CHAP Authentication Variation 2

Ronique Young

2/22/2025

Alright, so I had these tasks to get `R1` and `R2` connected over a serial link, and I’ll Walk you through how I handled them, like I’m figuring it out as I go.

A diagram of a router

AI-generated content may be incorrect.

Preliminary Stages:

Task 1: I started by setting the hostnames on the routers to match the topology. On `R1`, I jumped in with `R1>enable` to hit privileged mode, then `R1#config t` to get into global config mode. I typed `R1(config)#hostname R1` to name it `R1`—keeps it clean and simple. Then I swung over to `R2`, entered `R2#config t`, and set `R2(config)#hostname R2`. I just wanted them labeled right so I wouldn’t get confused later. Easy first step!

Task 2: was about enabling the `Serial0` interfaces and sorting out the DCE side of things. The topology marked `R2`’s `Serial0` as the DCE, meaning it’s the one driving the clock. On `R1`, I went `R1(config)#interface Serial0` and fired it up with `R1(config-if)#no shutdown`. Then on `R2`, I did `R2(config)#interface Serial0` and `R2(config-if)#no shutdown` to bring it online too. Since `R2` is the DCE, I double-checked with `R2#show controllers Serial0`, and it confirmed “DCE” in the output—perfect, no surprises there. As the DCE, `R2` needed to set the clock speed, so I went back with `R2(config)#interface Serial0` and typed `R2(config-if)#clock rate 800000` for 768 Kbps, just as the task wanted. To make sure `R1` was getting it, I ran `R1#show controllers Serial0` on `R1`, and it showed the clock signal coming through on the DTE side—`R2` was definitely doing its job.

Task 3: I switched to PPP encapsulation and plugged in those IP addresses you gave me. On `R1`, I went `R1(config)#interface Serial0` and set `R1(config-if)#encapsulation ppp` to get PPP going. Then I added the IP with `R1(config-if)#ip address 192.168.50.33 255.255.255.224`—that `/27` translates to a `255.255.255.224` mask, so I rolled with it. Over on `R2`, I did `R2(config)#interface Serial0`, typed `R2(config-if)#encapsulation ppp`, and set `R2(config-if)#ip address 192.168.50.34 255.255.255.224` to match. PPP felt like the right call for this link, and with `192.168.50.33` and `.34` in the same `/27` subnet—`192.168.50.32` to `.63`—they should have no trouble chatting.

Task 4 was my chance to check my work and test the connection. To verify encapsulation, I ran `R1#show interface Serial0` on `R1`, and it popped up “Encapsulation PPP,” exactly what I was after. Same thing on `R2` with `R2#show interface Serial0`—PPP was there too, so I knew I’d set it right. Now for the real test: pinging. From `R1`, I typed `R1#ping 192.168.50.34`, and I got those happy reply dots back—`R2` was responding! I flipped to `R2` and ran `R2#ping 192.168.50.33`, and it worked too. Those pings told me the interfaces were up, PPP was solid, and the IPs were spot on.

A screenshot of a computer

AI-generated content may be incorrect.For Task 5 of this project, I configured PPP CHAP authentication on Router1 and Router2 to secure their serial connection, using the specified credentials: Router1 with the CHAP username "Router1" and password "MyPass," and Router2 with the username "Router2" and password "MyPass." Starting with Router1, I entered global configuration mode by typing "configure terminal." In this mode, I set up the expected credentials for Router2 using the command "username Router2 password MyPass," which allows Router1 to verify Router2’s identity during authentication. Then, I accessed the serial interface with "interface Serial0/0" and enabled PPP encapsulation using "encapsulation ppp," a prerequisite for CHAP. Next, I activated CHAP authentication with "ppp authentication chap" and defined Router1’s CHAP identity as "Router1" using "ppp chap hostname Router1." This completed Router1’s configuration for Task 5.

A screenshot of a computer

AI-generated content may be incorrect.

For Task 5 of this project, I configured Router2 to establish PPP CHAP authentication with Router1, adhering to the specified credentials: Router2 using the CHAP username "Router2" and password "MyPass," and recognizing Router1 with the username "Router1" and the same password. I initiated the configuration by entering global configuration mode on Router2 with "Router2#config t," allowing me to input commands one per line, to be terminated with CTRL/Z. My first step was to configure the serial interface by typing "int se0," shorthand for "interface Serial0," though ideally, I’d use the full notation like "Serial0/0" for clarity. Upon entering this mode, I observed a system message: "%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state to down," indicating the interface protocol went down, likely due to a mismatch in configuration or encapsulation with Router1 at that point.

Next, I enabled CHAP authentication with "ppp authentication chap," setting Router2 to require CHAP for the link. I then specified Router2’s CHAP identity with "ppp chap hostname Router2," ensuring it identifies itself correctly to Router1. Attempting to set the password directly on the interface, I entered "ppp chap password MyPass," which is a valid command option, though not the approach specified in Task 5’s global username requirement. Exiting the interface mode, I mistyped "exir," resulting in an error message: "% Invalid input detected at '^' marker," highlighting the typo. I corrected this with "exit" to return to global configuration mode. In global config, I entered "username Router1 password MyPass" to define Router1’s expected credentials, allowing Router2 to authenticate Router1. After completing the configuration with "end," the system logged "%SYS-5-CONFIG\_I: Configured from console by console," confirming the changes.

Post-configuration, I observed mixed interface status updates: "%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state to up" followed immediately by "%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial0, changed state to down." This suggests the PPP link briefly established but then failed, possibly due to encapsulation not being explicitly set to PPP on Router2 (requiring "encapsulation ppp" on the interface) or a mismatch with Router1’s configuration. For Task 5’s requirements, Router2’s setup partially aligns by using CHAP with the correct hostname and recognizing Router1’s credentials globally, though adding "encapsulation ppp" and ensuring Router1’s matching configuration would stabilize the link.

A screenshot of a computer

AI-generated content may be incorrect.

For Task 6 of this project, I enabled PPP authentication debugging on Router1 to monitor the CHAP authentication process between Router1 and Router2, then performed a shutdown and no shutdown sequence on Router2’s Serial0 interface to trigger authentication, verified the debugging output, and subsequently disabled debugging. Starting with Router1, I entered privileged EXEC mode and activated PPP debugging with the command "debug ppp authentication." This command enables detailed logging of PPP authentication events, specifically the CHAP handshake, which Task 5 configured with Router1 using username "Router1" and password "MyPass," and Router2 using username "Router2" with the same password. With debugging enabled on Router1, I expected to see messages confirming mutual authentication once the interface was cycled.

Next, I moved to Router2 to perform the interface cycle as specified. Entering global configuration mode with "Router2#config t," I accessed the serial interface using "int se0," shorthand for "interface Serial0" (though "Serial0/0" would be more precise). I issued the "shut" command to administratively disable the interface, and the system responded with "%LINK-5-CHANGED: Interface Serial0, changed state to administratively down," confirming the shutdown. Immediately after, I entered "no shut" to reactivate the interface, receiving "%LINK-5-CHANGED: Interface Serial0, changed state to up," indicating the physical link was restored. This cycle was intended to reset the PPP session and trigger CHAP authentication, observable via Router1’s debug output.

Back on Router1, with "debug ppp authentication" active, I anticipated seeing log messages such as "CHAP: O CHALLENGE" (Router1 sending a challenge), "CHAP: I CHALLENGE" (Router2’s response), followed by "CHAP: O RESPONSE" and "CHAP: I RESPONSE" as each router authenticates the other, culminating in "CHAP: O SUCCESS" and "CHAP: I SUCCESS" to confirm successful mutual authentication. These messages would verify that Router1 (hostname "Router1") and Router2 (hostname "Router2") correctly exchanged and validated credentials using the shared password "MyPass," as configured in Task 5. The absence of "FAILURE" messages would further indicate a successful CHAP process. Once verification was complete, I disabled debugging on Router1 with "undebug all," receiving the confirmation "All possible debugging has been turned off," ensuring no further debug overhead.

** Click on the Packet Tracer PKT File To See the Full Configuration**