

Cómo configurar el soporte para TACACS+ en el motor de la memoria caché

Contenido

[Introducción](#)

[prerrequisitos](#)

[Requisitos](#)

[Componentes Utilizados](#)

[Convenciones](#)

[Configurar](#)

[Diagrama de la red](#)

[Configure el motor del caché para el soporte de TACACS+](#)

[Verificación](#)

[Comandos para resolución de problemas](#)

[Información Relacionada](#)

[Introducción](#)

Este documento describe cómo configurar el Terminal Access Controller Access Control System más el soporte (TACACS+) para acceder el Cisco Cache Engine. Las instrucciones en este documento permiten que usted valide contra un servidor/una base de datos del telecontrol TACACS+ cuando usted telnet al motor del caché. Si el servidor no incluye una entrada para su identificación del usuario, marca localmente para saber si hay la información de acceso válida.

[prerrequisitos](#)

[Requisitos](#)

No hay requisitos específicos para este documento.

[Componentes Utilizados](#)

La información que contiene este documento se basa en las siguientes versiones de software y hardware.

- Cisco Cache Engine 505 en un ambiente de laboratorio con las configuraciones despejadas
- Versión 2.3.1 del software Cisco Cache Engine.
- CiscoSecure para UNIX

La información que contiene este documento se creó a partir de los dispositivos en un ambiente de laboratorio específico. Todos los dispositivos que se utilizan en este documento se pusieron en funcionamiento con una configuración verificada (predeterminada). Si la red está funcionando,

asegúrese de haber comprendido el impacto que puede tener cualquier comando.

[Convenciones](#)

Refiera a los [convenios de los consejos técnicos de Cisco](#) para la información sobre las convenciones sobre documentos.

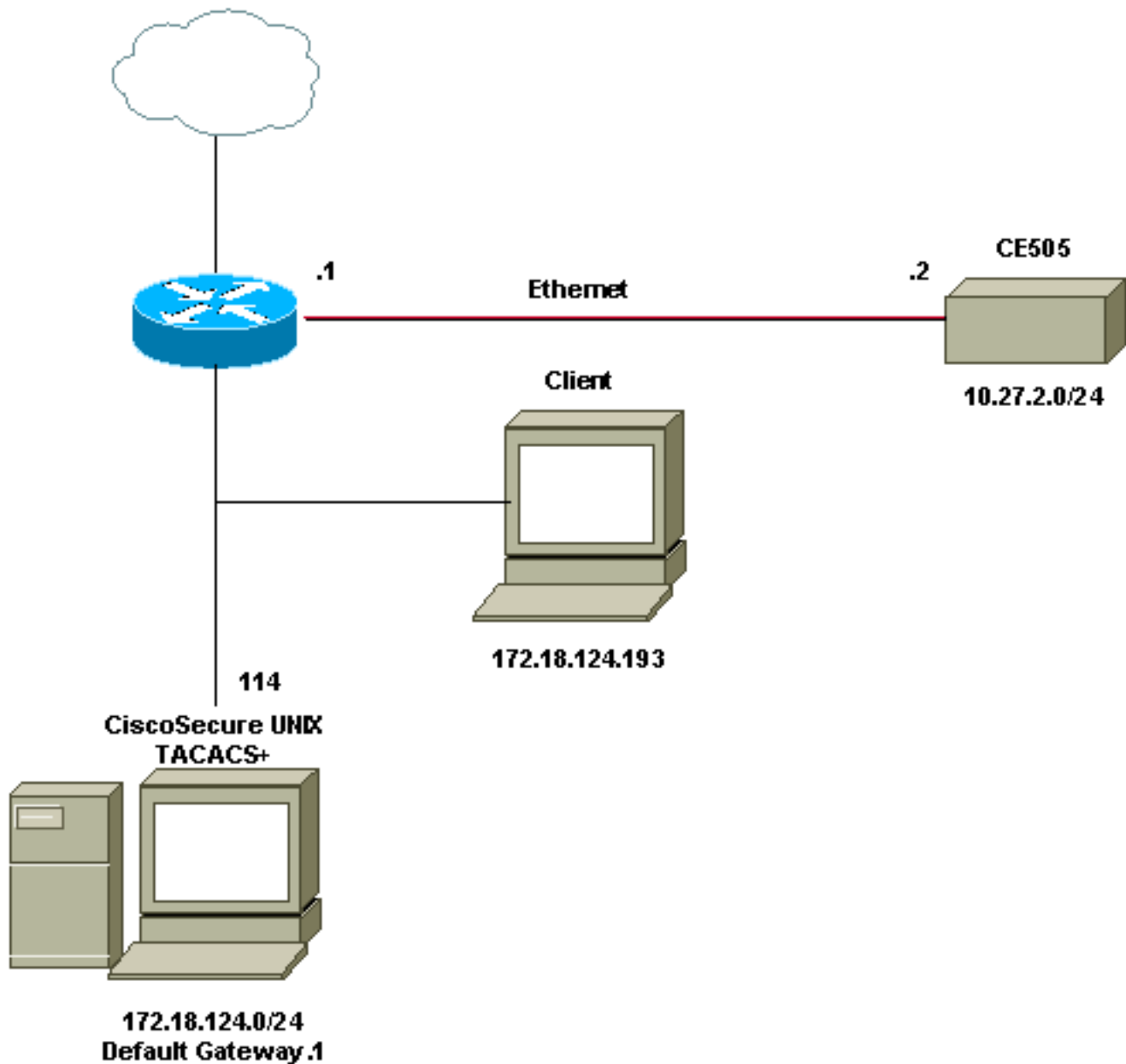
[Configurar](#)

En esta sección encontrará la información para configurar las funciones descritas en este documento.

Nota: Utilice la herramienta [Command Lookup Tool](#) ([clientes registrados solamente](#)) para obtener más información sobre los comandos utilizados en esta sección.

[Diagrama de la red](#)

En este documento, se utiliza esta configuración de red:



[Configure el motor del caché para el soporte de TACACS+](#)

Complete estos pasos para configurar el motor del caché para el soporte de TACACS+:

1. Configuración del Cache Engine para la versión correspondiente del Protocolo de comunicación de la memoria caché de la Web (WCCP).
2. Utilice estos comandos para la configuración predeterminada:

```
authentication login local enable authentication configuration local enable
```
3. Configure el dirección IP del servidor TACACS+. Si los servidores múltiples especifican cuál dirección es primaria, entonces los servidores secundarios quedan como opciones en blanco.
4. Configure la autenticación al servidor TACACS+ como primaria. Si el servidor no está disponible, el valor predeterminado será la autenticación especificada localmente.
5. Autenticación de la configuración a la información fundamental TACACS+ en caso necesario.

Nota: Usted debe habilitar el TACACS+ en el Cisco Cache Engine porque los motores del Cisco Cache utilizan el PPP para autenticar con el servidor TACACS, a diferencia del Routers que no requiere el PPP. Para habilitar el TACACS+ en los motores del Cisco Cache, el Cisco Secure

ACS abierto 2.6, hacer clic la lengüeta de la **configuración de grupo**, y marcar la casilla de verificación **IP PPP** situada en el área de las configuraciones TACACS+.

Sus líneas de comando deben aparecer similares a esta salida:

```
cepro(config)#tacacs server 172.18.124.114 cepro(config)#authentication login tacacs ena primary
cepro(config)#authen configuration tacacs enab
```

Verificación

Use esta sección para confirmar que su configuración funciona correctamente.

[La herramienta Output Interpreter Tool \(clientes registrados solamente\)](#) (OIT) soporta ciertos comandos show. Utilice la OIT para ver un análisis del resultado del comando show.

- **versión de la demostración** — Visualiza el software que se ejecuta en el motor del caché, así como algunos otros componentes como el tiempo de actividad del sistema (tal como donde el código fue iniciado previamente y la fecha en que fue compilado).
cepro#show version Cisco Cache Engine Copyright (c) 1986-2001 by Cisco Systems, Inc. Software Release: CE ver 2.31 (Build: FCS 02/16/01) Compiled: 11:20:14 Feb 22 2001 by bbalagot Image text-base 0x108000, data_base 0x437534 System restarted by Reload The system has been up for 20 hours, 42 minutes, 59 seconds. System booted from "flash"
- **hardware de la demostración** — Visualiza la misma información que el comando show version, así como a los componentes de hardware del motor del caché.
cepro#show hardware Cisco Cache Engine Copyright (c) 1986-2001 by Cisco Systems, Inc. Software Release: CE ver 2.31 (Build: FCS 02/16/01) Compiled: 11:20:14 Feb 22 2001 by bbalagot Image text-base 0x108000, data_base 0x437534 System restarted by Reload The system has been up for 21 hours, 15 minutes, 16 seconds. System booted from "flash" Cisco Cache Engine CE505 with CPU AMD-K6 (model 8) (rev. 12) AuthenticAMD 2 Ethernet/IEEE 802.3 interfaces 1 Console interface. 134213632 bytes of Physical Memory 131072 bytes of ROM memory. 8388608 bytes of flash memory. List of disk drives: /c0t0d0 (scsi bus 0, unit 0, lun 0)
- **ejecutar-config de la demostración** — Visualiza la configuración corriente en el motor del caché.
cepro#show running-config Building configuration... Current configuration: !! user add admin uid 0 password 1 "eeSdy9dcy" capability admin-access user add chbanks uid 5001 password 1 "eeSdy9dcy" capability admin-access !! hostname cepro ! interface ethernet 0 ip address 10.27.2.2 255.255.255.0 ip broadcast-address 10.27.2.255 exit !! interface ethernet 1 exit ! ip default-gateway 10.27.2.1 ip route 0.0.0.0 0.0.0.0 10.27.2.1 cron file /local/etc/crontab ! wccp router-list 1 10.27.2.1 wccp web-cache router-list-num 1 ! authentication login tacacs enable primary authentication login local enable !--- on by default ---! authentication configuration tacacs enable authentication configuration local enable !---- on by default ---! tacacs server 172.18.124.114 primary rule no-cache url-regex .*cgi-bin.* rule no-cache url-regex .*aw-cgi.* !! end cepro#
- **tacacs de la demostración** — Visualiza las configuraciones para el servidor TACACS+.
cepro#show tacacs Login Authentication for Console/Telnet Session: enabled (primary) Configuration Authentication for Console/Telnet Session: enabled TACACS Configuration: ----- Key = Timeout = 5 seconds Retransmit = 2 times Server Status ----- 172.18.124.114 primary
- **muestre los tacacs de las estadísticas** — Estadísticas de las visualizaciones TACACS+.
cepro#show statistics tacacs TACACS+ Statistics ----- Number of access requests: 13 Number of access deny responses: 7 Number of access allow responses: 0
- **autenticación de la demostración** — Visualiza la configuración actual actual de la autenticación y autorización TACACS+.
cepro#show authentication Login Authentication: Console/Telnet Session ----- local enabled tacacs enabled (primary) Configuration Authentication: Console/Telnet Session ----- local enabled tacacs enabled cepro#

Comandos para resolución de problemas

En esta sección encontrará información que puede utilizar para solucionar problemas de configuración.

[La herramienta Output Interpreter Tool \(clientes registrados solamente\)](#) (OIT) soporta ciertos comandos show. Utilice la OIT para ver un análisis del resultado del comando show.

Nota: Consulte [Información Importante sobre Comandos de Debug](#) antes de usar un comando debug.

- **debug de la demostración** — Visualiza los comandos debug se habilitan que.
`cepro#show debug Authentication debugging is on Tacacs debugging is on`
- **monitor terminal** — Visualiza las salidas de los debugs a la pantalla. Esta salida visualiza los resultados de los comandos **debug authentication and debug tacacs**.
`cepro#terminal monitor`
`cepro#authenticateUser(): Begin setRemoteIPAddress(): pRemoteAddress 172.18.124.193`
`bAuthentication(): Begin bAuthenticationIntersection(): Begin bAuthenticationIntersection(): telnet_access 1 setAuthenticatedService(): nServiceToAuthenticate 6`
`getAuthenticatedService(): Begin getAuthenticatedService(): nServiceToAuthenticate = 6`
`bAuthenticationIntersection() getAuthenticatedService 6 setErrorDisplayed(): Begin bStatus 0`
`getLocalLoginAuthEnable(): Begin getLocalLoginAuthEnable(): uiState = 1`
`getTacacsLoginAuthEnable(): Begin getTacacsLoginAuthEnable(): uiState = 1`
`getTacacsLoginAuthPrimary(): Begin getTacacsLoginAuthPrimary(): uiState = 1`
`IncrementTacacsStatRequest(): Begin tacacs_plus_login() Begin isConsole() Begin`
`getAuthenticatedService(): Begin getAuthenticatedService(): nServiceToAuthenticate = 6`
`isConsole() nReturn 0 telnet tacacs_plus_login() sWhatService() tty = telnet`
`getRemoteIPAddress(): Begin getRemoteIPAddress(): pRemoteAddress = 172.18.124.193`
`tacacs_plus_login() getRemoteIPAddress sHostIp 172.18.124.193 tacacs_malloc() Begin 164`
`tacacs_malloc() PSkmalloc ptr getUserStruct() malloc_named ustr tacacs_plus_login()`
`allocated memory for ustruct aaa_update_user() Begin debug_authen_svc() Begin`
`aaa_update_user(): user='admin' ruser='system' port='telnet' rem_addr='172.18.124.193'`
`authen_type=1 tacacs_plus_login() updated user getNumTacacsLoginAttempts(): Begin`
`getNumTacacsLoginAttempts(): ulRetransmit = 2 ##### tacacs_plus_login() num_tries 1`
`aaa_start_login() Begin debug_start_login() Begin debug_start_login()/AUTHEN/START (0):`
`port='telnet' list=(null) action=LOGIN service=LOGIN aaa_randomize_id() Begin`
`tacacs_plus_start_login() Begin tacacs_parse_server() Begin user_str admin`
`getTacacsDirectRequestEnable(): Begin getTacacsDirectRequestEnable(): cDirectRequestEnable =`
`0 printIpAddr() Begin printIpAddr() 0.0.0.0 tacacs_plus_start_login() server.ip_addr 0.0.0.0`
`server.type 0 server.length 0 choose_version() Begin create_authen_start() Begin`
`create_authen_start() len 45 tacacs_malloc() Begin 45 tacacs_malloc() PSkmalloc ptr`
`create_authen_start() malloc_named tac_pak fill_tacacs_plus_hdr() Begin encrypt 1`
`fill_tacacs_plus_hdr() len 33, tac_pak->length 33 ##### fill_tacacs_plus_hdr() tac_pak-`
`>encrypted 1 ##### fill_tacacs_plus_hdr() TEST nTestLen 33 create_authen_start() len 33,`
`tac_pak->length 33 create_authen_start() u->priv_lvl 15 start->priv_lvl 15`
`create_authen_start() start->action 1 create_authen_start() start->authen_type 1`
`create_authen_start() start->service 1 create_authen_start() user_len 5`
`create_authen_start() port_len 6 create_authen_start() addr_len 14 create_authen_start()`
`out_len 33 tacacs_plus_start_login() TACACS+: send AUTHEN/START packet ver=192 id=1541646967`
`tacacs_plus_start_login() login to TACACS+ server: printIpAddr() Begin printIpAddr() 0.0.0.0`
`tacacs_plus_get_conn() Begin server(0) printIpAddr() Begin printIpAddr() 0.0.0.0`
`tacacs_plus_get_conn() **pSocketHandleIndex 89434348 tacacs_plus_get_conn() Look at server`
`in the TACACS+ server list tacacs_plus_get_conn() TACACS+: This is a loop through server`
`list tacacs_plus_openconn() Begin printIpAddr() Begin printIpAddr() 172.18.124.114`
`open_handle() Begin tacacs_plus_socket() Begin tacacs_plus_socket Socket: return nSocket 784`
`nSockFdTbl[28] = 784 printIpAddr() Begin printIpAddr() 172.18.124.114 open_handle() TACACS+:`
`Opening TCP/IP connection to 172.18.124.114 open_handle() nSockFdTbl[28]= 784`
`setCurrentServer() Begin SaveCurrentServer->ip_addr 172.18.124.114`
`IncrementTacacsStatPerServerRequest(): Begin ##### IncrementTacacsStatPerServerRequest`
`Server->ip_addr 1920733868 tacacs_root.ulTacacsServerAddr open_handle() socket(28) 784`

```

tacacs_plus_connect() Begin tacacs_plus_connect() socket(28) 784 tacacs_plus_connect() End
open_handle() is connected open_handle() *connection_handle 28 open_handle()
**pSocketHandleIndex 28 tacacs_plus_openconn() **pSocketHandleIndex 28 get_server() Begin
tacacs_plus_openconn() server->opens++ tacacs_plus_get_conn() **pSocketHandleIndex 28
tacacs_plus_get_conn() oldServerCount: 0, count:0 tacacs_plus_start_login() **pHandleIndex
28 tacacs_plus_send_receive() Begin tacacs_plus_proc_send_receive() Begin
tacacs_plus_proc_send_receive() length 33 copy_tac_plus_packet() Begin tacacs_malloc() Begin
45 tacacs_malloc() PSkmalloc ptr copy_tac_plus_packet() malloc_named copy
tacacs_plus_encrypt() Begin getTacacsKey(): Begin getTacacsKey(): sKey =
tacacs_plus_encrypt() key tacacs_plus_encrypt() sizeof(tacacs_plus_pkt_hdr) 12
tacacs_plus_encrypt() sizeof(uchar) 1 tacacs_plus_encrypt() tac_pak->encrypted 1
tacacs_plus_encrypt() tac_pak->encrypted = TAC_PLUS_CLEAR && key is empty
tacacs_plus_proc_send_receive() out_pak->encrypted 1 tacacs_plus_proc_send_receive()
out_pak->encrypted 1 tacacs_plus_proc_send_receive() PSkfree dump_pak
tacacs_plus_proc_send_receive() ntohl(out_pak->length) 33 dump_start_session() Begin
ntohl(out_pak->length) 33 getTacacsKey(): Begin getTacacsKey(): sKey = 0xc0 0x1 0x1 0x1 0x77
0xaa 0xe3 0x5b 0x0 0x0 0x0 0x21 0x1 0xf 0x1 0x1 0x5 0x6 0xe 0x0 0x61 0x64 0x6d
encrypt_md5_xor() Begin encrypt_md5_xor() no key dump_summarise_incoming_packet_type() Begin
Read AUTHEN/START size=45 dump_nas_pak() Begin dump_header() Begin PACKET: key= version 192
(0xc0), type 1, seq no 1, encrypted 1 session_id 2007688027 (0x77aae35b), Data length 33
(0x21) End header type=AUTHEN/START, priv_lvl = 15action=login authen_type=ascii
service=login user_len=5 port_len=6 (0x6), rem_addr_len=14 (0xe) data_len=0 User: port:
rem_addr: data: End packet dump_start_session() PSkfree test getTacacsTimeout(): Begin
getTacacsTimeout(): ulTimeout = 5 tacacs_plus_sockwrite() Begin
tacacs_plus_proc_send_receive() PSkfree out_pak getTacacsTimeout(): Begin
getTacacsTimeout(): ulTimeout = 5 sockread() Begin tacacs_plus_proc_send_receive() read
tacacs_malloc() Begin 18 tacacs_malloc() PSkmalloc ptr tacacs_plus_proc_send_receive()
malloc_named *in tacacs_plus_proc_send_receive() allocated memory getTacacsTimeout(): Begin
getTacacsTimeout(): ulTimeout = 5 sockread() Begin tacacs_plus_proc_send_receive() OK
tacacs_plus_decrypt() Begin getTacacsKey(): Begin getTacacsKey(): sKey =
tacacs_plus_decrypt() key tacacs_plus_decrypt() tac_pak->encrypted = TAC_PLUS_CLEAR && key
is empty authen_resp_sanity_check() Begin tacacs_plus_hdr_sanity_check() Begin
authen_debug_response() Begin authen_debug_response() TACACS+: ver=192 id=1541646967
received AUTHEN status = FAIL tacacs_plus_start_login() PSkfree out_tac_pak
unload_authen_resp() Begin tacacs_plus_start_login() PSkfree in_tac_pak
debug_authen_status() Begin TACACS+/AUTHEN (2007688027): status = FAIL tacacs_plus_login()
Authentication failed. tacacs_plus_login() labell aaa_cleanup_login() Begin
aaa_close_connection() Begin tacacs_plus_closeconn() Begin get_server() Begin close_handle()
Begin close_handle() nHandleIndex 28 nSockFdTbl[**handle] 784 aaa_set_password() Begin
aaa_free_user() Begin debug_authen_svc() Begin aaa_close_connection() Begin TACACS+/AUTHEN:
free user admin system telnet 172.18.124.193 authen_type=ASCII service=LOGIN priv_lv
aaa_free_user() PSkfree ustr ##### tacacs_plus_login() num_tries 2 aaa_start_login() Begin
debug_start_login() Begin debug_start_login()/AUTHEN/START (0): port='unknown' list='(null)'
action=LOGIN service=LOGIN TACACS+/AUTHEN/START aaa_start_login() (0): ERROR (no ustruct)
tacacs_plus_login() TACACS+: aaa_start aaa_free_user() Begin tacacs_plus_login()
try_local_login AUTHENTICATION_INTERNAL_ERROR IncrementTacacsStatDenyAccess(): Begin
localAuthentication(): Begin localAuthentication() usrName admin localAuthentication()
passwd system localAuthentication() pUId 89435294 localAuthentication() telnet_access
localAuthentication() rc == TRUE AuthenticationIntersection(): bTacacsLogin 0
IncrementLocalLoginStat(): Begin getLocalConfigAuthEnable(): Begin
getLocalConfigAuthEnable(): uiState = 1 getTacacsConfigAuthEnable(): Begin
getTacacsConfigAuthEnable(): uiState = 1 getTacacsConfigAuthPrimary(): Begin
getTacacsConfigAuthPrimary(): uiState = 0 localAuthentication(): Begin localAuthentication()
usrName admin localAuthentication() passwd system localAuthentication() pUId 89435294
localAuthentication() telnet_access localAuthentication() rc == TRUE
AuthenticationIntersection(): bTacacsConfig 0 AuthenticationIntersection():= Local Database
Authentication == IncrementLocalConfigStat(): Begin AuthenticationIntersection(): user has
been found AuthenticationIntersection(): bTacacsLogin pUId 89435294
AuthenticationIntersection(): GOT ACCESS capab 0 Admin 0 Ftp 0 Http 0 Telnet 0
authenticateUser() AUTHENTICATION IS OK authenticateUser() AUTHENTICATION #2

```

[Información Relacionada](#)

- [Release Note para el Software Cisco Cache, versión 2.3.0 - Memoria caché de FTP](#)
- [Productos y servicios del Cisco 500 Series Cache Engine](#)
- [Soporte Técnico y Documentación - Cisco Systems](#)