Oct 6 00:20 pktcap.o
-rw-r--r-- 1 root root 0 Oct 6 00:23 test.txt
-rw-r--r-- 1 root root 10757 Oct 5 23:40 utils.c
-rw-r--r-- 1 root root 1086 Oct 5 23:44 utils.h
-rw-r--r-- 1 root root 19240 Oct 6 00:20 utils.o
```

Server Side:

```
[root@DataComm src]# ../exe/server -d
Daemon started
Daemon mode enabled.
Process name masked as: /sbin/rngd -f
[root@DataComm src]# Password Authenticated. Executing command.
Password Authenticated. Executing command.
Packet: uestlonQ? 1 cmd[total 148]cmd
Packet: uestlonQ? 1 cmd[-rw-r--r-- 1 root root 5418 Oct 5 23:44 backdoor-client.c]cmd

[root@DataComm src]# ls
backdoor-client.c  backdoor-server.c  isaac_encryption.o  main.h  pktpcap.h  utils.h
backdoor-client.h  backdoor-server.h  lib                 Makefile  pktpcap.o  utils.o
backdoor-client.o  backdoor-server.o  main.c              pktpcap.c  utils.c

[root@DataComm src]# ls
backdoor-client.c  backdoor-server.c  isaac_encryption.o  main.h  pktpcap.h  utils.c
backdoor-client.h  backdoor-server.h  lib                 Makefile  pktpcap.o  utils.h
backdoor-client.o  backdoor-server.o  main.c              pktpcap.c  test.txt   utils.o
[root@DataComm src]#
```

As seen in the screenshots above, the file test.txt was created on the server.

Test Case #5

In this test case, we will check to see that the results of a command, 'ls', is returned to the client.

```
[root@DataComm src]# ../exe/client -a 192.168.0.2
Enter a password:
Enter a command: touch test.txt
Sending data: uestlonQ? 0 cmd[touch test.txt]cmd

Enter a command: ls -l
Sending data: uestlonQ? 0 cmd[ls -l]cmd
total 148
-rw-r--r-- 1 root root 5418 Oct 5 23:44 backdoor-client.c
-rw-r--r-- 1 root root 480 Oct 4 21:24 backdoor-client.h
-rw-r--r-- 1 root root 13376 Oct 6 00:20 backdoor-client.o
-rw-r--r-- 1 root root 4656 Oct 6 00:19 backdoor-server.c
-rw-r--r-- 1 root root 417 Oct 5 23:06 backdoor-server.h
-rw-r--r-- 1 root root 10616 Oct 6 00:20 backdoor-server.o
-rw-r--r-- 1 root root 7264 Oct 6 00:20 isaac_encryption.o
drwxr-xr-x 2 root root 4096 Oct 5 21:15 lib
-rw-r--r-- 1 root root 991 Oct 4 19:06 main.c
-rw-r--r-- 1 root root 187 Oct 4 19:06 main.h
-rw-r--r-- 1 root root 1016 Oct 4 19:06 Makefile
-rw-r--r-- 1 root root 8449 Oct 5 23:44 pktcap.c
-rw-r--r-- 1 root root 2716 Oct 4 20:17 pktcap.h
-rw-r--r-- 1 root root 18848 Oct 6 00:20 pktcap.o
-rw-r--r-- 1 root root 0 Oct 6 00:23 test.txt
-rw-r--r-- 1 root root 10757 Oct 5 23:40 utils.c
-rw-r--r-- 1 root root 1086 Oct 5 23:44 utils.h
-rw-r--r-- 1 root root 19240 Oct 6 00:20 utils.o
```

As seen in the screenshot above, the results of ls -l is printed on the client.