Part 1 Template – Ethan Roepke

(Note: the points for each row are given on an all-or-nothing basis. You must properly fill out each column in the row to receive all of the points. Properly filling out ⅔ of the columns earns zero points. All of the fields in the row must be adequate to receive full credit for the given row.)

Host discovery (2 points per row; **10 total points**)

Host Machines	Open Ports/Services	Operating System
Ambulance Laptop (135.75.54.158)	Port 135(TCP) - msrpc - Windows RPC Port 139(TCP) - Netbios-ssn - Windows netbios-ssn Port 445(TCP) - Microsoft-ds - Windows 7-10 microsoft-ds Port 3389(TCP) - - Ms-wbt-server - Microsoft Terminal Servies	Windows 10
Database (135.75.54.156)	All scanned ports are in ignored state	Ubuntu linux 20
Web Server (135.75.54.150)	Port 22(TCP) - SSH - OpenSSH 7.6p1 ubuntu Port 8000(TCP) - http - nginx 1.14.0	Ubuntu linux 18
Clinician Desktop (135.75.54.154)	Port 135(TCP) - msrpc - Windows RPC Port 139(TCP) - Netbios-ssn - Windows netbios-ssn Port 445(TCP) - Microsoft-ds - Windows XP microsoft-ds	Windows XP
Reception Desktop (135.75.54.152)	Port 135(TCP) - msrpc - Windows RPC Port 139(TCP) - Netbios-ssn	Windows 10

	- Windows netbios-ssn Port 445(TCP) - Microsoft-ds Port 3389(TCP) - Ms-wbt-server - Microsoft Terminal Servies	

Exploiting three machines (10 points per row; **30 total points**)

Host Machines	**How did you gain access?**	**What specific harm could be done?**	**How can you remediate it?**
Ambulance Laptop	I first ran a Nessus scan on 135.75.54.150-160. I was mainly looking at the scans for machines .150, .156, and .158 since we are allowed to exploit these machines. Once the scanned finished, 135.75.54.158 came back with a few vulnerabilities. One that stood out was a Microsoft Windows SMB server for eternalblue. I started a metsaploit session and searched through smb ms17_010. I used exploit/windows/smb/ms17_010_eternalblue at first but realized I did not get the right hashes so then I tried exploit/windows/smb/ms17_010_psexec next. I went through the Metasploit steps by setting RHOST to 135.75.54.158 and then ran "exploit". I then ran "post/windows/gather/hashdump" and saved the hashes into my own file. I tried cracking the hashes with John and hashcat but I couldn't crack them. I used Pass The Hash with a reverse shell on Tom who is the head security and my netcat listner was connected. I ran whoami to verify I was on Tom's account and returned with "desktop-tnl6g94/tom"	Since we were able to exploit and gain access on the meterpreter shell. We would be able to steal files and personal information. We were able to gather the hashes through a hashdump command. Gathering the hashes we could try to crack the passwords or get a reverse shell on one of the users. We could obtain a PowerShell on the windows machine and have full access to everything.	First we found a vulnerability for eternalblue when ran a Nessus scan so we should make the proper OS updates so the version is updated to be protected from vulnerabilities. The next thing we got were the hashes and was able to use Pass The Hash on them. We would need to make the hashes salted so we would have random characters added to the hash. To remediate from Pass The Hash, port 445 is a SMB we should add firewall rules to restrict access to port 445.

Database	I modified the settings for the database using “kali-linux-2022.4-live-amd64.iso" and changed the delay to 5000ms to allow me to get on the setup page for the live kali image at restart. After successfully loading kali, I opened a terminal and “mount /dev/sda2 /mnt”. I opened up /etc/shadow and saw root as well as other employers on their. With this I went ahead a deleted the password hash for root. I “umount /mnt” at the end and restarted the machine and let the linux server open to the login page. I was able to login as root with password required. Separately for a test I deleted hashes from employers in /etc/shadow and was able to login with no password required as well	The database holds all employer information, company information, and patient information on their. I was able to gain root access and find patients SSN and credit card info running a mySQL. None of this information was encrypted as companies should hide and protect customer information at all costs. An attacker can steal the credit card number and SSN for identity theft.	First we were able to modify the database settings for me to open a kali on restart. To remediate this, we can add a password to this so the attacker cannot modify the settings for the database and as well can block unsigned bootloaders. Next were able to gather patients personal data. So we should encrypt all of patients data as well as any other personal information stored on the database.
Web Server	Machine 135.75.54.150 had port 22 which is SSH. We should be able to login using SSH with one of the accounts. I tried with multiple accounts but all had passwords and was not able to crack any passwords with John or Hashcat. So then I tried to SSH with this command “ssh 135.75.54.150” and gave me root access immediately with no password required.	Any web server will have a lot of customer data, which can include username/passwords, credit card information, SSN and much more. An attacker can steal customers identity and credit card numbers. I was able to read/modify flaskApp.py file so an attacker can modify the website and find other files in the code. The attacker can hold the website as ransomware.	We were able to SSH without no password required straight into root access. First to remediate is to not allow empty passwords for any user so no one can login with no password. Next we were given root access right away. We should get authentication for SSH when you login anytime.

Sensitive information (10 points per row; **30 total points**)

Host Machines	**What information I found, and why it's bad that I can see it.**
Ambulance Laptop	Being able to access hashes is not terrible if they are salted because the attacker would need to know the hash and salt value. However if they are unsalted and an attacker is knowledgeable in cracking hashes, therefore this is really bad information for attackers to be able to access. If the attacker is able to crack the password and have your username, they will be able to login to your account. To take it to another level, most people reuse the same password for other accounts or very similar passwords and can crack similar passwords to gain access to other accounts as well.
Database	After gaining access to the database, they had a file I found named "Importantinfo.txt" which reminded them to encrypt the user data. Its not bad to see a text file in general but if you have a text file with import information, this is bad that I can see this. With the user data not being encrypted, an attacker would try to find the user data as quick as possible. I was able to find the user information by running mySQL and searched tables for user data and then searched for the customer data. After seeing the table for customer data, I was able to see customers names, as well their SSN and credit card numbers. An attacker can use identity theft/fraud to open bank account or many things under their name. The attacker as well can make purchases with the credit card number.
Web Server	Being able to see all the employeeMemo information is not technically bad, however in this case it is bad because a lot of the employees stored important information inside their to give attackers leverage in the attack. An example is Bob who is told to stop keeping his login information under his computer. An attacker can plan a visit to the office and blend in as an employer and easily steal that file and have all of Bobs login information. I found a python file that is readable named "secureCrypt.py" that has their encryption algorithm inside to change plaintext into ciphertext. This is bad for an attacker to see because if they crack the encryption algorithm, all that plaintext information that's encrypted is no longer secure.

Remediation (10 points per row; **30 total points**)

Host Machines	**Vulnerabilities, misconfigurations, sensitive information disclosures, malpractices**	**Does the issue need to be fixed? Why, or why not?**	**If actions were taken, how did you remediate it**
Ambulance Laptop	I checked which admins were local by running "get-LocalGroupMember-Group Administrators". After running that Patricia's information was local and as well has patient care reports that I am able to access since privileges are misconfigured to allow everyone to view.	Keeping information needs to be fixed to only allow that information to be locally on machines. An attacker can access the data if it is only on the local machine. As well if the data is erased or server goes down, you would not be able to access that information again. The patient care report needs to change the privileges because everyone should not be allowed to access this information.	To properly remediate these flaws, they should implement a cloud to back up all their information which will be secure and private. To remediate the patient care reports, we would need to change the privileges so Patricia(lead paramedic) can modify and view the report as well as others who should be allowed to access the report.
Database	When I gained access to root on database, I was able to access mySQL. When looking through mySQL, I found patient data inside, such as their credit card information and their SSN. As well when I logged on, I did a search "ls" to view what directories or files was on their. I was able to see a Importantinfo.txt on home directory that I was successful to open. In the file, it reminded them to encrypt the user data.	To be able to access the mySQL with no security needs to be fixed to prevent an attacker from being granted access right away. To fix this, we need to add a level of security or password to gain access to the mySQL. The text file found on home directory should not be allowed to be readable by everyone since it has private information to the company. If it didn't store private information then it may be fine. To fix this, we should modify the privileges to allow root and whoever else to be able to read.	To remediate the mySQL issue, thye should add a password to be able to access the database. An additional level of security we can have someone check logs on mySQL to prevent unwanted users on. Since importinfo.txt has sensitive data that can harm them, they should change privileges to allow only root and other users that are allowed to view the file to be able to access it.

Web Server	I was able to login into the web server with “ssh 135.75.54.150” and required no password. This tells me no password is required for users. You can verify this by checking shadow and see if they have hashes for the password or not. I was automatically put into root as well which is not good. Bob had a folder in their called employeeMemo and I was able to read this. It held important information in it that should not be readable to everyone.	Being able to login into the server, as well as in root with no password at all is a big issue that can give attackers leverage to do damage that should be fixed. Looking through the sshd config file, I noticed PermitRootLogin was set to yes which allows easy access to root. It is a good practice to set to no so it has some level of security to be authenticated.	To remediate the issues with no password on user accounts, i would configure the sshd config to require passwords on login for root and just trying to ssh into the server. For gaining root with no authentication needs to be changed so it requires authentication to gain access. PermitRootLogin should be set to no so anyone that’s trying to get as root needs some form of authentication to access.