



Fehér Könyv (technikai)

[www.balabit.com](http://www.balabit.com)

# Single-sign-on autentikáció Zorp tűzfalon

**Kivonat:** A hálózati jogosultsági rendszerek kihívásai  
**Verzió:** 1.1  
**Dátum:** 2005-08-19



## Single-sign-on autentikáció

### A hálózati jogosultsági rendszerek kihívásai

A vállalati folyamatok magas szintű elektronizálása után – melyet a hatékonyság növelése hajtott előre – rövid idővel megjelent az ezt megvalósító rendszerek biztonsága iránti igény is, mely legalább azt a biztonsági szintet követelte meg az elektronikus adattárolás és elérés területén, amit a papíralapú rendszerek nyújtottak. Az összetett jogosultsági rendszerek bevezetése után azonban az informatikai vezetők ijedten tapasztalták, hogy bizonyos üzemi folyamatok hatékonysága drámaian visszaesett; akár a papír alapú rendszerek szintje alá is.

Ennek okai legfőképp a hálózati szolgáltatások egyedi autentikációs rendszereiben keresendők. A fájl szerver eléréséhez, a levelek elolvasásához, a nyomtatáshoz, az adatbázisokhoz és a különböző intranetes alkalmazásokhoz mind külön azonosítania kell magát a felhasználónak. Ráadásul, amennyiben egy rendszergazda szigorúan veszi a biztonság alapvető szabályait, ragaszkodnia kell ahhoz is, hogy ezek a jelszavak különbözőek legyenek. Sőt, egyes szolgáltatásokhoz erős autentikációt is megkövetelhet, például egy token behelyezését az USB foglalatba.

Minél több hálózati szolgáltatást tartalmaz egy üzemi folyamat, annál jelentősebb mértékben rontja a hatékonyságát a független biztonsági rendszerekből adódó hibák lehetősége. Egy lejárt, vagy elfelejtett jelszó; egy nem megfelelően beállított vagy nem időben frissített jogosultság; egy véletlen vagy szándékosan kizárt felhasználó hosszú órákra megakaszthatja a munkavégzést. Mindez olyan fokú bizonytalanságot visz egy összetett rendszerbe, ami éves szinten komolyan csökkentheti a termelékenységet.

Égetően szükségessé vált tehát egy vállalati környezetre kifejlesztett központi autentikációs séma.

### Hogyan is működött ez „régén”, a papír alapú irodákban?

Mielőtt valakit felvettek dolgozni, a vezető egy felvételi beszélgetésen alaposan megvizsgálta, alkalmas-e ez az ember a feladatra (ez persze most is így működik). Amennyiben az illető felvételt nyert, bemutatták a kollégáknak és (írásban is) megjelölték a munkaköréhez tartozó jogokat. Tehát egy hatóságként működő személy – vezető - autentikálta a dolgozót és definiálta a jogait.

Az erőforrásokat (pl. anyagraktár, irattár, kassza, ...) pedig erőforrás-felelősök kezelték, akik a vezetői utasítás szerint engedtek hozzáférést ezekhez az eszközökhöz.

Nagyobb vállalatok esetén a dolgozók írásbeli engedélyt kértek a főnöktől az erőforrások használatához, majd ezzel a papírral lementek például a raktárba, és felvették a munkájukhoz szükséges anyagokat.

### És hogyan működik „ma”?

Minden bizonnyal ezt a való életből vett metódust tartották szemelőt a MIT (Massachusetts Institute of Technology) azon kutatói, akik a Kerberos hálózati autentikációs protokollt kifejlesztették. A Kerberos rendszerekben a KDC (Key Distribution Center) alaposan megvizsgálja az új felhasználót (erős autentikáció), ez után azonban többé már nem kell a felhasználónak azonosítania magát. A hálózaton elérhető szolgáltatásokat a KDC által kiállított megbízólevelek (ticket) alkalmazás szervereknek való bemutatásával érheti el.



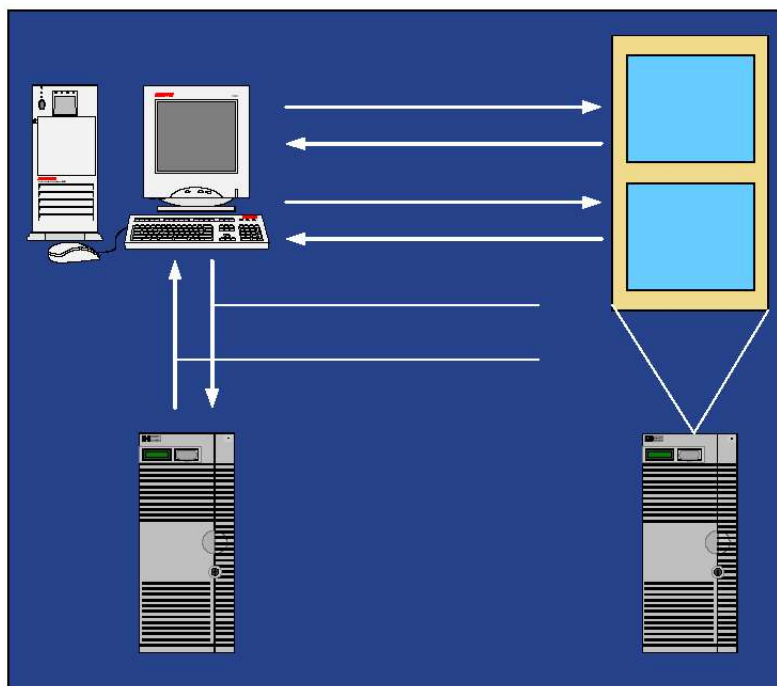
## Mi az a Kerberos?

A Kerberos egy hálózati autentikációs protokoll, amely erős központi hitelesítést tesz lehetővé olyan folyamatok esetében, amikor a kliens egy szerveren futó alkalmazást vagy szolgáltatást szeretne használni. A gyakorlatban ez azt jelenti, hogy a felhasználónak csak egyszer kell magát azonosítania ahhoz, hogy elérje a hálózat valamennyi szolgáltatását. Akkor, amikor bejelentkezik a központi kulcselosztóba (KDC – Key Distribution Center). Ezt követően már csak a KDC-től kapott megbízólevelet kell a szervereknek bemutatnia. Mivel ez a megbízólevél csupán az adott munkafolyamatra érvényes, és hamar le is jár, a módszer jóval nagyobb biztonságot nyújt a hagyományos jelszónál.

**Kerberos** (latinul Cerberus) a görög mitológia szerint Hádésznek – az alvilág istenének - háromfejű kutyája, aki az alvilág kapuját őrzi. Az ő elfogása volt az utolsó Herakles (latinul Herkules) tizenkét bajvívásából.

### A Kerberos főbb előnyei, hogy:

- a felhasználónak csak egyszer kell magát azonosítania, valamint, hogy
- a felhasználó jelszava nem közlekedik a hálózaton; és
- a felhasználó biztonságosan azonosíthatja a szervert.



Kerberos protokollt használó autentikációs rendszerek tulajdonképp minden platformra készültek már. Unixon (így Linuxon is) és Windowson is már jó ideje elérhető.

## A Kerberos autentikációs folyamat

1. A felhasználó autentikálja magát a saját gépén futó Kerberos kliensnek. Az azonosítás technológiája szinte bármi lehet az egyszerű jelszótól kezdve a tokenen át az újlenyomat érzékelésig. Ez az egyetlen alkalom a folyamatban, amikor felhasználói beavatkozás szükséges.
2. A kliens az egyes pontban megszerzett jelszót titkosító kulccsá (*long term key*) konvertálja egy nem visszafejthető algoritmussal (Hash), majd eltárolja.
3. A kliens egy időpecsétet küld a központi kulcselosztó (KDC) szervernek, amelyet szimmetrikusan titkosít



a *long term key* használatával.

4. A felhasználó jelszavából képzett *long term key* a KDC-ben is el van tárolva, amivel az visszafejti az időpecsétet. Az autentikáció akkor sikeres, ha a visszafejtett időpecsétben értelmezhető adatok vannak, és az időkülönbség is kisebb a tolerancia küszöbnél.

5. A KDC ekkor generál a felhasználó számára egy *logon session key*-t, ami a rendszerből való kijelentkezésig érvényes. Ezt ugyancsak szimmetrikus titkosítással (az előbb is használt *long term key* használatával) elküldi a kliensnek, amely eltárolja. A továbbiakban a kliens-szerver kommunikáció már ezzel az új kulccsal lesz titkosítva.

6. Amikor a kliens egy alkalmazásszerverhez kíván kapcsolódni a 3-5 pontban leírt folyamat megismétlődik, minek eredményképp a kliens kap a KDC-től egy ticketet, ami többek között egy *service session key*-t tartalmaz. Ezt a kulcsot a ticket valójában kétszer is tartalmazza. Egyszer a kliens *logon session key*-ével titkosítva, másodszer pedig az aktuális alkalmazásszerver *long term key*-el titkosítva.

Hogy miért van erre szükség? Mivel a Kerberos protokollban a biztonságos kommunikációt szimmetrikus titkosítás garantálja, a szervernek ismernie kell azt a *service session key*-t, amivel a kliens szeretne vele kommunikálni. Ez biztonságosan csakis a KDC-től származhat az alkalmazásszerver *long term key*-ével titkosítva.

7. Tehát, amikor a kliens a szerverhez fordul, hogy kapcsolódjon a szolgáltatáshoz, elküld egy időpecsétet az aktuális *service session key*-el titkosítva, és elküldi magát a *service session key*-t is a szerver *long term key*-ével titkosítva (utóbbit a kliens nem tudja értelmezni, egyszerűen tovább küldi, ahogy a KDC-től kapta).

8. A szerver először dekódolja a *service session key*-t, majd ezzel a kulccsal az időpecsétet is. Amennyiben az időpecsét értelmezhető és megfelelően „friss”, a szerver autentikálta a klienst.

9. A legtöbb esetben itt a Kerberos szerepe lezárult, és a szolgáltatás által forgalmazott adatok titkosítás nélkül vagy egy SSL csatornán keresztül haladnak. Ám a Kerberos nem csak az autentikációt képes kezelni, de segítségével a teljes kliens-szerver forgalom titkosítása is lehetséges.

Ez kétféleképp érhető el: Egyrészt a Kerberos integrálódhat az SSL-be, amikor is az SSL-ben használt tanúsítvány helyébe egy Kerberos szimmetrikus titkosító kulcs lép (session key). Másrészt, a Kerberos képes ezt a kulcsot átadni az alkalmazásnak is, hogy az még magas szinten titkosítsa az adatot. Utóbbi módszert meglehetősen ritkán használják, hiszen erre az alkalmazásokat külön fel kell készíteni.

## Kerberos a tűzfalon

A Kerberos több évtizedes múltja után ma már nem igen találunk olyan hálózati szolgáltatást, amely nem alkalmas arra, hogy Kerberossal autentikálja a hozzá kapcsolódó klienseket. Egy azért mégis van: a hálózati tűzfal. Sajnos látszólag minden megmagyarázható ok nélkül eddig szinte még egy gyártó sem vette a fáradságot arra, hogy a tűzfalon keresztülhaladó felhasználókat Kerberos segítségével azonosítsa. Az ok valószínűleg az, hogy vállalati környezetben leginkább elterjedt Windows operációs rendszer mindössze a Windows 2000 óta tartalmaz Kerberos klienst, valamint a cégek is csak manapság kezdik felismerni a tűzfalon történő felhasználó-azonosítás fontosságát.

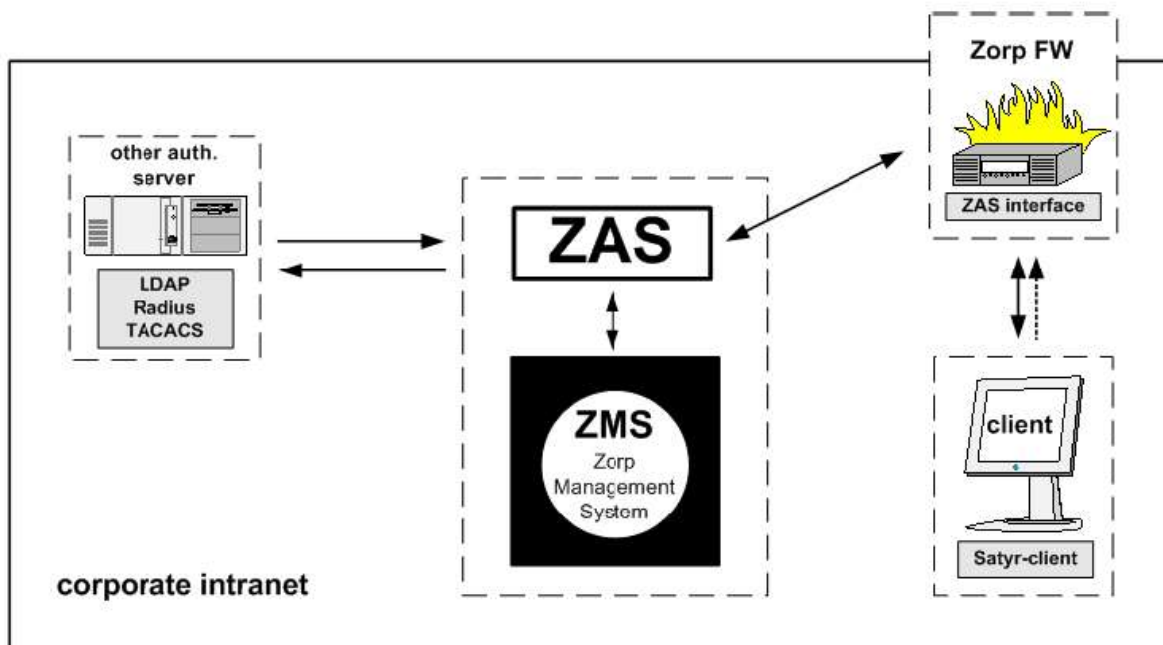
Pedig ez utóbbi rendkívüli módon növeli a hálózati biztonságot, hiszen a támadók a támadások jelentős részét valamilyen kémprogrammal készítik elő, amik érzékeny adatokat (főleg jelszavakat, hitelkártya számokat, stb.) juttatnak ki a belső hálózatról. Amennyiben az Internetre nyíló tűzfalon csak autentikált felhasználók juthatnak keresztül, a kémprogramok mozgástere drámaian leszűkül.

Valójában, a fenti védelmet kizárólag protokollon kívüli (out of band) autentikációval lehet megvalósítani, melyre a jelenleg kapható tűzfalak csak elenyésző hányada alkalmas. Így azon sem csodálkozhatunk, hogy ebből a maroknyi termékből ma még egyik sem képes Kerberossal együttműködni.

## A Zorp autentikációs rendszere

A Zorp igen fejlett out-of-band autentikációs rendszerrel rendelkezik, mely a következő elemekből áll:

- **Satyr kliens**, mely a felhasználó gépén fut. Feladata, hogy felugró ablakban azonosításra szólítsa fel a felhasználót, majd a megszerzett adatokat biztonságos csatornán eljuttassa a tűzfalnak (Zorp).
- **ZAS interface**, amely a tűzfal szerves része. Feladata, hogy kommunikáljon a ZAS-sal és a Satyr klenssel.
- **ZAS** (Zorp Authentication Server), amely szegmentált (több tűzfalat is tartalmazó) hálózatban célszerűen a központi menedzsment szerverrel fut egy hardveren, de attól független. Feladata, hogy a tűzfal kérésére autorizálja a felhasználót, vagyis azonosítsa és állapítsa meg a jogosultságát a kért szolgáltatásra. Az ehhez szükséges adatbázist vagy a ZAS tartalmazza XML formátumban, vagy harmadik szervertől kéri el (LDAP, RADIUS, Kerberos KDC, stb.).
- **ZMS** (Zorp Management Server), amely a központi menedzsment szerver. Több Zorp tűzfal egy felületről történő kezelésére szolgál.



### Az autentikáció folyamata pedig a következő:

1. A kliens kérést intéz a tűzfalhoz az Interneten található valamely szerver eléréséhez egy bizonyos protokollon keresztül.
2. A tűzfal felszólítja a Satyr klienst, hogy kérje be a felhasználó adatait. A Satyr több módon is megteheti ezt. Képes jelszót kérni, tokenel (vagy akár biometrikus eszközzel) együttműködni, és képes a gépen futó Kerberos klienssel is kommunikálni.
3. A Satyr a megszerzett adatokat továbbítja a tűzfalnak, aki pedig a ZAS szervernek küldi azokat tovább.
4. A ZAS azonosítja a felhasználót, és ezt a személyazonosságot igazolja a tűzfalnak.
5. A tűzfal saját szabálylistája alapján eldönti, jogosult-e a felhasználó a megjelölt szerver eléréséhez.



Kérdéseivel, észrevételeivel keresse meg cégünket az alábbi címen: BalaBit IT Security, 1116 Budapest Csurgói út 20/B  
Telefon: 06 1 371 0540 Fax: 06 1 208 0875 E-mail: [info@balabit.hu](mailto:info@balabit.hu) Web: <http://www.balabit.hu>

© 2005 BalaBit IT Security. Minden jog fenntartva.

A dokumentumra vonatkozó jogi kitételeket a következő weboldal tartalmazza:  
[http://www.balabit.com/products/zorp/docs/legal\\_notice.bbq](http://www.balabit.com/products/zorp/docs/legal_notice.bbq)