Date Stamp:

The University of Nottingham

UNITED KINGDOM · CHINA · MALAYSIA

SCHOOL OF COMPUTER SCIENCE

COURSEWORK COVER SHEET

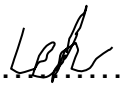
PLEASE COMPLETE THE FOLLOWING IN CAPITAL LETTERS

Module……COMP3028 UNMC ……. Coursework No……1……..……..

SURNAME……Loh…… FORENAME……Jin Xian……

Student ID……016763……. Course ……Computer Science…….

CS Username……khcy6ljx…… IS username……..

Tutor……Dr Marina Ng…… Student's Signature……  ……...

Markers Comments:

Grade:

THANK YOU FOR YOUR COURSEWORK SUBMISSION. YOUR RECEIPT WILL BE ISSUED AUTOMATICALLY VIA CS EMAIL.

QUESTION 1: PASSWORDS

In this day and age, when a user is using a new online service, there is a high chance that the user might need to create a new account with a strong password. Strong password policies are needed to protect the user from system breaches and further attack. Therefore, it is the responsibility of the system administrator to enforce password policies that will protect the users while still maintaining the ease of use of the system. After some serious thoughts, below are the password policies or additional authentication that I will implement for the network:

1. **Minimum Password Length Policy**

   This policy requires the user to have a password with the minimum length for example 8 characters. With longer passwords, it will take more time and effort for the password to be brute force. Hence, protecting the user's account.

2. **Maximum Password Age Policy**

   Just like UNM Moodle service, this policy requires the user to change his or her password after some time. As for this example, a timeframe of 180 days would be used which mean the user will change the password every 6 months, keeping the password fresh and less susceptible to the same attack.

3. **Passwords Must Meet Complexity Requirements Policy,**

   One of the most common yet effective policies to be enforced to the users. This policy requires the user to have a more complex password by combining lowercase and uppercase letters, numbers and at least 1 special symbol, once again preventing easy brute-force attack in combination with minimum password length policy.

4. **Password History Policy**

   This policy restricts the user from using old passwords for a certain time. As for this example, the user is prohibited from reusing the last 10 passwords. This ensures the user doesn't recycle the same passwords over and over again during the password changing phase.

5. **The maximum password entered attempts**

   This policy restricts the number of times a user can type in the wrong passwords. As for this case, having 5 attempts limit with a buffer time after 5 attempts is ideal to prevent brute-force attack.

When the user hit the attempts limit, the user is prohibited to access the account for 5 minutes. If the user hit the attempts limit again, the account will be locked for another 10 minutes. Therefore, the buffer time is increased for every 5 wrong attempts. Moreover, an account can also be locked permanently if 50 wrong attempts are made and the only way to unlock the account is to contact the system administrator.

6. **Connect the User's Phone Number to the Account upon Signup**

   Besides having a password, an additional way to authenticate an account is to connect the user's phone number to his account upon signup. A shortcode will be sent to the user via SMS to verify that the user is not a bot, at the same time having an extra layer of security for his account. If the user forgets his password, he can use his phone number to login instead.

(498 words)

QUESTION 2: FIREWALLS

Why does an administrator want to block ports from normal traffic? One of the reasons is that the administrator might know and assure that there will be no file transferring happening at the ports. By closing the ports, it will decrease the chances of someone exploiting it for selfish purposes. Other than that, the administrator can have better control over deciding which ports can be reachable on a certain IP using a certain protocol. Moreover, closing ports stops users from running their own servers to do things that shouldn't be allowed on the network like gaming etc.

SSH also is known as a secure shell, is a network communication protocol that enables 2 computers to communicate and share data. The inherent feature of SSH is that all communication between computers is encrypted, which is very useful for use in insecure networks since the attackers are not able to gain access to the information within the traffic flow. SSH is also an alternative to non-protected login protocols like telnet and insecure file transfer methods such as FTP.

To circumvent firewall restrictions using SSH, it can be done in 3 ways which are local or remote port forwarding and dynamic port forwarding. For local remote port forwarding, the user needs to create a tunnel through a server that is not on the network, then access websites from the tunnelled server. As for remote port forwarding, the user will need another computer that is publicly accessible and have SSH access to it. After that, use SSH to create a tunnel that opens a port on a chosen server on the internet, and connects it to a local port on the user's machine. As for dynamic port forwarding, the user is allowing a full range of TCP communication across a range of ports where SSH will act as a SOCKS proxy, causing the requests to be handled first before being directed to the remote computer. Thus, dynamic port forwarding over SSH is like a simple alternative to VPN (a virtual private network).

An example of a time when someone might use SSH tunnelling is when the user wants to take control of remote workstations and access file remotely. All these are done via SSH tunnelling because it is more secure than the traditional Telnet, preventing eavesdroppers to get hold of sensitive information. As for using SSH tunnelling for disreputable purposes, it can also be used for malicious purposes like data exfiltration because it is invisible to most network monitoring and traffic filtering solutions. SSH tunnelling can be used to hide the source of attack by bouncing attacks off systems and devices where SSH port forwarding are allowed. If there is a present of unmanaged SSH keys, an attacker can use SSH tunnelling to utilize stolen SSH keys for an intranet from the public internet.

(466 words)

QUESTION 3: SERVER SECURITY

After running and testing the vulnerable Ubuntu server based on the labs, the following section will show a few actions and how-to-guide that need to be taken in order to improve the overall security of the server.

1. **Disable unused ports**

   As seen in question 2, open ports are the gateway for both hackers and attackers to exploit. Thus, scanning and disabling ports that will not be used is a good way to further secure the server.

   How:

   - Scan for open ports using Nmap (Network Mapper)
   - For example: to close port number 3000, type this command in the terminal
     `sudo kill $(sudo lsof -t -i:3000)`

2. **Disable IPv6**

   IPv6 is often used by hackers to send malicious files over the traffic. This is possible as IPv6 malware can assign its own IP address without a DHCP (dynamic host configuration protocol) server, which gives the malware author ability to send crafted IPv6 packets capable of bypassing perimeter established security. Therefore, disabling IPv6 is a simple way to boost Linux server security.

   How:

   - Edit “sysctl.conf” file at /etc/sysctl.conf
   - Add the following lines at the bottom of the file
     `net.ipv6.conf.all.disable_ipv6 = 1`
     `net.ipv6.conf.default.disable_ipv6 = 1`
     `net.ipv6.conf.lo.disable_ipv6 = 1`

   - Run `sudo sysctl -p` or reboot the server

3. **Turn on UFW (Uncomplicated Firewall)**

   By having the firewall turned on, we can filter traffic and allow access from trusted IP address and ports only, prohibiting attackers from using open unused ports to attack the server. UFW is shipped with Ubuntu but it is not enabled by default.

   How:

   - Enter the command sudo ufw enable in the terminal
   - Enter command sudo ufw status to check whether UFW is enabled
   - Only allow ports that will actually be used. For example port 80, by entering the command sudo ufw allow 80

4. **Prevent IP spoofing**

   An attacker that uses IP spoofing will modify the source address of a packet header to make the receiving computer system thinks it is from a trusted source and accept it. By preventing IP spoofing, we can better protect the server from Denial-of-service (DoS) attacks that utilized a spoofed IP address.

   How:

   - Edit “host.conf” file at /etc/host.conf
   - Add the following lines at the end of the file and save the changes
     order bind,hosts
     nospoof on

5. **Check for rootkits**

   Rootkits is a collection of computer software that enable access to a computer that is not allowed in the first place by masking its existence or existence of other software. These types of software are often malicious and difficult to detect as a rootkit may be able to subvert the software that is intended to find it. Hence, rootkit detection software like RKHunter is installed to scan for rootkits, backdoors and possible local exploits.

   How:

   - In the terminal, type sudo apt-get install rkhunter
   - Update and run RKHunter program by typing
     sudo rkhunter --update

sudo rkhunter --propupd

sudo rkhunter –check

6. **Secure shared memory**

Shared memory allows database server threads and processes to share data by sharing access to polls of memory. However, it can also be used in an attack against running service. Instead of mounting the shared memory as read/write, we can modify it to be read-only, which can prevent running services to be changed or modified.

How:

- Edit “fstab” file at /etc/fstab
- Add the following lines at the bottom of the file

  tmpfs /dev/shm tmpfs defaults,noexec,nosuid 0 0
- Reboot server

7. **Disable Open DNS Recursion and Remove Version Info - BIND DNS Server**

DNS recursion enables the server to find the website in question in its local cache. If no answers are found, it will other DNS servers until the address is successfully found. This type of DNS request is vulnerable to fake requests from a spoofed IP address. By turning off recursive DNS lookups, the DNS server is susceptible to attackers in part of the amplified attack on a victim.

How:

- Edit “named.conf.options” file at /etc/bind/named.conf.options
- Add the following lines

  recursion no;

  version "Not Disclosed";
- Restart BIND DNS server by entering the following command

  sudo /etc/init.d/bind9 restart

8. **Disable SSL v3 support in Apache**

SSL (Secure Socket Layer) was created by Netscape but it is less secured than the newly released TLS (Transport Layer Support) protocols. Since TLS is more secure, it is better to force Apache that is inside Ubuntu 12.04 to use the newer protocol.

How:

- Edit “ssl.conf” file at /etc/apache2/mods-available/ssl.conf
- Change the line

  SSLProtocol all -SSLv2

  to

  SSLProtocol all -SSLv2 -SSLv3
- Save the file and restart Apache server by entering this command

  sudo /etc/init.d/apache2 restart

9. **Restrict Apache Information Leakage**

By default, most pre-packaged Apache installations come with full information leakage like the Apache version, the Linux distro being used, and the PHP version. By securing this information, attackers will have lesser information on how to attack our server.

How:

- Edit “security” file at /etc/apache2/conf.d/security
- Add the following lines and save the changes

  ServerTokens Prod

  ServerSignature Off

  TraceEnable Off

  Header unset ETag

  FileETag None
- Restart Apache server once again by entering the command

  sudo /etc/init.d/apache2 restart

10. Upgrade to the latest LTS (Long Term Support) version of Ubuntu

Since the server being tested is on Ubuntu 12.04, the Linux distro has already reached the end of life and no security maintenances were provided anymore, causing the server susceptible to any new types of malicious attacks. If there is extra time to download and install a new Linux distro, it is highly advisable to upgrade the server to using Ubuntu LTS 18.04 that has the up-to-date security patches and enhancements where bugs like Meltdown and Spectre that target hardware vulnerabilities can be further prevented with tighter security control in the new distro version.

(936 words)

REFERENCES

1. https://wiki.openssl.org/index.php/SSL_and_TLS_Protocols
2. https://help.ubuntu.com/community/RootSudo
3. https://www.ostechnix.com/ubuntu-server-secure-script-secure-harden-ubuntu/
4. https://dimitar.me/dynamic-port-forwarding-with-socks-over-ssh/
5. https://www.hostinger.my/tutorials/ssh-tutorial-how-does-ssh-work
6. https://www.ssh.com/iam/ssh-key-management/
7. https://help.fasthosts.co.uk/app/answers/detail/a_id/1276/~/what-is-recursive-dns-and-why-is-it-not-recommended%3F
8. https://zvelo.com/ipv6-malware-examples-and-other-web-attacks/
9. https://www.ssh.com/malware/
10. http://www.panix.com/~ruari/censorship.html
11. https://meltdownattack.com/
12. https://www.makeuseof.com/tag/reasons-upgrade-ubuntu/
13. https://www.cmu.edu/iso/governance/guidelines/password-management.html
14. http://www.steves-internet-guide.com/tcpip-ports-sockets/
15. https://www.digitalocean.com/community/tutorials/ssh-essentials-working-with-ssh-servers-clients-and-keys