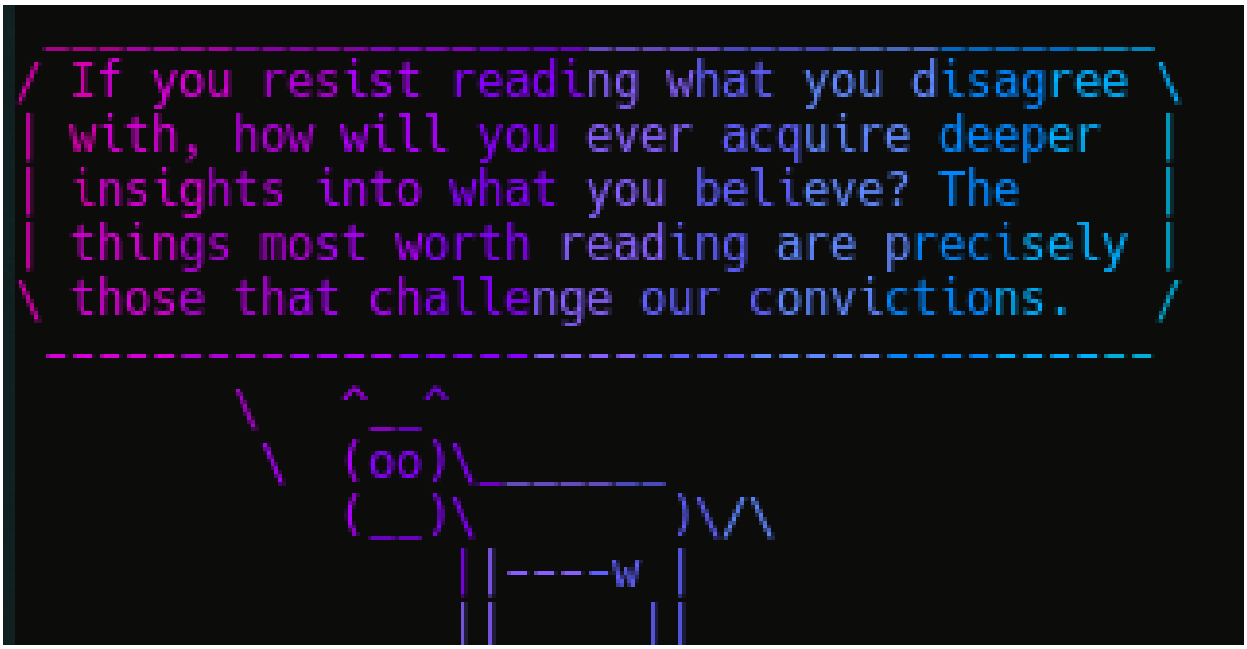
Identify the vulnerability	What harm could it do?	How to fix it
1. There is a user called backdoor with no password that has the same gid and uid as root. There are also similar users jry and toor that have roots gid and uid but have weak passwords.	These users all become root upon login and can access anything. They can literally modify and read any file that they want. They can destroy the machine at any moment, make it impossible to login, steal all the credentials of users for the system or mysql. They have and control everything on the system!	Just remove the unneeded users like backdoor and recreate the needed users to have their own uid/gid, so they don't become root at login. To fix the bad passwords, enforce better passwords like longer passwords and to prevent dictionary attacks when creating users. (You can also not delete the needed users and just modify the IDs. But I think it is probably easier to recreate them.)
2. File permissions on /etc/shadow allow anyone to read hashes of user passwords	This allows any unprivileged user on the machine to read and copy the hashes of users' passwords. They can put these hashes through a system like John the Ripper, and with a little amount of time they will have the passwords to all users. (I did this on the desktop computer and it cracked all the passwords in less than a day.)	Modify the permissions of the file to only allow the user root to read and write and group shadow to read. (May also be for shadow backup if that is also misconfigured) Command: # chmod 640 /etc/shadow
3. Web Page on http is vulnerable to SQL injection through its user authentication and XSS in the message board	This allows anyone to bypass authentication and be any user they want on the system. They then can set up a script to execute on the system to gain information on the machine or have it embed a script to run anytime a user accesses the message board. This could steal the users cookies and give an attacker access to that person's account.	Reprogram the backend of the web page to sanitize the inputs it gets from the login page and message board text boxes. It should prevent the use of sql commands, quotations, logic statements like OR that rework logic, and characters such as <, /, and > that are used by html tags.
4. SSH allows root login, empty passwords and the use protocol v1	This allows anyone to attempt to connect to ssh as a root user or a user with no password. These are insecure as it can give people unlimited access, an easy way onto the machine or both (ie the backdoor user). As well, a user could connect via v1 of ssh which isn't secure anymore. The encryption can easily be broken, so it's almost as if it uses plaintext.	Modify the /etc/ssh/sshd_config file to disable the options by assigning no to PermitEmptyPasswords and PermitRootLogin. As well, modify the Protocol option to only be 2 and not 1,2 or 2,1 in sshd_config.

5. Unencrypted telnet, http, ftp, smtp, finger, mysql, mongod services are open	These connections are plaintext to anyone that can view them (ie just about anyone between the capstone machine and the client.) This means people can just easily read the login credentials to login on the web server or connect to ftp, telnet, mysql or mongodb. They can then can use those themselves to gain unauthorized access to the services.	Set up credentials and enable SSL/TLS to apache, postfix, vsftpd, mysql, mongodb. Telnet can be closed in replacement of only using SSH and FTP can also be replaced for SFTP. As well, another option is set up ssh tunneling / port forwarding for ftp, mysql, and mongodb. If finger and telnet are unneeded, then they can be disabled by deleting their files in /etc/xinetd.d/.
6. Files uploaded via ftp by anonymous user (which has no password) are modified to be owned by the root user	This allows anyone to upload a file, and it gets chowned to become root. I think this means that any file a user uploads is impossible to remove except by a root or a wheel/sudoer. However, the directory does have to allow the ftp user or group or the world to write to it.	Modify the /etc/vsftpd.conf file to disable the misconfigurations by setting anonymous_enable to NO, anon_upload_enable to NO, anon_mkdir_write_enable to NO, and chown_uploads to NO.
7. There is an open port at 1337 ie “LEET” that executes any line of text sent to it as a bash command as the root user	This allows anyone who connects to this port to send any command they want to be run and the output will be sent back. They can remotely gain access to any file or execute any command. This is equally as dangerous and has the same possibilities if someone used the backdoor user except it can be done remotely.	Disable and delete the evil.service that initializes the opening of the port with netcat. Commands: # systemctl disable evil(.service) # rm /lib/systemd/system/evil.service

Don’t grade this:

Things I found but don’t know if they are a vulnerability, something odd, or just an issue I had
1. Remounting and binding of / and /etc under /srv/ftp so the anon ftp user has quick access
2. Snake oil cert (.pem and .key) in /etc/ssl that is set for ftp and postfix but not enabled
3. Old Linux version and kernel (capstone goes back to being a machine from 2015?)
4. Mysql doesn’t allow me to login as root so I can’t check the configuration or allowed users. I suspected that it would allow a remote user with no password or something.
5. Don’t know how to access mongoDB or understand its configuration all that much.


If you resist reading what you disagree with, how will you ever acquire deeper insights into what you believe? The things most worth reading are precisely those that challenge our convictions.
^__^
(oo)_____
(__)\ )\/\
||----w |
|| ||