

Tests

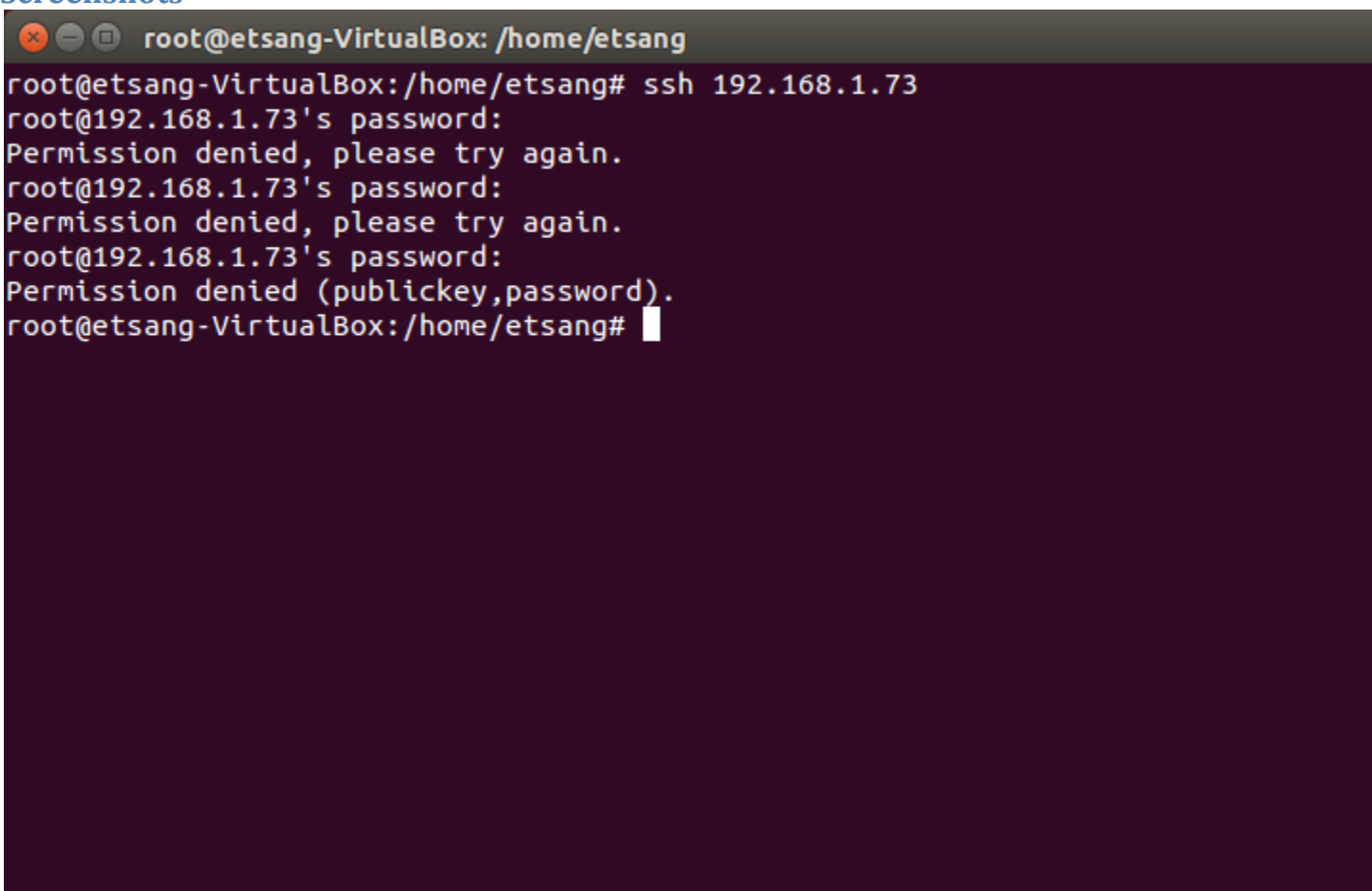
Test #	Test procedure / description	Expected Outcome	Actual Outcome & Remarks	Pass / Fail
1	<ol style="list-style-type: none">on a client machine, SSH into the server machineinput an invalid password 3 or more timescheck the security logs	<ul style="list-style-type: none">entries in the security log should show that there were password failed attempts	As expected	Pass
2	<ol style="list-style-type: none">enter "crontab -r" into a terminaluse the setjob.sh script to add the ips script as a cronjob with max attempts set to 3check the crontab file	<ul style="list-style-type: none">the crontab file should have an entry in it for regularly running the IPS script	As expected	Pass
3	<ol style="list-style-type: none">enter "crontab -r" into a terminaluse the setjob.sh script to add the ips script as a cronjob with max attempts set to 3try to log in 3 or more times from a client with invalid passwordsawait the ips' executioncheck the database, and iptables	<ul style="list-style-type: none">In iptables, it should have an entry to drop all packets from the client's IP address.The database should have an entry with the client's IP address, number of failed attempts, and timestamp of the last attempt.	As expected	Pass
4	<ol style="list-style-type: none">enter "crontab -r" into a terminaluse the setjob.sh script to add the ips script as a cronjob with max attempts set to 3try to log in 3 or more times from a client with invalid passwordsawait the ips' executionattempt to connect via SSh to the server again	<ul style="list-style-type: none">the SSH client should hang as it is trying to connect to the remote host	As expected	Pass
5	<ol style="list-style-type: none">enter "crontab -r" into a terminaluse the setjob.sh script to add the ips script as a cronjob with max attempts set to 3try to log in 3 or more times from a client with invalid passwordsawait the ips' executionwait for the user-specified time to elapse for unbanning an ip addresscheck the database, and iptables	<ul style="list-style-type: none">the database file should no longer have an entry with the client's IP addressin iptables, there should be the original rule dropping all packets from the client's IP address, but it should be preempted by a new rule that accepts all packets from that IP address.	As expected	Pass
6	<ol style="list-style-type: none">enter "crontab -r" into a terminaluse the setjob.sh script to add the ips script as a cronjob with max attempts set to 3try to log in 3 or more times from a client with invalid passwordsawait the ips' execution	<ul style="list-style-type: none">the SSH client should successfully connect, and prompt the user for a password	As expected	Pass

Test #	Test procedure / description	Expected Outcome	Actual Outcome & Remarks	Pass / Fail
	5. wait for the user-specified time to elapse for unbanning an ip address 6. attempt to connect via SSh to the server again			
7	1. enter "crontab -r" into a terminal 2. use the setjob.sh script to add the ips script as a cronjob 3. try to log in 2 or times from a client with invalid passwords 4. await the ips' execution 5. check the database file and security logs	<ul style="list-style-type: none"> the security logs should show some password failure attempts the database should have an entry for the client with its IP address, 2 for attempted logins, and a timestamp of its last login the SSH client shows that it has failed to login twice there should be no rules in the iptables for the client 	As expected	Pass
8	1. enter "crontab -r" into a terminal 2. use the setjob.sh script to add the ips script as a cronjob 3. try to log in 2 or times from a client with invalid passwords 4. await the ips' execution 5. check the database file and security logs 6. log in successfully with the correct password 7. await the ips' execution 8. check the database file	<ul style="list-style-type: none"> the security logs should show some password failure attempts as well as a successful login attempt the database should have no entry for the client the SSH client should connect there should be no rules in the iptables for the client 	As expected	Pass

Note

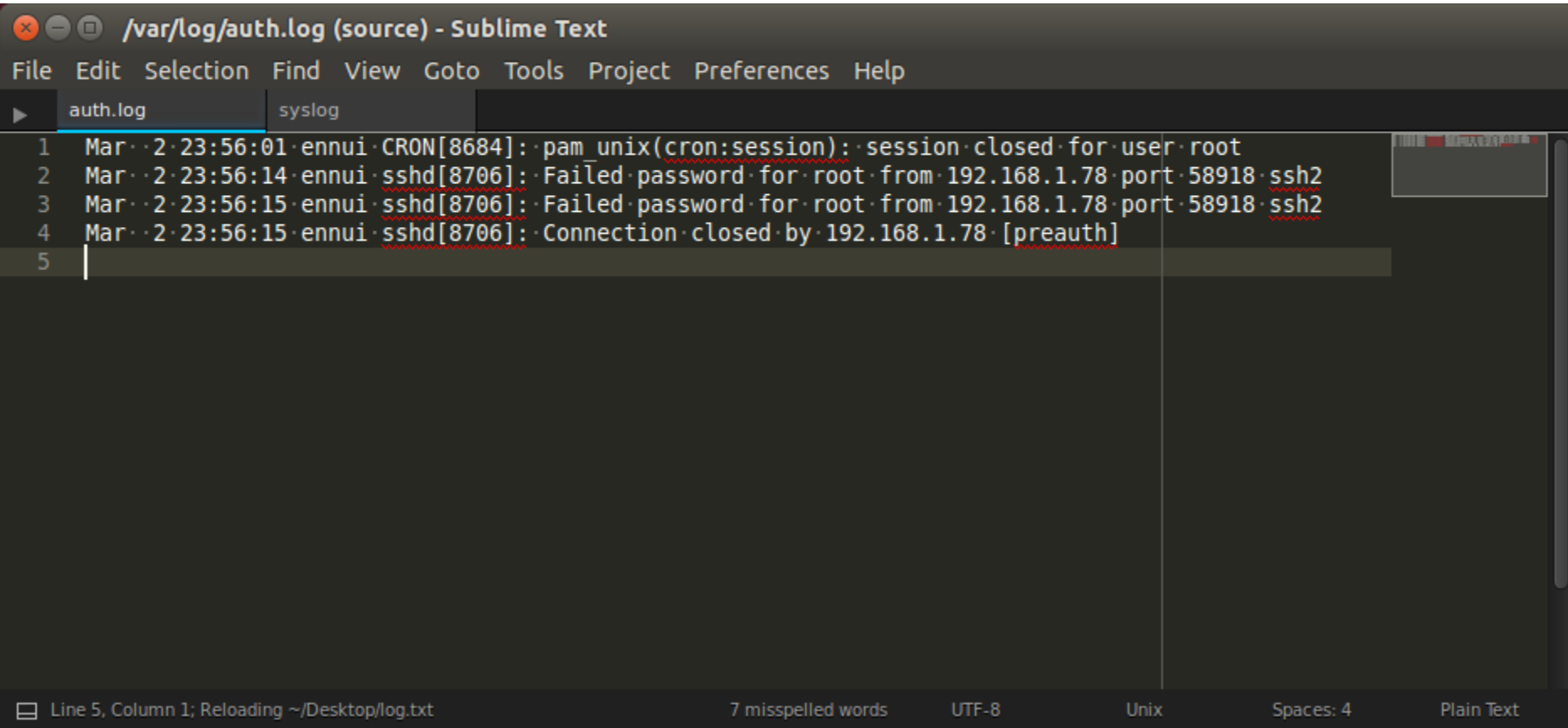
the tests here show only the IPS working when monitoring the ubuntu auth.log, however it has proven to work on Fedora 22's /var/log/messages as well as its /var/log/secure.

Screenshots

A terminal window titled 'root@etsang-VirtualBox: /home/etsang' with standard window controls. The terminal shows a user attempting to SSH into the IP 192.168.1.73. The first two password attempts are rejected with the message 'Permission denied, please try again.' The third attempt is rejected with the message 'Permission denied (publickey,password).' The prompt returns to the user's shell.

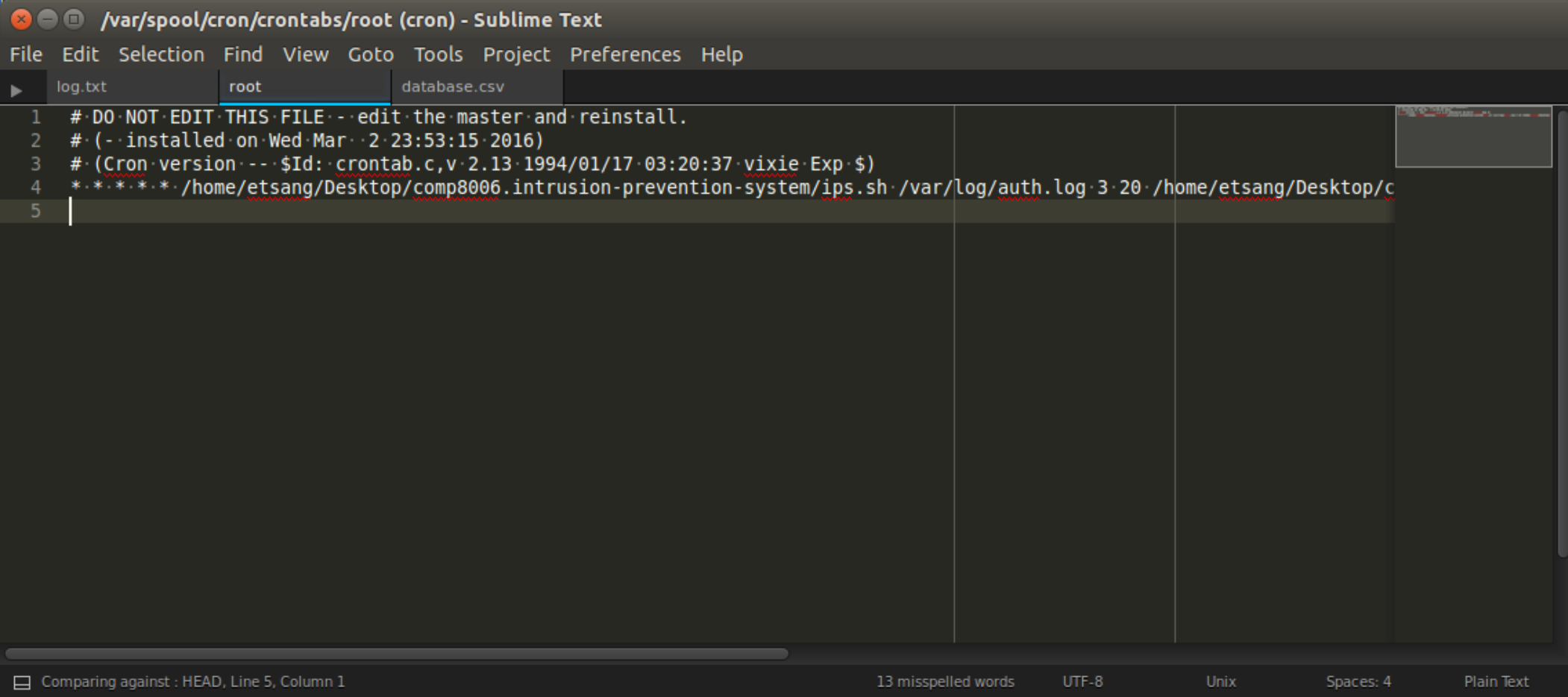
```
root@etsang-VirtualBox: /home/etsang# ssh 192.168.1.73
root@192.168.1.73's password:
Permission denied, please try again.
root@192.168.1.73's password:
Permission denied, please try again.
root@192.168.1.73's password:
Permission denied (publickey,password).
root@etsang-VirtualBox: /home/etsang#
```

Figure 1 Test 1, client is SSHing into the server with invalid passwords 3 times



```
File Edit Selection Find View Goto Tools Project Preferences Help
auth.log syslog
1 Mar · 2 · 23:56:01 · ennui · CRON[8684] : pam_unix(cron:session) : session closed for user root
2 Mar · 2 · 23:56:14 · ennui · sshd[8706] : Failed password for root from 192.168.1.78 port 58918 ssh2
3 Mar · 2 · 23:56:15 · ennui · sshd[8706] : Failed password for root from 192.168.1.78 port 58918 ssh2
4 Mar · 2 · 23:56:15 · ennui · sshd[8706] : Connection closed by 192.168.1.78 [preauth]
5
Line 5, Column 1; Reloading ~/Desktop/log.txt 7 misspelled words UTF-8 Unix Spaces: 4 Plain Text
```

Figure 2 Test 1, security logs show that there are failed password attempts



The image shows a Sublime Text editor window titled `/var/spool/cron/crontabs/root (cron) - Sublime Text`. The menu bar includes File, Edit, Selection, Find, View, Goto, Tools, Project, Preferences, and Help. The tab bar shows three open files: `log.txt`, `root` (which is the active file), and `database.csv`. The editor content shows a crontab file for the `root` user. The first four lines are comments: Line 1: `# DO NOT EDIT THIS FILE - edit the master and reinstall.`; Line 2: `# (.. installed on Wed Mar 2 23:53:15 2016)`; Line 3: `# (Cron version --- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)`; Line 4: `*.*.*.*.* /home/etsang/Desktop/comp8006.intrusion-prevention-system/ips.sh /var/log/auth.log 3 20 /home/etsang/Desktop/c`. Line 5 is empty. The status bar at the bottom displays: `Comparing against : HEAD, Line 5, Column 1`, `13 misspelled words`, `UTF-8`, `Unix`, `Spaces: 4`, and `Plain Text`.

```
# DO NOT EDIT THIS FILE - edit the master and reinstall.
# (.. installed on Wed Mar 2 23:53:15 2016)
# (Cron version --- $Id: crontab.c,v 2.13 1994/01/17 03:20:37 vixie Exp $)
*.*.*.*.* /home/etsang/Desktop/comp8006.intrusion-prevention-system/ips.sh /var/log/auth.log 3 20 /home/etsang/Desktop/c

```

Figure 3 Test 2, the crontab file for root has an entry in it for regularly running the IPS script

```
root@ennui: /var/log
root@ennui:/var/log# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
DROP       all  --  192.168.1.78          anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
root@ennui:/var/log#
```

Figure 4 Test 3, iptables shows that a new rule was appended to it, banning all traffic from the client IP address

```
~/Desktop/comp8006.intrusion-prevention-system/database.csv (cron) - Subl
File Edit Selection Find View Goto Tools Project Preferences Help
log.txt root database.csv
1 IP ADDRESS, ATTEMPTS, LAST ATTEMPT TIME
2 192.168.1.78,3,1456991821
3
Line 3, Column 1 0 misspelled words UTF-8 Unix Spaces: 4 Plain Text
```

Figure 5 Test 3, the database file shows the IP address of the client that failed to log in, how many password attempts there were, and a timestamp of their last attempt

```
root@etsang-VirtualBox: /home/etsang
root@etsang-VirtualBox:/home/etsang# ssh 192.168.1.73
root@192.168.1.73's password:
Permission denied, please try again.
root@192.168.1.73's password:
Permission denied, please try again.
root@192.168.1.73's password:
Permission denied (publickey,password).
root@etsang-VirtualBox:/home/etsang# ssh 192.168.1.73
```

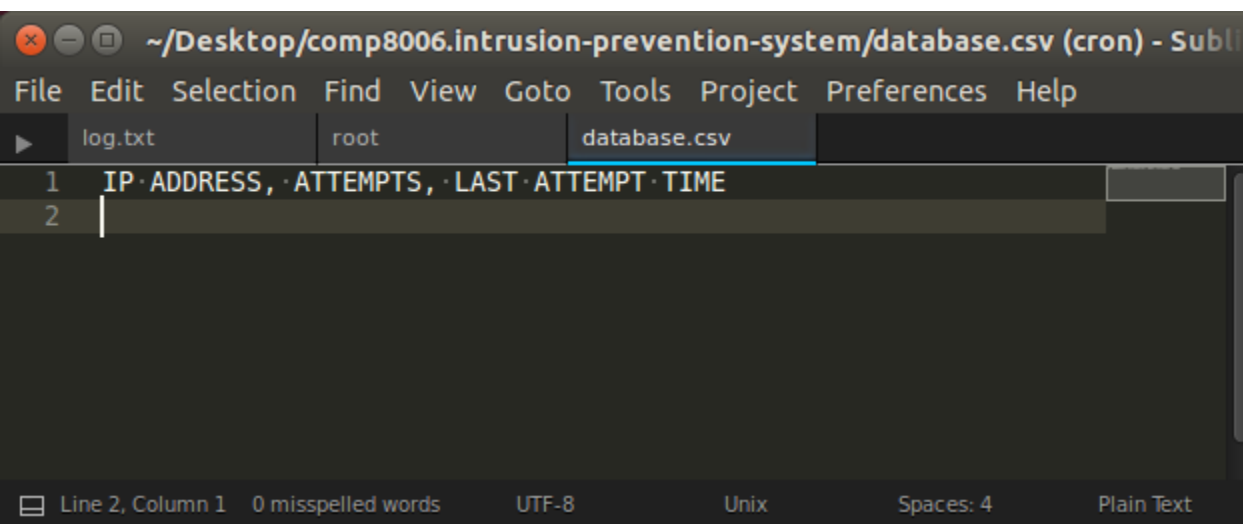
Figure 6 Test 4, SSH client hanging while trying to connect to the server, because the server has banned its IP address

```
root@ennui: /var/log
root@ennui:/var/log# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination
ACCEPT     all  --  192.168.1.78          anywhere
DROP       all  --  192.168.1.78          anywhere

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

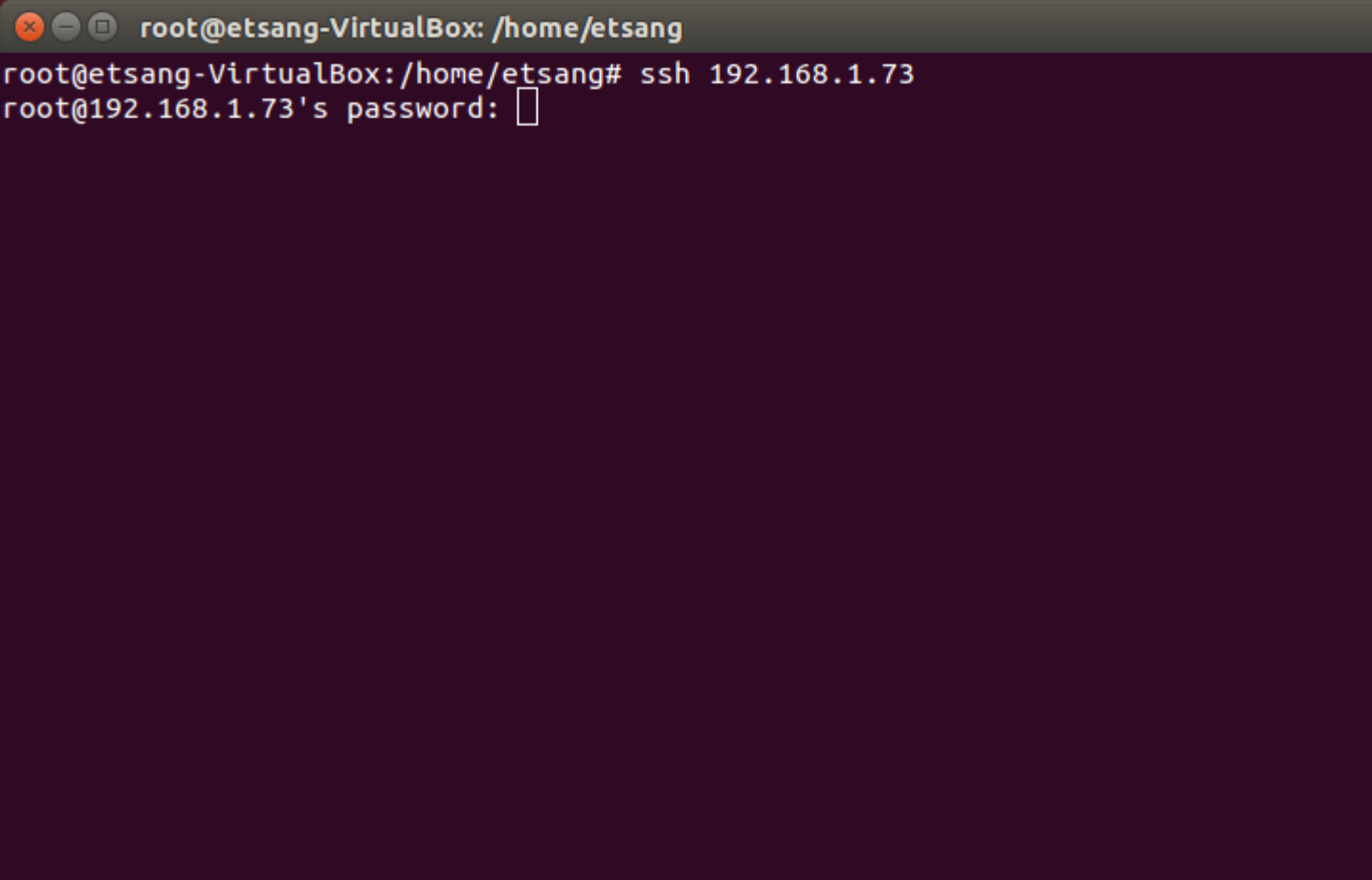
Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
root@ennui:/var/log#
```

Figure 7 Test 5, the iptables drop all packets from malicious client rule has been preempted with a rule that accepts all traffic from the client



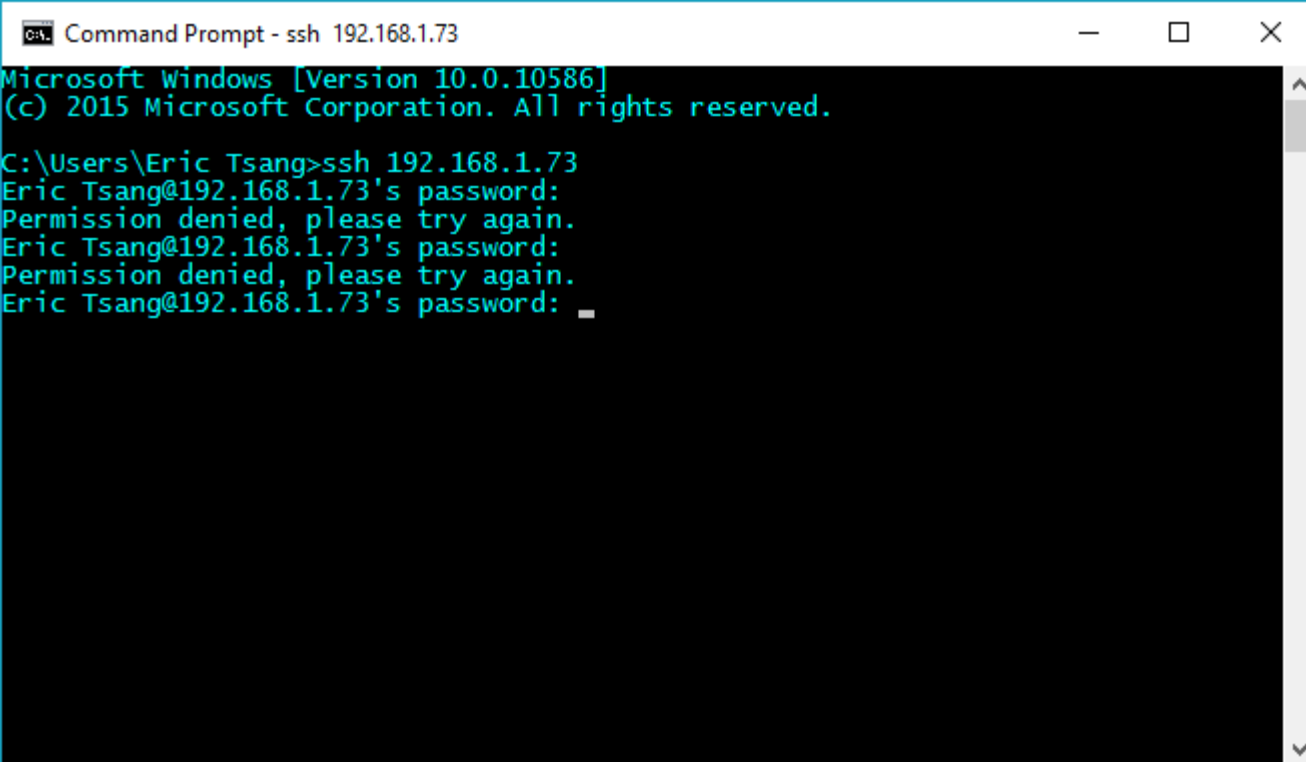
```
~/Desktop/comp8006.intrusion-prevention-system/database.csv (cron) - Subl
File Edit Selection Find View Goto Tools Project Preferences Help
log.txt root database.csv
1 IP ADDRESS, ATTEMPTS, LAST ATTEMPT TIME
2
```

Figure 8 Test 5, there is no more entry for banning the previously banned client because the ban time has elapsed

A terminal window with a dark background and light-colored text. The window title bar at the top shows three window control icons (close, minimize, maximize) followed by the text 'root@etsang-VirtualBox: /home/etsang'. The terminal content shows a user at the 'root@etsang-VirtualBox: /home/etsang' prompt typing the command 'ssh 192.168.1.73'. The next line shows the prompt 'root@192.168.1.73's password:' followed by a single white square character, likely representing a password input or a cursor.

```
root@etsang-VirtualBox: /home/etsang
root@etsang-VirtualBox:/home/etsang# ssh 192.168.1.73
root@192.168.1.73's password: □
```

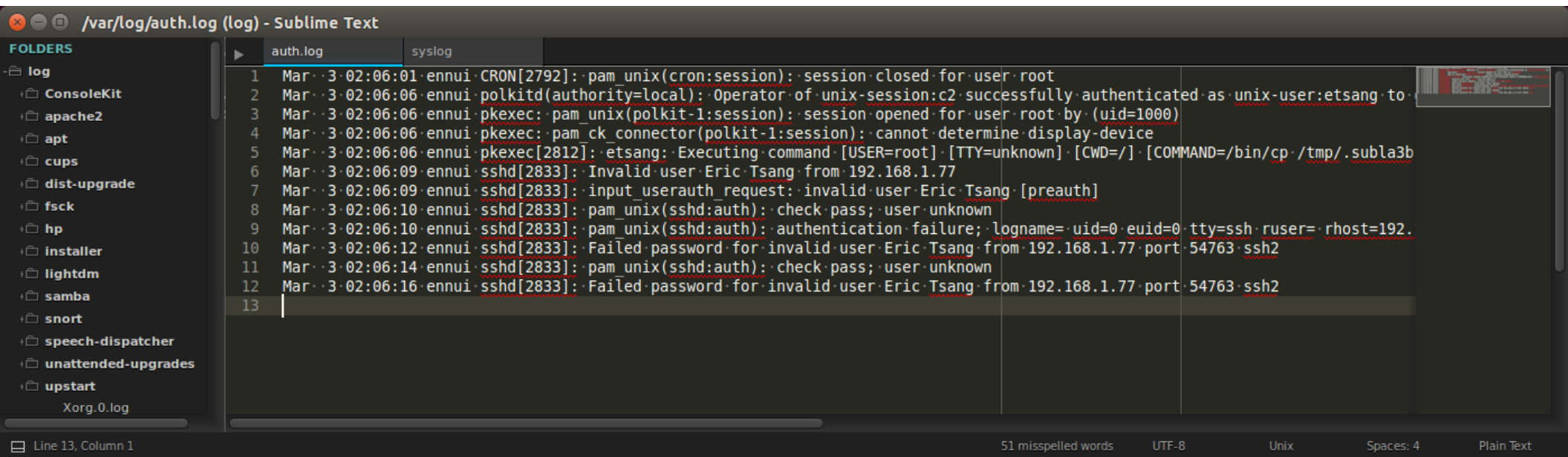
Figure 9 Test 6, the SSH client can now connect to the server after being unbanned



```
Microsoft Windows [Version 10.0.10586]
(c) 2015 Microsoft Corporation. All rights reserved.

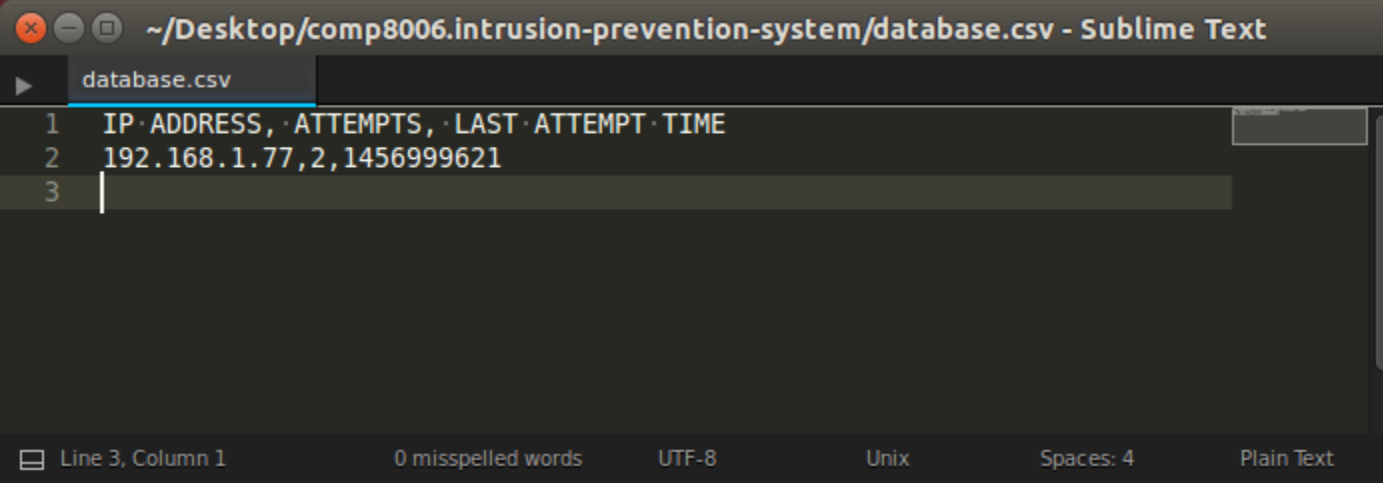
C:\Users\Eric Tsang>ssh 192.168.1.73
Eric Tsang@192.168.1.73's password:
Permission denied, please try again.
Eric Tsang@192.168.1.73's password:
Permission denied, please try again.
Eric Tsang@192.168.1.73's password: _
```

Figure 10 Test 7, SSH client fails to input valid password twice



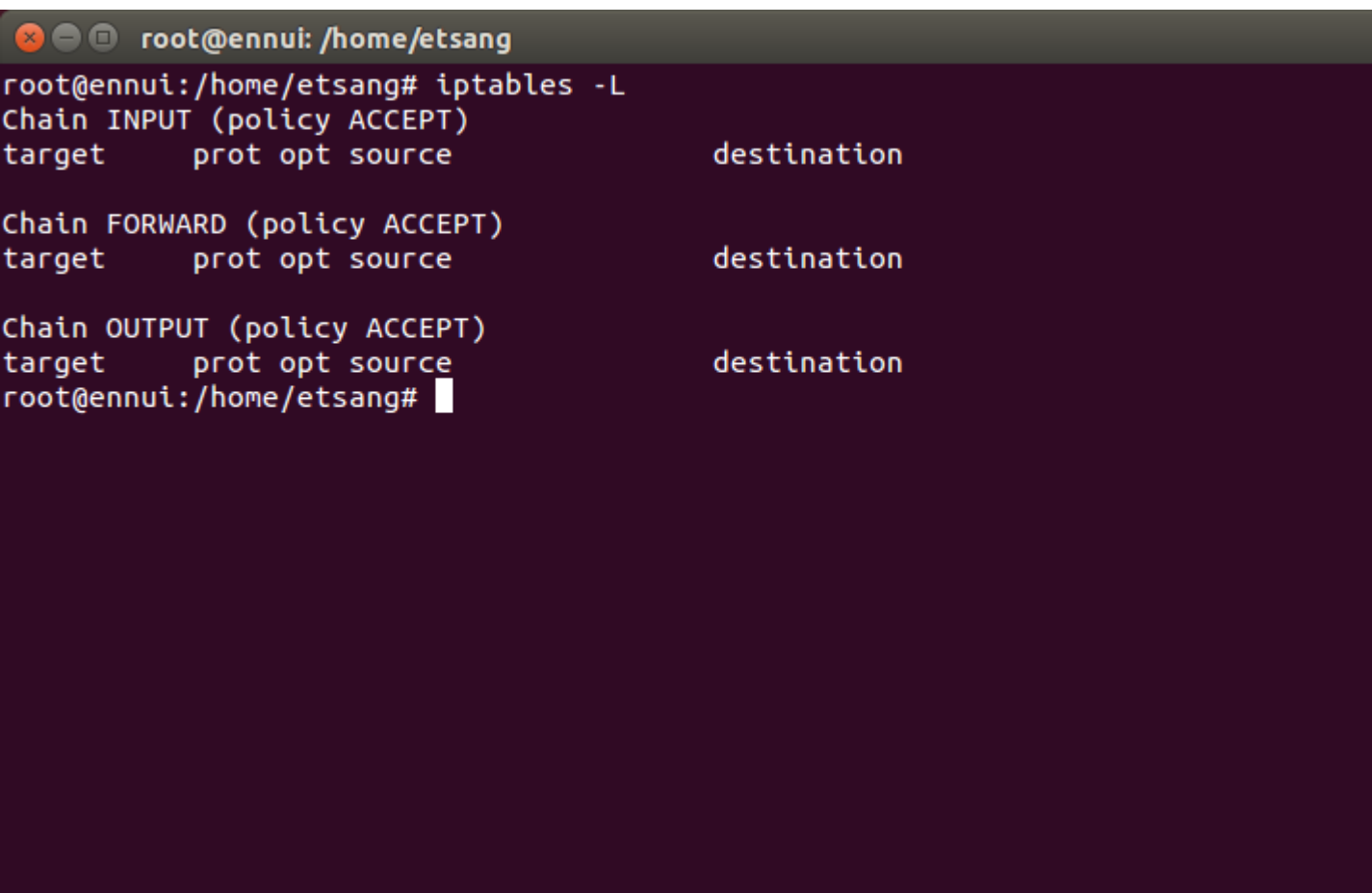
```
1 Mar  3 02:06:01 ennui CRON[2792]: pam_unix(cron:session): session closed for user root
2 Mar  3 02:06:06 ennui polkitd(authority=local): Operator of unix-session:c2 successfully authenticated as unix-user:etsang to
3 Mar  3 02:06:06 ennui pkexec: pam_unix(polkit-1:session): session opened for user root by (uid=1000)
4 Mar  3 02:06:06 ennui pkexec: pam_ck_connector(polkit-1:session): cannot determine display device
5 Mar  3 02:06:06 ennui pkexec[2812]: etsang: Executing command [USER=root] [TTY=unknown] [CWD=/] [COMMAND=/bin/cp /tmp/.subla3b
6 Mar  3 02:06:09 ennui sshd[2833]: Invalid user Eric Tsang from 192.168.1.77
7 Mar  3 02:06:09 ennui sshd[2833]: input_userauth_request: invalid user Eric Tsang [preauth]
8 Mar  3 02:06:10 ennui sshd[2833]: pam_unix(sshd:auth): check pass; user unknown
9 Mar  3 02:06:10 ennui sshd[2833]: pam_unix(sshd:auth): authentication failure; logname= uid=0 euid=0 tty=ssh ruser= rhost=192.
10 Mar  3 02:06:12 ennui sshd[2833]: Failed password for invalid user Eric Tsang from 192.168.1.77 port 54763 ssh2
11 Mar  3 02:06:14 ennui sshd[2833]: pam_unix(sshd:auth): check pass; user unknown
12 Mar  3 02:06:16 ennui sshd[2833]: Failed password for invalid user Eric Tsang from 192.168.1.77 port 54763 ssh2
13
```

Figure 11 Test 7, authentication logs from the server showing 2 invalid login attempts



```
1 IP ADDRESS, ATTEMPTS, LAST ATTEMPT TIME
2 192.168.1.77,2,1456999621
3
```

Figure 12 Test 7, ips notes that the client has tried to log on twice, but failed



```
root@ennui: /home/etsang
root@ennui:/home/etsang# iptables -L
Chain INPUT (policy ACCEPT)
target    prot opt source                destination

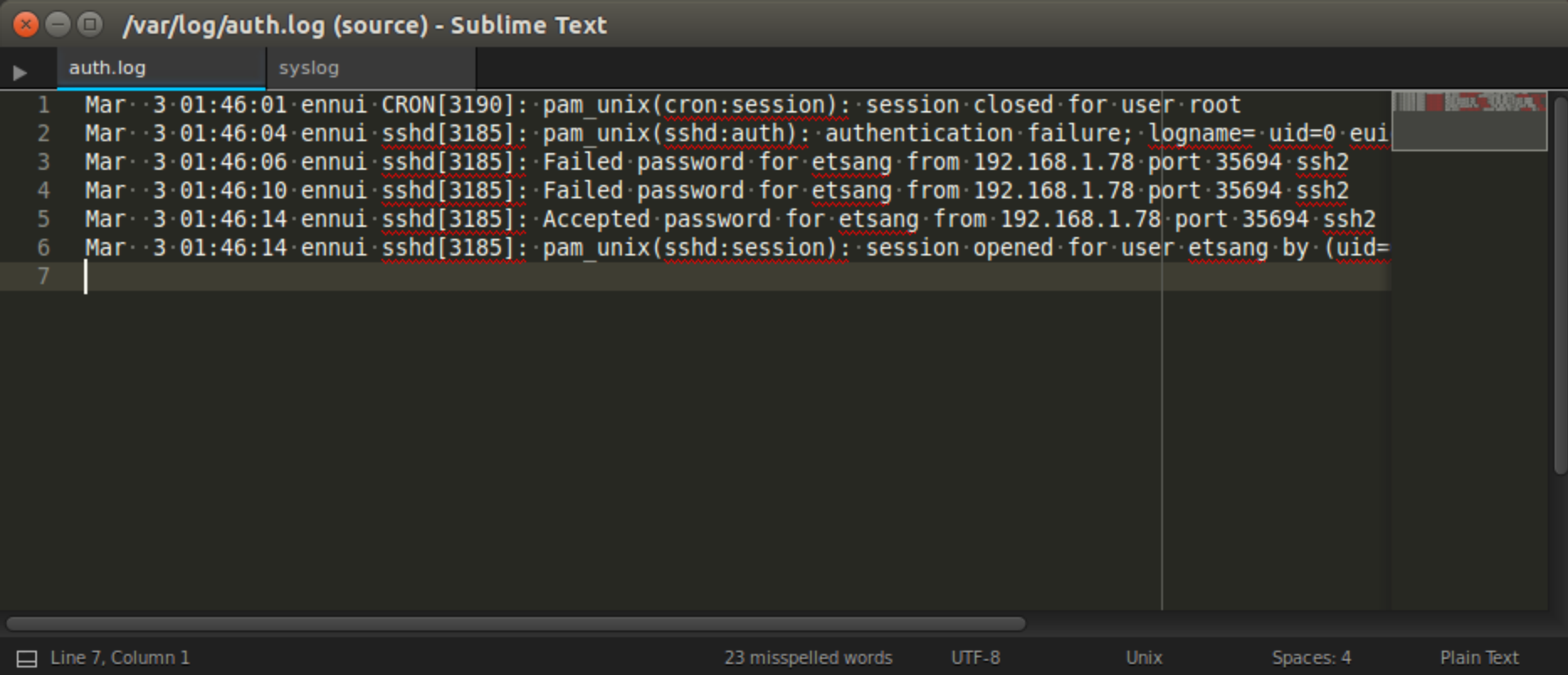
Chain FORWARD (policy ACCEPT)
target    prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target    prot opt source                destination
root@ennui:/home/etsang#
```

Figure 13 Test 7, no iptables entry is added to ban the client yet, because 2 attempts is still below the threshold

```
etsang@ennui: ~  
etsang@etsang-VirtualBox:~$ ssh 192.168.1.73  
etsang@192.168.1.73's password:  
Permission denied, please try again.  
etsang@192.168.1.73's password:  
Permission denied, please try again.  
etsang@192.168.1.73's password:  
Welcome to Ubuntu 14.04.3 LTS (GNU/Linux 3.16.0-60-generic x86_64)  
  
* Documentation:  https://help.ubuntu.com/  
  
The programs included with the Ubuntu system are free software;  
the exact distribution terms for each program are described in the  
individual files in /usr/share/doc/*/copyright.  
  
Ubuntu comes with ABSOLUTELY NO WARRANTY, to the extent permitted by  
applicable law.  
  
etsang@ennui:~$
```

Figure 14 Test 8, SSH client logs in successfully after 2 invalid passwords



```
1 Mar  3 01:46:01 ennui CRON[3190]: pam_unix(cron:session): session closed for user root
2 Mar  3 01:46:04 ennui sshd[3185]: pam_unix(sshd:auth): authentication failure; logname=uid=0 eui
3 Mar  3 01:46:06 ennui sshd[3185]: Failed password for etsang from 192.168.1.78 port 35694 ssh2
4 Mar  3 01:46:10 ennui sshd[3185]: Failed password for etsang from 192.168.1.78 port 35694 ssh2
5 Mar  3 01:46:14 ennui sshd[3185]: Accepted password for etsang from 192.168.1.78 port 35694 ssh2
6 Mar  3 01:46:14 ennui sshd[3185]: pam_unix(sshd:session): session opened for user etsang by (uid=
7
```

Line 7, Column 1 23 misspelled words UTF-8 Unix Spaces: 4 Plain Text

Figure 15 Test 8, security logs show that there were two failed login attempts followed by a successful one

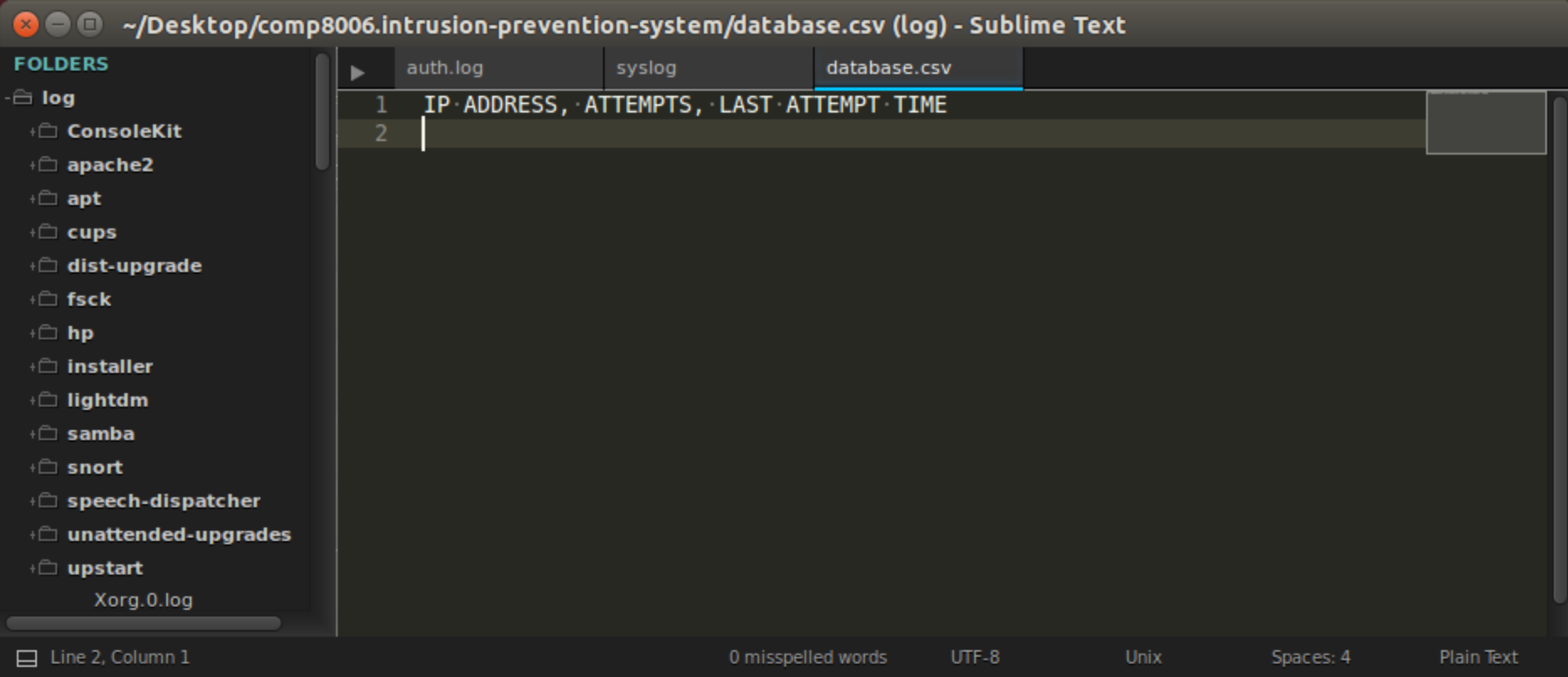


Figure 16 Test 8, there is no recorded in the database for the client because it logged in successfully

```
root@ennui: /home/etsang
root@ennui:/home/etsang# iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source                destination

Chain FORWARD (policy ACCEPT)
target     prot opt source                destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source                destination
root@ennui:/home/etsang#
```

Figure 17 Test 8, there is no entry in the iptables for the client because it is still below the threshold for being banned