SUITE SHADOW

ShadowLogin

Chi entra è chi dice di essere. Password protette come si deve, account ufficiali riconosciuti davvero, e un antibot che si accorge dell'ondata mentre sta arrivando.

VERSIONE	MINECRAFT	SERVER	JAVA
1.0.0	1.26.2	Paper 26.2	25

Manuale d'uso · Funzionalità, comandi, permessi, dipendenze e integrazioni, configurazione, domande frequenti

Indice

SE HAI GIÀ UN ALTRO PLUGIN DI ACCESSO

ShadowLogin **non gestisce l'ingresso** se trova AuthMe, nLogin, JPremium, LoginSecurity, FastLogin o un altro plugin della stessa famiglia. Non è prudenza eccessiva: due plugin che gestiscono l'ingresso non si sommano, si intralciano, e il risultato non è "più sicuro" ma *indeterminato*. Togli l'altro e riavvia. Nel frattempo i comandi di amministrazione restano disponibili, così puoi esportare i tuoi account senza disinstallare niente al buio.

Come leggere questo manuale

Se vuoi solo partire, ti bastano i capitoli **2** e **3**: due pagine e cinque minuti. Il capitolo **5** è quello che vale il prezzo, e va letto per intero se il tuo server accetta anche giocatori senza account originale. Il capitolo **9** serve quando vorrai sapere perché una cosa funziona come funziona.

1. Che cos'è ShadowLogin

È il plugin che decide chi entra sul tuo server. Se accetti anche giocatori senza account originale, è anche l'unica cosa che impedisce a uno sconosciuto di collegarsi con il nome di un altro e prendersi la sua base.

Il problema, in tre righe

Un server in `offline-mode` accetta chiunque dichiari un nome. Non c'è nessuna verifica: il client dice "sono Marco" e il server ci crede. Serve quindi qualcosa che chieda una password — e che la protegga come si protegge una password, non come si protegge un punteggio.

Cosa fa ShadowLogin

Funzione	In una riga
Registrazione e accesso	Password protette con BCrypt, tentativi limitati con attesa che raddoppia
Account ufficiali	Chi ha comprato il gioco entra senza password, perché l'ha già verificato Mojang
Protezione dei nomi	Un nome che appartiene a un account vero non lo può registrare nessun altro
Sessione	Chi rientra dallo stesso indirizzo entro N minuti non riscrive niente
Verifica in due passaggi	Facoltativa, con qualunque app di autenticazione, più codici di riserva
Recupero	Domanda di sicurezza, oppure un codice temporaneo dato dallo staff
Limbo	Prima dell'accesso: niente chat, niente comandi, niente inventario, niente posizione reale
Antibot	Limiti per indirizzo e una modalità stretta che si accende da sola
Esportazione	Tutti gli account in CSV, in qualunque momento, con un comando

Le tre regole con cui è scritto

1. **In caso di dubbio si chiede la password.** Mai il contrario. Database che non risponde, Mojang che non risponde, identificativo che non torna: la risposta è sempre "scrivi la password", che al massimo è una seccatura. La risposta opposta è l'account di qualcun altro.
2. **Niente di lento sul thread di gioco.** Calcolare un hash costa un quarto di secondo: dieci persone che entrano insieme bloccherebbero il server per due secondi e mezzo. Tutto quello che è lento avviene fuori, e il plugin è compatibile con Folia.
3. **Una password non finisce mai in un registro.** Nemmeno troncata, nemmeno in modalità debug. Il capitolo 4 spiega fin dove si può arrivare e dove sta il limite del gioco.

CONTRO COSA SI MISURA

AuthMe è gratuito ed è lo standard, ma è vecchio e la sua configurazione è enorme. JPremium è a pagamento e il suo argomento è proprio il riconoscimento degli account ufficiali. nLogin è a pagamento e molto curato. ShadowLogin punta su tre cose: il riconoscimento fatto *senza indovinare* (capitolo 5), una configurazione che si capisce leggendola, e un antibot che funziona davvero quando serve.

2. Requisiti e installazione

Server	Paper 1.26.2 o successivo (api-version 26.2)
Java	25
Database	Nessuno da installare: H2 locale incluso. MySQL/MariaDB facoltativo
Dipendenze	Nessuna. Né plugin, né librerie sul classpath
Folia	Compatibile

Installazione

1. Copia `ShadowLogin-1.0.0.jar` in `plugins/`.
2. Avvia il server. Il plugin crea da solo cartelle, configurazione, lingue e database locale.
3. Entra in gioco. Ti chiede di registrarti: `/register <password> <password>`.

Cosa trovi nella cartella

```
plugins/ShadowLogin/
  config.yml            impostazioni, con il perché di ogni scelta
  license.yml           la chiave, tenuta fuori dal config apposta
  lang/it_IT.yml        i messaggi in italiano
  lang/en_US.yml        i messaggi in inglese
  menus/login.yml       la schermata a clic: righe, slot, materiali, testi
  report.yml            le segnalazioni per il pannello di ShadowControl
  export/               i CSV di esportazione e importazione
  data/accounts.mv.db   il database locale
```

Se hai una rete di server

Metti le credenziali in `config.yml` sotto `storage.mysql` e riavvia: le tabelle se le crea da solo. Da quel momento una registrazione fatta su un server vale su tutti gli altri, e anche la sessione: chi passa dal lobby al survival non riscrive la password. Serve la versione Premium.

> **SEI DIETRO UN PROXY? LEGGI IL CAPITOLO 9, SEZIONE PROXY**
>
> Dietro un BungeeCord o un Velocity, il server vede l'indirizzo *del proxy*, non quello del giocatore. Senza l'inoltro configurato, tutti i tuoi giocatori risultano collegati dallo stesso indirizzo: i limiti dell'antibot bloccherebbero giocatori veri e le sessioni diventerebbero una sola per tutti. Il plugin se ne accorge da solo e te lo scrive in console — ma la soluzione sta nel proxy, non qui.

Se stai arrivando da un altro plugin

Esporta gli account dal vecchio plugin in un CSV con l'hash BCrypt (quasi tutti lo usano) e importalo con `/shadowlogin import <file>`. Gli account che esistono già non vengono toccati, e le righe con un hash

che non è BCrypt vengono saltate e contate: meglio dieci righe rifiutate e dette che dieci account che non entrano più senza sapere perché.

3. Gratuito e Premium

Un jar solo, con dentro tutte e due le versioni. Senza chiave gira quella gratuita, che è un plugin di accesso completo e sicuro; con una chiave firmata gira la Premium.

NESSUNA FUNZIONE DI SICUREZZA È A PAGAMENTO

BCrypt con lo stesso fattore di costo, i limiti dei tentativi contati per indirizzo e per nome, il limbo, le soglie fisse dell'antibot, la schermata a clic, l'esportazione di tutti gli account, entrambe le lingue: ci sono anche senza licenza. Un plugin che mette la protezione delle password dietro un paywall si guadagna una brutta reputazione la prima volta che qualcuno viene svuotato.

	Gratuita	Premium
Account registrati	100	illimitati
BCrypt, costo aggiornabile, ricalcolo automatico	✓	✓
Tentativi per indirizzo e per nome, attesa crescente	✓	✓
Limbo completo (niente chat, comandi, inventario, posizione)	✓	✓
Antibot a soglie fisse	✓	✓
Schermata a clic e sessioni	✓	✓
Esportazione e importazione	✓	✓
Recupero con domanda e codice dello staff	✓	✓
Riconoscimento degli account ufficiali	—	✓
Verifica in due passaggi	—	✓
MySQL condiviso fra più server	—	✓
Antibot che si stringe da solo sotto attacco	—	✓
Registro degli accessi sul database	ultimi 50 in memoria	completo

Come si comporta il limite

Si applica **in entrata** e non toglie mai niente a nessuno. Il centunesimo account non nasce; i cento che ci sono continuano tutti a funzionare, con le loro password e le loro sessioni. Se un giorno il server torna gratuito — un abbonamento scaduto, una chiave revocata — *nessuno viene chiuso fuori*: semplicemente non si registra più nessuno finché la chiave non torna.

> **IL PLUGIN NON SI SPEGNE MAI PER UN PROBLEMA DI LICENZA**
>
> In un plugin di accesso questo conta doppio: spegnersi vorrebbe dire o chiudere fuori tutti o far entrare tutti, e sono tutte e due peggio di qualunque pirateria. Il caso peggiore possibile è tornare alla versione gratuita, scrivendo in console il motivo. Un abbonamento appena scaduto o un indirizzo che non corrisponde danno **sette giorni di tolleranza**: il rinnovo è quasi sempre già partito, e un indirizzo che non torna è molto più spesso un NAT che un ladro.

Come si attiva una chiave

```
/shadowlogin license claim <codice-acquisto>   il codice ricevuto con l'acquisto
/shadowlogin license activate <chiave>         se hai già la chiave in mano
/shadowlogin license                           stato, limiti, impronta del server
```

La chiave viene scritta in `license.yml`, che sta apposta fuori dal config: una richiesta di assistenza che dice “mandami il config” non deve arrivare con dentro la licenza di qualcuno. Nei messaggi e in console la chiave è **sempre mascherata**, anche per un operatore.

4. Funzionalità

4.1 Le password

Sul disco non c'è nessuna password: c'è un hash **BCrypt**, che contiene già dentro di sé un sale casuale diverso per ogni account e il fattore di costo con cui è stato calcolato. Non si può tornare indietro, e nemmeno chi ha in mano il database può rispondere alla domanda "qual è la password di Marco".

Il fattore di costo (`security.hash.cost`, predefinito **12**) dice quante volte raddoppiare il lavoro: circa un quarto di secondo per calcolo. Chi entra lo paga una volta e non se ne accorge; chi prova un dizionario lo paga per ogni singolo tentativo, moltiplicato per milioni.

> **IL COSTO SI PUÒ ALZARE DOPO, SENZA CHIEDERE NIENTE A NESSUNO**
>
> Il numero giusto oggi sarà basso fra tre anni. Siccome ogni hash porta scritto il proprio costo, basta alzare `security.hash.cost`: al primo accesso di ognuno la password viene ricalcolata col costo nuovo. Nel giro di qualche giorno tutto il database è aggiornato, senza riavvii e senza che nessuno debba fare niente. Al contrario non succede mai: se abbassi il costo, gli hash più forti restano come sono.

Cosa si può usare come password

Poche regole, e sono quelle che contano: lunghezza minima (6), non uguale al proprio nome, non nell'elenco delle cento più usate al mondo. Le regole complicate — una maiuscola, un simbolo — producono password *peggiori* (`Marco1`, `Password!`) e non fermano nessuno: l'opzione c'è ed è spenta di serie.

4.2 I tentativi

Il primo errore costa cinque secondi, il secondo dieci, il terzo venti, e così via fino a un tetto. Dopo dieci errori l'account è fermo per mezz'ora. Chi ha davvero sbagliato a scrivere non se ne accorge: gli serve un tentativo, forse due.

I tentativi si contano **per indirizzo e per nome**, tutti e due, e vince il più severo. Non è ridondanza:

- contare solo il nome lascia passare chi prova `123456` su mille nomi diversi: nessun nome arriva a cinque tentativi e su mille account uno con quella password c'è;
- contare solo l'indirizzo lascia passare una botnet: mille indirizzi che provano cinque password ciascuno sullo stesso account fanno cinquemila tentativi.

4.3 Prima dell'accesso: il limbo

Chi non ha scritto la password non vede niente e non fa niente. Non è un vezzo: la schermata di accesso è l'unico posto dove uno sconosciuto può stare senza essere nessuno, e tutto quello che gli si lascia vedere è informazione regalata a chi sta provando a entrare nell'account di un altro.

Cosa	Come
La posizione vera	Messa da parte; la persona viene portata molto in alto, dove non c'è niente da guardare

Cosa	Come
L'inventario	Messo da parte e restituito dopo. Nessuno può buttare niente per terra né farsi passare oggetti
Gli altri giocatori	Non lo vedono e lui non vede loro
Chat e comandi	Spenti, tranne quelli di accesso
Danni e interazioni	Nessuna, in *nessuna* delle due direzioni
Movimento	Qualche blocco, così il client non "rimbalza" sullo schermo

> **SE IL SERVER SI SPEGNE MALE MENTRE QUALCUNO STA SCRIVENDO**
>
> La fotografia di posizione e inventario sta sul *database*, non in memoria, e viene scritta **prima** di svuotare l'inventario: se quella scrittura non riesce, l'inventario non viene toccato affatto. Al primo accesso dopo il riavvio torna tutto al suo posto.

4.4 La sessione

Chi rientra dallo stesso indirizzo entro mezz'ora non riscrive la password. È la funzione che decide se un plugin di accesso è amato o odiato: senza, chi ha una connessione instabile scrive la password dieci volte in una sera. È legata all'indirizzo apposta — senza quel vincolo la password servirebbe una volta sola per sempre.

4.5 Il dispositivo fidato — "ricordami" (Premium)

La sessione dura mezz'ora e cade quando cambia l'indirizzo. Il dispositivo fidato è la stessa idea portata avanti di un mese: chi ha scritto la password una volta su *questo* computer non la riscrive più. Dopo un accesso con password compare un pulsante:

```
[ RICORDAMI ]            un clic, e questo computer è ricordato
/rememberme status         quanti dispositivi ricorda il tuo account
/rememberme forget         li toglie tutti, subito
```

Sotto il pulsante non c'è niente di magico: il server dà al client un numero casuale di 32 byte, il *gettone*, e sul database ne tiene solo l'impronta SHA-256 — esattamente come per le password, un dump del database non fa entrare nessuno. Alla connessione successiva il client ripresenta il gettone, l'impronta torna, e la password non viene chiesta.

PERCHÉ NON È UN BUCO, IN CINQUE REGOLE

1. Il gettone si consuma a ogni uso. Ogni ingresso ne emette uno nuovo e brucia il vecchio. Una copia rubata funziona una volta sola, e solo se arriva *prima* del proprietario; se arriva dopo non funziona, e se arriva prima il proprietario si ritrova a riscrivere la password e capisce che qualcosa non va.
2. Lo emette solo la password. Non una sessione ripresa, non un accesso ufficiale, non un ingresso forzato dallo staff. Chi si siede al computer di un altro mentre è collegato non si porta via niente.
3. La fiducia muore quando cambia qualcosa: password cambiata, password reimpostata dallo staff, codice di recupero generato, `/logout`, account cancellato. Tutti i dispositivi, non solo quello in uso.
4. Sul disco non c'è mai il gettone, solo la sua impronta.
5. Il gettone non è la password. Non permette di cambiarla, non permette di leggerla, e non salta la verifica in due passaggi — il gettone dimostra che il computer è quello di sempre, non che davanti ci sia la persona giusta.

IL LIMITE ONESTO: DURA FINCHÉ IL GIOCO RESTA APERTO

Il client tiene i cookie in memoria, non su disco. Chi chiude Minecraft perde il gettone e riscrive la password. È una scelta del gioco, non nostra, e va detta ai giocatori: "ricordami" qui vuol dire *finché non chiudi il gioco*, non "per sempre". La scadenza in `config.yml` è il tetto massimo, non la durata che si vedrà nella pratica.

SE HAI UN PROXY (VELOCITY, BUNGEECORD, WATERFALL)

Il cookie appartiene alla connessione fra il client e l'indirizzo a cui si è collegato, che con un proxy è *il proxy*. Quindi tutti i server dietro allo stesso proxy vedono lo stesso cookie, e le tre situazioni possibili sono queste:

- **Database condiviso** (MySQL o MariaDB): il caso migliore. Ci si fida di un dispositivo una volta e vale per tutta la rete, anche cambiando server.
- **Database separati:** il gettone di un server non corrisponde a nessuna riga dell'altro e viene semplicemente ignorato. Nessuno entra di nascosto, nessuno resta fuori: si riscrive la password, come prima.
- **Vuoi che non si vedano nemmeno il cookie?** Dai a ogni server un `trusted-device.cookie-name` diverso.

Se il proxy è vecchio e non inoltra i cookie durante l'accesso, la risposta non arriva, scade `read-timeout` e si chiede la password. Non si rompe niente.

4.6 La verifica in due passaggi

Facoltativa, e la accende chi vuole sul proprio account, con qualunque app di autenticazione (Google Authenticator, Aegis, Bitwarden, 1Password). Serve ai pochi account che valgono qualcosa: lo staff, chi ha comprato gradi, chi ha una base piena di roba. Imporla a tutti la fa disattivare ai giocatori nel modo più semplice: andando a giocare da un'altra parte.

```
/2fa enable              genera il segreto e lo mostra una volta sola
/2fa confirm <codice>   la accende, dopo aver provato che funziona
/2fa disable <codice>     la spegne: serve un codice valido
```

Il secondo passaggio non è burocrazia: accendere la verifica senza aver provato un codice significa scoprire che l'orologio del telefono è sbagliato al primo accesso successivo, cioè quando ci si è già chiusi fuori. Insieme arrivano **otto codici di riserva**, ognuno valido una volta sola: sono per il giorno in cui il telefono si rompe.

4.7 Il recupero

Un server Minecraft non ha gli indirizzi di posta dei suoi giocatori e non deve averli. Quindi il recupero passa da quello che c'è:

- **una domanda** scelta dal giocatore, con una risposta che sul database è un hash come una password;
- **un codice dato dallo staff**, valido pochi minuti e una volta sola. Un operatore riconosce la persona (dal canale vocale, dal forum, da com'è fatta la sua base) e le consegna un gettone temporaneo.

> **NESSUNO PUÒ VEDERE UNA PASSWORD, NEMMENO UN OPERATORE**
>
> Non esiste da nessuna parte: sul database c'è solo un hash. E la strada prevista (`resetpassword`) genera un codice *da consegnare*, non entra nell'account. La differenza sembra sottile ed è tutta: la prima strada lascia una traccia e richiede che la persona faccia qualcosa, la seconda permetterebbe a un moderatore di prendersi l'account di chiunque.

4.8 L'antibot

Un attacco da bot non prova password: apre connessioni. Trecento client falsi al secondo, ognuno con un nome inventato, ognuno che occupa uno slot e una riga di database. Il server non cade per il carico di gioco — cade perché ogni connessione costa una lettura e una scrittura.

Difesa	Cosa ferma
Forma del nome	I generatori più banali, prima di qualunque altra cosa
Connessioni per indirizzo	Una botnet su un indirizzo solo (tre alla volta di serie)
Registrazioni per indirizzo	Gli account inventati: è l'unica cosa che *resta* dopo che il bot se n'è andato
Modalità stretta	L'ondata vera: per qualche minuto entra solo chi ha già un account

La modalità stretta **si accende da sola** sopra soglia e si spegne da sola dopo qualche minuto di calma. Perché un attacco arriva alle quattro del mattino, e un interruttore da premere a mano viene premuto venti minuti dopo — venti minuti sono abbastanza per riempire un database di diecimila account inventati. I tuoi giocatori non se ne accorgono nemmeno: hanno già un account, quindi entrano.

4.9 La schermata a clic

`/loginmenu` apre una tastiera numerica: si clicca, e sullo schermo compaiono dei pallini. Esiste perché un comando scritto in chat lo legge chiunque guardi lo schermo.

ONESTAMENTE, SU "CANCELLARE IL COMANDO DALLA CRONOLOGIA"

Nessun plugin può cancellare la cronologia locale del client: la freccia in su ripescherà comunque quello che è stato digitato, ed è un limite del gioco, non una dimenticanza. ShadowLogin fa le due cose che *può* fare: spinge subito la riga fuori dallo schermo, e — soprattutto — **non lascia che il comando arrivi al dispatcher del server**, che altrimenti lo scriverebbe in console. Agli altri plugin arriva una riga con dei pallini al posto della password. Se la cosa ti preoccupa davvero, la risposta è la tastiera: lì la password non passa mai da un canale che qualcun altro può leggere.

4.10 L'esportazione

`/shadowlogin export` scrive tutti gli account in un CSV che si apre con un foglio di calcolo. Dentro c'è l'hash BCrypt — quello che serve per non far rifare la registrazione a nessuno se un giorno cambi plugin. **Non** c'è il segreto della verifica in due passaggi: quello aprirebbe gli account, e chi migra rifà quel pezzo in trenta secondi per persona.

IL FILE ESPORTATO VA TRATTATO COME UN BACKUP DEL DATABASE

Gli hash non sono password e non ci si può risalire, ma chi ha quel file può provarci offline quanto vuole. Non lasciarlo in una cartella condivisa e non allegarlo a una richiesta di assistenza.

5. Gli account ufficiali

È la funzione per cui si compra questo plugin, ed è anche quella dove è più facile sbagliare in modo invisibile. Questo capitolo spiega come funziona davvero e cosa dipende da come è configurata la tua rete.

5.1 La domanda ovvia, che è quella sbagliata

La domanda ovvia è *"questo nome appartiene a un account Minecraft comprato?"*. Chiederlo a Mojang è facile, e da sola quella risposta **non basta e non va mai usata per far entrare qualcuno**: che esista un account chiamato Marco non dice niente su chi sta bussando adesso. Un plugin che facesse entrare senza password chiunque scriva un nome ufficiale sarebbe il modo più rapido di regalare tutti gli account comprati del server.

5.2 La domanda giusta

Questa sessione è già stata autenticata da Mojang? Se la risposta è sì, la password l'ha già verificata qualcun altro e chiederla di nuovo è solo una seccatura. E si può rispondere con dei fatti, senza toccare i pacchetti:

1. **Il server è in `online-mode`.** Allora chi è entrato ha superato la verifica di Mojang: è il server stesso ad averla fatta.
2. **L'identificativo della sessione è quello vero di Mojang** per quel nome. Su un server senza autenticazione l'identificativo viene calcolato dal nome e non coincide mai con quello vero. Se invece coincide, vuol dire che arriva da monte — dal proxy che ha autenticato la persona — perché il client non ha nessun modo di sceglierselo.

> **IL SECONDO CASO DIPENDE DA COME È CONFIGURATA LA TUA RETE**
>
> Se il trasferimento dell'identità dal proxy non è protetto — la porta del server aperta al mondo, oppure il vecchio inoltro di BungeeCord senza segreto condiviso — chiunque può collegarsi *direttamente* al server dichiarando l'identificativo che vuole, e questo controllo diventa inutile.
>
> Non è un difetto di ShadowLogin e nessun plugin lo può riparare da dentro: è un server che si può scavalcare. Con Velocity usa il *modern forwarding*; con BungeeCord metti il server dietro un firewall che accetta solo il proxy. Se non sei sicuro della tua rete, metti `premium.trust-proxy-authentication: false` : il riconoscimento continuerà a funzionare in `online-mode` e la protezione dei nomi resterà comunque attiva.

5.3 A cosa serve allora la risposta di Mojang

Due cose utili, e nessuna delle due è far entrare qualcuno:

- **Protegge i nomi.** Un nome che appartiene a un account vero non si può registrare da una sessione non autenticata. Così nessuno si prende il nome di un altro prima che arrivi il proprietario, che è il furto più comune su un server misto.
- **Fornisce l'identificativo vero**, che serve al confronto del punto 2.

5.4 Quando Mojang non risponde

Succede: hanno avuto interruzioni di ore. La regola è una sola e non si discute: **non si blocca mai l'ingresso**. Si usa quello che dice la cache (tre giorni di serie) e, se non c'è niente in cache, si chiede la password — che è la strada normale per tutti gli altri e non fa male a nessuno.

5.5 L'identificativo: il capitolo da leggere due volte

Lo stesso nome può arrivare con **due identificativi diversi**. Chi ha comprato il gioco porta con sé quello di Mojang; chi non l'ha comprato ne riceve uno calcolato dal nome, e i due non coincidono mai. Minecraft tiene inventario, progressi e posizione *per identificativo*.

identity.mode	Cosa fa	Per chi
`UUID` predefinito	Un account per identificativo. Il "Marco" ufficiale e il "Marco" non ufficiale sono due account distinti, con due password e due inventari — che è anche quello che il server farà comunque.	Chi parte da zero
`NAME`	Un account per nome, come AuthMe. L'identificativo della registrazione viene comunque salvato, e se un giorno ne arriva un altro con lo stesso nome si applica `on-uuid-mismatch`.	Chi importa da AuthMe

In modalità `NAME`, cosa fare quando il nome è quello giusto ma l'identificativo no:

on-uuid-mismatch	Cosa succede
`DENY` predefinito	La connessione viene rifiutata e in console resta scritto. È esattamente il caso pericoloso
`ASK_PASSWORD`	Può provare con la password, senza nessuna scorciatoia: niente sessione, niente accesso automatico
`ALLOW`	Il nuovo identificativo diventa quello dell'account. Solo per una migrazione, e per il tempo che serve

> **PER CAPIRE COSA STA SUCCEDENDO A UN NOME**
>
> `/shadowlogin premium <nome>` mostra se quel nome appartiene a un account vero, qual è il suo identificativo di Mojang e quale sarebbe quello non ufficiale. È il comando da usare quando un giocatore dice "non riesco a entrare" e la spiegazione non è ovvia.

6. Comandi

> **NEI MESSAGGI I COMANDI NON SONO MAI SCRITTI A MANO**
>
> Se `/login` era già di un altro plugin, il nostro si chiama `/shadowlogin:login`, e i messaggi del plugin useranno *quello*. All'avvio la console scrive quali alias sono andati persi. Qui sotto i comandi sono nella forma preferita, cioè quella che si ottiene su un server dove non c'è nessuno a contenderli.

6.1 I comandi dei giocatori

Comando	Alias	A cosa serve
`/login <password>`	`/l`, `/signin`	Entrare
`/register <pw> <pw>`	`/reg`	Creare il proprio account
`/changepassword <vecchia> <nuova>`	`/changepass`, `/passwd`	Cambiare la password
`/logout`	`/signout`	Chiudere la sessione
`/2fa <codice>`	`/totp`	Il codice dell'app
`/recover <codice> <nuova>`	`/recovery`	Recuperare l'accesso
`/loginmenu`	`/logingui`	La schermata a clic
`/rememberme [status|forget]`	`/trustdevice`	Ricordare questo computer

Nessuno di questi ha un permesso: sono i comandi che si usano *prima* di essere qualcuno, e un permesso li renderebbe inutilizzabili proprio a chi ne ha bisogno. L'unica eccezione è `/rememberme`, che si usa *dopo* l'accesso e ha quindi un permesso vero — acceso per tutti, ma toglibile a chi ha più da perdere.

6.2 I comandi dello staff

Radice `/shadowlogin`, alias `/slogin`, `/shadowauth`, `/slg`. Permesso di base `shadowlogin.admin`.

Comando	A cosa serve
`/shadowlogin help`	L'elenco, in gioco
`/shadowlogin info <giocatore>`	Tutto quello che sappiamo di un account
`/shadowlogin status`	Account, accessi, antibot, licenza, archivio
`/shadowlogin accounts [pagina]`	L'elenco degli account
`/shadowlogin unregister <g>`	Cancella un account. Chiede conferma a pulsanti
`/shadowlogin setpassword <g> <pw>`	Cambia una password. Lascia una riga nel registro

Comando	A cosa serve
`/shadowlogin resetpassword <g>`	Genera un codice di recupero da consegnare
`/shadowlogin forcelogin <g>`	Lo fa entrare a mano
`/shadowlogin logout <g>`	Lo rimette in attesa
`/shadowlogin session clear <g>`	Annulla la sua sessione
`/shadowlogin 2fa disable <g>`	Spegne la sua verifica in due passaggi
`/shadowlogin premium [refresh] <nome>`	Chiede a Mojang, e mostra i due identificativi
`/shadowlogin trust <g> [revoke]`	I suoi dispositivi ricordati, e il modo di toglierli
`/shadowlogin history [giocatore]`	Il registro degli accessi
`/shadowlogin antibot [on|off]`	Modalità stretta, a mano
`/shadowlogin export [file]`	Esporta gli account in CSV
`/shadowlogin import <file>`	Importa da un CSV. Chiede conferma
`/shadowlogin reload`	Rilegge configurazione e lingue
`/shadowlogin license`	Stato della licenza

LE CONFERME SONO PULSANTI

Cancellare un account o spegnere la verifica in due passaggi di qualcuno sono cose che non si annullano. Nessun comando va riscritto per confermare: arrivano due pulsanti in chat, **[CONFERMA]** e **[ANNULLA]**. Il gettone dentro al pulsante è casuale e vale due minuti, quindi un clic per sbaglio su un messaggio di ieri non fa niente.

7. Permessi

Permesso	Predefinito	A cosa dà diritto
`shadowlogin.admin`	op	I comandi di amministrazione
`shadowlogin.reload`	op	Ricaricare configurazione e lingue
`shadowlogin.admin.license`	op	Vedere e attivare la licenza
`shadowlogin.admin.accounts`	op	Vedere l'elenco degli account
`shadowlogin.admin.password`	op	Cambiare la password di un altro
`shadowlogin.admin.export`	op	Esportare e importare
`shadowlogin.admin.trust`	op	Vedere e togliere i dispositivi ricordati di un altro
`shadowlogin.twofactor`	tutti	Accendere la verifica in due passaggi sul proprio account
`shadowlogin.trusteddevice`	tutti	Far ricordare il proprio dispositivo
`shadowlogin.*`	op	Tutti quelli amministrativi

NON C'È UN PERMESSO PER SALTARE L'ANTIBOT

E non è una dimenticanza: quei controlli avvengono *prima che il giocatore esista*, quando il server non ha ancora nessun oggetto a cui chiedere se ha un permesso. Un permesso del genere sarebbe scritto nella tabella e non funzionerebbe mai. Se hai bisogno che un indirizzo non venga contato, alza i limiti: sono per indirizzo, non per persona.

Perché i permessi non valgono prima dell'accesso

Non è che glieli togliamo: è che *non c'è niente che possa usarli*. Prima dell'accesso ogni comando che non sia di accesso viene annullato, la chat è spenta e nessun evento passa. Se ShadowPermissions è installato, i permessi vengono **ricalcolati dopo** l'accesso, mai prima.

ATTENZIONE A CHI DÀ I PERMESSI CON UN PLUGIN CHE NON SA DELL'ACCESSO

Se usi un plugin di permessi che assegna il grado all'ingresso e un altro plugin reagisce a quel momento, tieni presente che per ShadowLogin l'ingresso vero è *dopo* la password. I plugin della suite lo sanno; per gli altri, il segnale da ascoltare è `PlayerAuthenticateEvent`.

8. Dipendenze e integrazioni

8.1 Cosa serve

Niente. ShadowLogin funziona su un server dove non c'è nient'altro installato. Il pool di connessioni, il driver MariaDB, il database H2 e la libreria BCrypt viaggiano dentro al jar sotto un nome di package nostro: sul classpath del server non compare niente, e non c'è nessun conflitto possibile con nessun altro plugin.

8.2 Cosa succede se c'è il resto della suite

Plugin	Cosa cambia
ShadowBasics	Il teletrasporto allo spawn avviene *dopo* l'accesso, e solo al primo accesso di sempre. Prima significherebbe mostrare lo spawn a chi non ha ancora dimostrato di essere chi dice.
ShadowPermissions	I permessi vengono ricalcolati dopo l'accesso, mai prima.
ShadowControl	Le registrazioni sospette, i tentativi ripetuti, le ondate di bot e i problemi di configurazione del proxy diventano segnalazioni nel pannello. Se l'assistente è attivo, può riassumere allo staff cosa è successo. Nessuna password e nessun segreto finiscono lì dentro: solo cosa è successo e a chi.

Ogni integrazione è facoltativa e sta dietro un controllo di presenza. Quando un pezzo si spegne perché manca un altro plugin, in console c'è scritto una volta sola cosa manca e cosa si è spento di conseguenza.

8.3 Per chi scrive plugin

ShadowLogin lancia `PlayerAuthenticateEvent` quando qualcuno è entrato davvero — dopo la password e dopo che inventario e posizione sono tornati al loro posto. È il momento in cui gli altri plugin devono fare quello che di solito farebbero all'ingresso.

```
@EventHandler
public void onAuth(PlayerAuthenticateEvent event) {
    Player player = event.getPlayer();
    // event.kind(): PASSWORD, SESSION, PREMIUM, FORCED
    // event.firstTime(): vero se l'account e' appena nato
}
```

8.4 Con chi non convive

Con nessun altro plugin di accesso: AuthMe, nLogin, JPremium, LoginSecurity, FastLogin, LibreLogin e simili. Se ne trova uno, ShadowLogin **non gestisce l'ingresso** e lo scrive in console — i comandi di amministrazione restano disponibili, così puoi esportare i tuoi account prima di decidere.

FASTLOGIN MERITA UNA NOTA

Non è un plugin di accesso: è un'aggiunta ad AuthMe che fa il riconoscimento degli account ufficiali. Con ShadowLogin non serve — quel lavoro lo facciamo noi — e tenerli tutti e due significa due plugin che decidono la stessa cosa. Per questo è in elenco.

9. Configurazione

Ogni valore ha già il numero giusto per un server normale: si può copiare il jar e non aprire mai `config.yml`. Qui sotto c'è cosa cambia quando lo si apre.

9.1 Sicurezza

Chiave	Predef.	Cosa fa
`security.hash.cost`	12	Il costo di BCrypt. Ogni +1 raddoppia il tempo. 12 ≈ 250 ms
`security.hash.rehash-on-login`	true	Ricalcola col costo nuovo al primo accesso
`security.password.min-length`	6	Lunghezza minima
`security.password.max-length`	64	Oltre i 72 byte BCrypt ignora il resto
`security.password.forbidden`	elenco	Le password che non si possono usare. Aggiungi il nome del server
`security.attempts.max`	5	Tentativi prima dell'espulsione
`security.attempts.backoff-base`	5s	L'attesa dopo il primo errore, poi raddoppia
`security.attempts.lock-after`	10	Dopo quanti errori l'account si blocca
`security.attempts.lock-duration`	30m	Per quanto
`security.login-timeout`	60s	Quanto tempo c'è per entrare

9.2 Account ufficiali

Chiave	Predef.	Cosa fa
`premium.enabled`	true	Spegne tutto il capitolo
`premium.mode`	AUTO	`AUTO` o `OFF`
`premium.trust-proxy-authentication`	true	Fidarsi dell'identificativo che arriva dal proxy
`premium.force-password`	false	Chiedere la password anche a chi è già autenticato
`premium.protect-names`	true	Un nome ufficiale non si registra da una sessione non autenticata
`premium.cache-hours`	72	Per quante ore ricordare la risposta di Mojang

Chiave	Predef.	Cosa fa
`premium.on-lookup-failure`	CACHE_THEN_PASSWORD	Cosa fare quando Mojang non risponde

9.3 Identità, sessione, limbo

Chiave	Predef.	Cosa fa
`identity.mode`	UUID	Vedi capitolo 5.5
`identity.on-uuid-mismatch`	DENY	Vedi capitolo 5.5
`session.duration`	30m	Quanto dura la sessione
`session.require-same-ip`	true	Legarla all'indirizzo
`limbo.use-spawn`	true	Usare lo spawn del mondo, alzato
`limbo.location.y`	320	A che altezza aspetta chi non è ancora entrato
`limbo.radius`	3	Quanti blocchi ci si può muovere
`limbo.hide-inventory`	true	Mettere da parte l'inventario

9.4 Il dispositivo fidato

```
trusted-device:
  enabled: true
  automatic: false          false = pulsante, true = si emette da solo
  duration: "30d"           il tetto massimo, non la durata reale (vedi 4.5)
  max-per-account: 3        quanti computer può ricordare un account
  cookie-name: "trust"      cambialo per isolare i server dietro a un proxy
  skips-two-factor: false   da lasciare a false
  read-timeout: "1500ms"    quanto aspettare la risposta del client
```

`automatic: false` è il valore consigliato: la fiducia si dà, non si subisce, e un giocatore che non ha mai chiesto di essere ricordato non deve scoprire di esserlo. `max-per-account` esiste perché senza un tetto chi entra da venti postazioni lascerebbe venti chiavi valide in giro; quando arriva la ventunesima, la più vecchia esce. `read-timeout` è corto apposta: il gettone si chiede *durante* la connessione, e un'attesa lunga terrebbe fermo un thread di rete per ogni persona che entra.

9.5 Antibot e proxy

Chiave	Predef.	Cosa fa
`antibot.max-connections-per-ip`	3	Connessioni contemporanee dallo stesso indirizzo
`antibot.max-registrations-per-ip`	2	Registrazioni per indirizzo dentro la finestra
`antibot.join-rate.connections`	8	Sopra questa soglia scatta la modalità stretta

Chiave	Predef.	Cosa fa
`antibot.adaptive.duration`	5m	Quanto dura
`antibot.name-pattern`	3-16	La forma di un nome. Da cambiare se usi Geyser
`proxy.use-forwarded-address`	true	Usare l'indirizzo vero che il proxy dichiara

SE HAI GIOCATORI DA BEDROCK (GEYSER)

I nomi che arrivano da un ponte Bedrock hanno un prefisso (di solito un punto) e possono essere più lunghi di sedici caratteri: con l'espressione predefinita verrebbero rifiutati dall'antibot. Usa `antibot.name-pattern: "^[.A-Za-z0-9_ ]{3,20}$"`.

9.6 Come si entra

Chiave	Predef.	Cosa fa
`login.input`	BOTH	`CHAT`, `GUI` o `BOTH`
`login.gui.open-on-join`	false	Aprire la tastiera da sola
`login.gui.pin-length`	6	Quante cifre ha il PIN
`login.hide-join-message-until-login`	true	Annunciare l'ingresso solo dopo la password

10. Domande frequenti

Un giocatore dice che la password è giusta ma non entra

Nell'ordine: le maiuscole (la password le distingue, il nome no); un'attesa in corso per gli errori precedenti; l'identificativo, se sei in modalità `NAME` e qualcosa è cambiato nella rete. `/shadowlogin info <nome>` e `/shadowlogin premium <nome>` rispondono quasi sempre in due righe.

Ho dimenticato la password del mio account amministratore

Dalla console del server: `/shadowlogin resetpassword <tuonome>`. Genera un codice valido pochi minuti; poi in gioco `/recover <codice> <nuova>`. La console non passa da nessuna verifica di accesso, quindi funziona anche se sei chiuso fuori.

Posso vedere la password di un giocatore?

No, e non perché sia vietato: *non esiste da nessuna parte*. Sul database c'è solo un hash, e da un hash BCrypt non si torna indietro. Si può solo reimpostarla.

Perché a un giocatore non viene chiesta la password?

Tre possibilità: ha l'account originale e la sessione è già stata autenticata (capitolo 5); ha una sessione ancora valida dallo stesso indirizzo (mezz'ora di serie); oppure il server è in `online-mode`, e allora la verifica l'ha fatta Mojang per tutti.

Il server è dietro un proxy e tutti risultano dallo stesso indirizzo

L'inoltro dell'indirizzo reale non è configurato. Con Velocity serve il *modern forwarding* più `proxies.velocity.enabled: true` in `paper-global.yml` con il segreto giusto; con BungeeCord serve `ip_forward: true` **e** un firewall che lasci arrivare solo il proxy. Finché resta così, alza i limiti dell'antibot o non li potrai usare.

Un giocatore dice che "ricordami" non funziona più

Nell'ordine, sono queste tre le cause quasi sempre: **ha chiuso il gioco** (il client tiene i cookie in memoria, e chiudendolo il gettone sparisce — è il limite spiegato al paragrafo 4.5); **ha cambiato password**, o gliel'ha reimpostata lo staff, e in quel momento tutti i suoi dispositivi vengono dimenticati apposta; oppure **è entrato da un quarto computer** e il più vecchio è uscito per fare spazio. Nessuno dei tre è un guasto. `/shadowlogin trust <nome>` dice quanti dispositivi risultano adesso.

Ho attivato la verifica in due passaggi e ho perso il telefono

I codici di riserva consegnati all'attivazione: ognuno vale una volta sola. Se non li hai più, un operatore può spegnerla con `/shadowlogin 2fa disable <nome>`.

Voglio cambiare plugin di accesso

`/shadowlogin export`. Il file esce in `plugins/ShadowLogin/export/` con dentro gli hash BCrypt, che quasi tutti i plugin seri sanno leggere. Nessuno dei tuoi giocatori dovrà rifare la registrazione.

Quanto costa in prestazioni?

A server acceso, praticamente zero: non c'è nessun compito periodico che guardi il mondo, e l'unico listener sul movimento esce alla prima riga quando il blocco non è cambiato. Il costo sta tutto nel momento dell'accesso — un calcolo BCrypt e una query — e avviene fuori dal thread di gioco.

Funziona su Folia?

Sì. Ogni operazione gira sul thread giusto: il giocatore si tocca dal proprio, il database e gli hash da un thread asincrono, e nessun compito tocca il mondo fuori dal suo scheduler.

Il plugin è partito con dei giocatori già collegati

Succede con una ricarica a caldo. ShadowLogin li rimette in attesa e chiede a tutti l'accesso: se non lo facesse, resterebbero dentro senza aver mai scritto niente. In console c'è scritto quanti erano.

> **UNA COSA DA FARE UNA VOLTA SOLA**
>
> Aggiungi il nome del tuo server a `security.password.forbidden`. È sempre nella lista di chi prova le password a caso, ed è la prima che indovina.