**CS140 Computer Security**

**Worksheet 4: Looking for Security Defects in Linux**

**Answer Sheet**

|  |
| --- |
| Name:  Manpreet Kaur Bahtra |
| ID number:  2208321 |

If you require extra space, please copy and paste more defect boxes. Your final submission must be submitted via Tabula by 1200 noon, Mon, 23th Jan, 2023. If you hand in late without an extension having been granted, marks will be deducted according to the standard University policy (5% per day).

|  |  |  |
| --- | --- | --- |
| Describe the defect | There is no firewall in the coursework 2 image. However, in lab 1 image under ps auxf command, firewall was present. It is a defect because without a firewall malicious data will be permitted to enter and exit the system unrestricted, hence putting the system at risk. | For marker’s use only |
| How would you correct the problem? | By enabling the firewall to ensure no malicious data is granted access to the system. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | Anacron running in coursework 2 image under ps auxf command. Virtual machines are not always on, and anacron doesn’t require the machine to be on. Anacron is used to run commands periodically. Anacron could be used to run commands to inject viruses onto the system on a routine basis. | For marker’s use only |
| How would you correct the problem? | By disabling anacron and using alternative such as cron. This limits the time that routine commands can for to only when the machine is on. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In the /etc folder, shadow file is permitted to be read and written by user and group. However, since the shadow file contains the hash, only the root should be able to see it. It is unsafe that group can read/write the hash. Hash could be decrypted to obtain the original password. | For marker’s use only |
| How would you correct the problem? | By only giving user read and write permissions. (rw………) |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In the /etc shadow file, the field of the hash function is 1 for users Adam, Gordon, Dirk. This means that MD5 is used for hashing, but MD5 has been declared not secure. | For marker’s use only |
| How would you correct the problem? | Use a stronger hash function, such as SHA-256 or SHA512. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In the /etc/sysconfig selinux file, selinux state is permissive, which means that if a program launched that doesn’t follow the selinux policy, it prints warnings and doesn’t enforce the policy. This means that only warning will be shown and malicious software would be allowed to run. | For marker’s use only |
| How would you correct the problem? | Set the selinux state to enforcing which imposes selinux security policy and doesn’t allow programs to run, which are not permitted by SELINUX enforcing policy. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In the /root directory, a hidden file. “john-1.9.0.tar.gz” is a password cracker file which contains a number of hashed passwords and can crack them. | For marker’s use only |
| How would you correct the problem? | Use salt with passwords since the salt values would need to be precomputed with hashed passwords which makes password cracking difficult. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In the /etc/sudoers.d, sudoers.d is world writable. In sudoers.d file, users can be granted elevated privileges, so write permission must be restricted to root. | For marker’s use only |
| How would you correct the problem? | Modify permissions using chmod command so that only root is able to write. (chmod 755). |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In /etc folder, when ls -Z passwd command is used to view the security label, + appears. This means that /etc/passwd file has an extended Access Control list.  -rw-rw-r--+ root root system\_u:object\_r:passwd\_file\_t:s0 passwd | For marker’s use only |
| How would you correct the problem? | Remove the extended ACL from the file.  setfacl - -remove-all /etc /passwd |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In the /home/reg mbox file, username and password are mentioned. Password is not encrypted and is in plain text. Unauthorized users could login and modify account details of the original user, and original user will not be allowed access to his/her own account. Mail should be encrypted and should not be read by group or others. | For marker’s use only |
| How would you correct the problem? | Asymmetric encryption could be used, so that sender can use recipient’s public key and recipient of the mail can use their own private key. Change file permissions to -rw------ of mbox. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In the /etc/init.d functions file umask (user file-creation mode mask) is set to 022, which means the root can read, write and execute and others and group can read and execute newly created files. There may be files that the only root should have private access to. If others and group are able to execute and read private files, security is breached. | For marker’s use only |
| How would you correct the problem? | Change umask value to 077, which means read, write and execute permissions for owner and no permissions for group or others. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | The change root command, chroot, changes the root directory and restricts the part of the filesystem available to users. Only files below the new root file will be accessible. This is a defect because all the programs or system files that users might require will now not be accessible if the root directory is changed. If an attacker is able to use chroot command, it will restrict file access. | For marker’s use only |
| How would you correct the problem? | Change permissions so that chroot command can only be executed by the root/superuser. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | Typing the mount command produces a result in which “system-1 on /proc/sys/fs/binfmt\_misc type autofs (rw, relatime,fd=30, pgrp=1, timeout=0, minproto=5, maxproto=5,direct,pipe\_ino=10116). It doesn’t contain the label nosuid, which can be a security defect. This is because the mounted filesystems could contain a file with SUID to root programs. Once the filesystem has been mounted, the attacker could obtain root status running such program. | For marker’s use only |
| How would you correct the problem? | By adding nosuid and noexec options. These commands turn off the SUID and SGID bits on the mounted filesystem and ensure no binaries can be executed from the mounted filesystem. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In the /home/dirk folder, update file is permitted to be read by user, group and others and written by user and group. However, since the file contains private information, it should only be allowed to be read and written by user only. | For marker’s use only |
| How would you correct the problem? | Change file permissions to -rw--------. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In the /root folder’s .bash\_history file, ‘setfacl -m u:reg:rw /etc/passwd’ command has been run. This is to give reg(a user) permission to read and write the passwd file, however passwd file should have read write permission for root, and readable by group and others. Since reg is not the root, reg must not be given access. | For marker’s use only |
| How would you correct the problem? | Change file permissions to -rw-r - - r- -. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In the /var/log folder, messages file has read and write permissions for user and group and read permissions for others. However, messages should have read and write permissions for root user only, since it contains sensitive system information which can be misused. | For marker’s use only |
| How would you correct the problem? | Change file permissions to rw--------. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In the /home/gordon folder, .bash\_history file shows that Gordon (user) su to reg and root. Su reg command does not require password, therefore Gordon was successfully able to get into reg’s account and access reg’s files. Users must not be able to access other users private account and access their files. | For marker’s use only |
| How would you correct the problem? | Add password authentication for reg. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In the /home/reg folder .bash\_history file, sudo rm -rf tmp2/ command has been executed. This command allows removing root owned files/directories recursively and without confirmation or displaying a warning. This command is unsecure since it could delete critical data. | For marker’s use only |
| How would you correct the problem? | Use protect command. When deleting protected files (for example system files), question will be asked. If the answer provided is incorrect, file would not be deleted. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In the /etc directory, ssh folder should have drwx (read, write and execute) for the user, and must not be accessible by group or others. However, in cwk2a image directory permissions are drwxr-xr-x, which means it is readable and executable for group and others. | For marker’s use only |
| How would you correct the problem? | Change file permissions to drwx------. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In the /etc/ssh folder all private ssh\_keys (ssh\_host\_ecdsa\_key, ssh\_host\_ed25519\_key, ssh\_host\_rsa\_key) have file permissions of read and write by user and read by group, whereas only user should have read and write permission, since it contains private key. Private key can be used to decrypt messages that were encrypted with the user’s public key, hence unintended users will be able to view information/data. | For marker’s use only |
| How would you correct the problem? | Change file permissions to -rw-------. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In the /mnt/disk/regstorage/tmp1/etc shadow file has permissions world readable and writeable by user. However, since the shadow file contains hashes, it should only be read and write for user and no permissions for group and others. Also, in the file (for root and tc) MD5 is used, which is insecure. (Note- I was only able to get to regstorage folder after running ./extract.sh in /home/reg/projecttc, and then cd to /mnt/disk/regstorage from there). | For marker’s use only |
| How would you correct the problem? | Change file permissions to rw- - - - - - and use SHA 256 or 512 for encryption. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In //mnt/disk/regstorage/tmp1/home/tc .ashrc file a secret Linux installation is found and bootCentos.sh can be run to reboot. It finds processor ids and kills processes based on the parameters. Home/reg/projecttc runnotc.sh file is copying the OS boot files (notcboot) to /boot/grub2 folder. In home/reg/projecttc, run.sh reboots the system, so that secret installation can come into effect. /mnt/disk/notcboot grub.cfg file shows that an attacker has intruded the system. E.g. videos are loaded and custom menu entries are added. | For marker’s use only |
| How would you correct the problem? | Prevent files being executed by shell, so that unsafe installations do not occur. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In /home/reg/projecttc when extract.sh file is run (using ./extract.sh) opt command, which installs application software packages, is run with tcemirror, bootsync.sh, shutdown.sh, bootlocal.sh. Bootsync.sh, bootlocal.sh and shutdown.sh modify the behaviour from the base system. They block the boot process until they are finished. Hence are a defect in Linux. | For marker’s use only |
| How would you correct the problem? | Restricting users to run opt with specific arguments such as bootlocal.sh, shutdown.sh and bootsync.sh. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In /home/reg folder’s myscriptlauncher.c file has the command setuid(getuid()). It is also executing myscript.sh program. myscriptlauncher.c can set the environment variables for myscript.sh to arbitrary values. This is a weakness since the invoker of the myscriptlauncher.c is in control of the environment variables that are given. An attacker can try to take control of execution by setting the environment variables to dangerous values. For example, an attacker may set IFS (an environment variable) to unusual values to prevent protection mechanisms that filter out dangerous inputs. | For marker’s use only |
| How would you correct the problem? | As a countermeasure, myscriptlauncher.c could erase the entire environment and reset necessary environment variables to safe values. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | Searchpath for adam, svlad, dirk, Gordon, noel, reg and susan were incorrect. Search path was found from path variable in each user’s .bash\_profile file in /home/*user (*Replace user with adam, svlad, etc.). This means execution of programs is taken from $PATH:$HOME/.local/bin:$HOME/bin rather than the user’s home directory. A weakness could be that it is possible to insert a trojan by giving it the same name as an existing program and putting it in a directory which is searched earlier than the directory containing original program. | For marker’s use only |
| How would you correct the problem? | Call programs by giving their full path name, e.g. /bin/su instead of su and correct the searchpath to user’s home directory. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In /home directory named Gordon and noel have permissions drwxrwxrw- which means group can read, write and view and others can read and write the files in the Gordon and Noel home directory. The data inside the folders might be sensitive. Therefore, the user’s home directory should have permissions so that only user is able to view their own files. | For marker’s use only |
| How would you correct the problem? | Change folder permissions to drwx---------. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In /etc passwd file has -rw-rw-r--+ permissions, which means it is writable by group. An attacker is able to edit passwd can become superuser by changing its UID to 0. | For marker’s use only |
| How would you correct the problem? | Passwd file has to be write protected. Change file permissions to -rw-r - - r- -. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | In /home/svlad .bash\_history contains svlad’s password. Attackers can intercept the file, and login as svlad, damaging svlad’s data. | For marker’s use only |
| How would you correct the problem? | Change svlad’s password to a stronger password. Remove it from .bash\_history file. |  |

|  |  |  |
| --- | --- | --- |
| Describe the defect | Su svlad doesn’t require a password and changes the user to svlad. This is a defect because all users of the system (and attackers) can impersonate svlad and inject malicious files and damage the system. Su adams, su Gordon, etc. all require a password. | For marker’s use only |
| How would you correct the problem? | Change so that su svlad requires a password to change user to svlad. |  |

**Please write the total number of security defects you found in the following box:**

**28**

12