g7_report

Table of Contents

1 Services and Configuration *

2 Part 1:

Linux root password: Qni@rjeIy4Nwl6nLHmo67Wy*yDUOuY4$ynyRmFoJ%6fy3q$m%TwqOTeffo*W40vh **Linux**

student password: In&JRDhntDg5gnTlx7RKSxA8I4!*yey60fOyG4q7uRmRJ@8qi1g4ZHyEd3jQEaXC **Linux professor password:** bzoXulCOJBcGupZjKcySdp74pxO6Npyfv5XTQQv9x3pq7Yr75 **Windows Student password:** #Are there other windows account passwords, and is windows admin the same as Linux? **Windows Alice password:** $6zv7yMX4!O55j$z@Yy1h2614o37AaE7r1wY9eH6M102X7184pm!qnzJFB2f06oM **Windows Bob password:** UgG822a2y5f9K7i2slGlPNa118*99qWqFY0!3GTnPprWTq0dA13e9kH5A0E901 **NOTE:** please contact us if you use lastpass, and we can share the passwords with you to make life a little easier (because these are truly sadistic passwords) without the use of a password manager

2.1 Reasoning:

- Passwords are randomly generated using LastPass (very reputable password service) to prevent dictionary/rainbow attacks and ensure integrity of random generation
- We wanted to ensure that it was impossible to brute force a password
    - It takes 16.69 million trillion trillion trillion trillion trillion

  trillion trillion centuries to exhaustively search this password space (Assuming one hundred trillion guesses per second)

- The ease of use, and proven security benefits of using a password manager and randomly generated passwords makes the use of user created passwords inexcuseable
- If we were implimenting this assignment in the real world, these ports would be open to online brute force/dictionary/rainbow attacks (excepting the use of something like Fail2Ban), so it is a neccessity to have an extremely secure password to prevent unauthorized access to the server(s)

3 Part 2

3.1 How did you select the passwords for the user accounts on Linux and Windows

- Passwords are randomly generated using LastPass (very reputable password service) to prevent

dictionary/rainbow attacks and ensure integrity of random generation
- We wanted to ensure that it was impossible to brute force a password
    - It takes 16.69 million trillion trillion trillion trillion trillion

  trillion trillion centuries to exhaustively search this password space (Assuming one hundred trillion guesses per second)

- The ease of use, and proven security benefits of using a password manager and randomly generated passwords makes the use of user created passwords inexcuseable, as compromising a user account can give an attacker a foothold to gain higher access privileges on the server
- If we were implimenting this assignment in the real world, these ports would be open to online brute force/dictionary/rainbow attacks (excepting the use of something like Fail2Ban), so it is a neccessity to have an extremely secure password to prevent unauthorized access to the server(s)

3.2 Compare and contrast how the authentication in the web server works, ftp server, and in the tftp server

3.2.1 Web Server:

- Authentication in the Apache web server uses $\text{mod}_{\text{authsspi}}$, which is middleware

to use Microsoft's NTLM stack for authentication in the Window's encvironment/

- **NTLM:** uses a challenege response scheme for authentication
    1. The client negotiates a connection to the server
    2. The server responds with a challenge to identify the client
    3. The client responds to the challenge with an authentication message, which is most likely an encrypted or hashed version of a username and password.
    4. The server responds indicating a success or failure.

3.2.2 Ftp Server:

- Implimented by VSFTPD, which supports virtual users with PAM (pluggable authentication modules)
    - **Linux PAM:** separates the tasks of authentication into four independent management groups
        1. Account modules check that the specified account is a valid

        authentication target under current conditions. (may include conditions like account expiration, time of day, and that the user has access to the requested service)

        1. Authentication modules verify the user's identity, for example

        by requesting and checking a, in this case, password.

        1. Password modules are responsible for updating passwords, and are

        generally coupled to modules in the authentication step. may also be used to enforce strong passwords

        1. Session modules define actions that are performed at the

        beginning and end of sessions. A session starts after a user has authenticated

    - A virtual user is a user login which does not exist as a real login on the

    system in /etc/passwd and /etc/shadow file. Virtual users can therefore be more secure than real users, because a compromised account can only use the FTP server but cannot login to system to use other services such as SSH or SMTP)

3.2.3 Tftp Server:

- OpenTFTP server (and TFTP in general)includes no login or access control mechanisms, and thereby

provides no authentication of users.
    - It is accessible by

any anonymous user. It does not provide the ability to manipulate what directory the user has access to, so upon setting up a tftp connection, the user only has accesss to the contents of the specified directory

3.2.4 Compare & Contrast:

- FTP provides the ability to authenticate, and control what users can access the server contents, whereas the TFTP server only allows control of what directory a user has access to
- FTP and TFTP both provide control of what directories a user is able to access
- FTP and TFTP both do not allow remote code execution in the form of shell access
- FTP does allow the user to navigate directories within the space they are permitted to access
- Web Server relies on using third party middleware, to take advantage of the host's built in authentication scheme.

4 Part 3

TCP and UDP are the most commonly used protocals on the internet. TCP and UDP perform the same task of sending packets. However the main difference is that TCP guarantees thay all packets will reach the destination in the correct order, while UDP does not. TCP is basically the same as UDP except for the fact that it does an extra check, so that's why we think the rules between the two are compatible

4.1 IP Table rules:

#Delete all default and existing rules iptables –flush

#Refuse all inboud traffic default iptables -P INPUT DROP

#Allow all incoming SSH traffic on default ssh port 22 iptables -A INPUT -p tcp –dport ssh -j ACCEPT iptables -A OUTPUT -p tcp –sport 22 -j ACCEPT #might not need

#Allow outside users to ping servers iptables -A INPUT -p icmp –icmp-type echo-request -j ACCEPT iptables -A OUTPUT -p icmp –icmp-type echo-reply -j ACCEPT #might not need

#Allow 10.229.*.* to access ftp, default port 20 iptables -A INPUT -s 10.229.0.0/16 -p tcp –dport 20 -j ACCEPT #maybe should use port 21 #Block 10.229.100.96 iptables -A INPUT -s 10.229.100.96 -p tcp –dport 20 -j REJECT #Block 10.229.96.* iptables -A INPUT -s 10.229.96.0/24 -p tcp –dport 20 -j REJECT #block group 8 (x+1) iptables -A INPUT -s 10.229.8.0 -p tcp –dport 20 -j REJECT

#outbound iptables -A OUTPUT -s 10.229.100.96 -p tcp –dport 20 -j REJECT iptables -A OUTPUT -s 10.229.96.0/24 -p tcp –dport 20 -j REJECT iptables -A OUTPUT -s 10.229.8.0 -p tcp –dport 20 -j REJECT

#Allow 10.229.*.* to access http on ports 80, 8080 iptables -A INPUT -s 10.229.0.0/16 -p tcp –dport 80, 8080 -j ACCEPT #Block 10.229.100.97 iptables -A INPUT -s 10.229.100.97 -p tcp –dport 80, 8080 -j REJECT #Block 10.229.97.* iptables -A INPUT -s 10.229.97.0/24 -p tcp –dport 80, 8080 -j REJECT #Block group 6 (x-1) iptables -A INPUT -s 10.229.6.0 -p tcp –dport 80, 8080 -j REJECT

#Allow 10.229.*.* to access tftp, default port 69 iptables -A INPUT -s 10.229.0.0/16 -p udp –dport 69 -j ACCEPT #Block 10,229.100.96 iptables -A INPUT -s 10.229.100.96 -p udp –dport 69 -j REJECT #Block 10.229.96.* iptables -A INPUT -s 10.229.96.0/24 -p udp –dport 69 -j REJECT #block group 9(x+2) iptables -A INPUT -s 10.229.9.0 -p udp –dport 69 -j REJECT

#logging, may have to be slightly changed, following this http://stackoverflow.com/questions/21771684/iptables-log-and-drop-in-one- rule iptables -N LOGGING iptables -A INPUT -j LOGGING iptables -A OUTPUT -j LOGGING iptables -A LOGGING -m limit –limit 2/min -j LOG –log-prefix "IPTables-Dropped"

5 Delegation of duties

- All group members collaborated on all portions of the assignment, however each member assumed a leadership role/responsibiltiy for the completion of one portion of the assignment
- Justin was in charge of setting up the web server on the Windows machine, and had a secondary role configuring ip tables, as well as configuring the Linux machine to support ip tables
- Vuk was in charge of cracking the passwords for the Assignment 1 sliding portion, as well as configuring ip tables
- Mackenzie was in charge of setting up the FTP and TFTP servers on the linux machine, as well as configuring NAT, and port forwarding for the web server

6 Difficulties Encountered

- Testing the iptables to ensure proper function was difficult to manage
    - In the future, it would be good to have a lab day dedicated to testing

  to make it easy to collaborate with other groups to test eachother's setups

- Setting up the Slackware provided TFTP server proved to be impossible, and the OpenTFTP server had to be installed instead. This also proved somewhat difficult but not insurmountable
    - In the future, specifically say not to setup the Slackware TFTP server,

  but others did not have the same trouble that Mackenzie did, so perhaps he is just not very smart

- Assignment forced us to learn more about linux permissions, and how firewalls work
- The assignment took a reasonable amount of effort, but less so than Assignment 1
- The workload was reasonable

7 Resources:

https://help.ubuntu.com/community/vsftpd http://docs.slackware.com/ http://www.m0rd0r.eu/slackware-as-basic-tftp-

server/ https://en.wikipedia.org/wiki/Linux_PAM https://en.wikipedia.org/wiki/NT_LAN_Manager

Author: Mackenzie Bligh
Created: 2016-11-04 Fri 16:57
Emacs 24.5.1 (Org mode 8.2.10)
Validate