XVI. FIATAL MŰSZAKIAK TUDOMÁNYOS ÜLÉSSZAKA

Kolozsvár, 2011. március 24–25.

GNU/LINUX HARDENING

FRIED Zoltán

Abstract

The linux hardening is a process of making linux even more secure. There are many ways to do that - from tweaking network policies to installing and configuring softwares and kernel security modules. For a desktop pc or a small home server, you don't need to worry about that since linux is already secure enough by default.

Key words:

Linux, hardening, security

Összefoglalás

A linux hardening egy olyan biztonságtechnikai folyamat, amely azt a célt szolgálja, hogy a linux szerverünk még biztonságosabb legyen. Az, hogy ebbe mi tartozik bele, és mi nem, már egy másik probléma. Ezért az első feladat mindig az, hogy meghatározzuk, hogy mit jelent számunkra a biztonság.

Kulcsszavak:

Linux, biztonságtechnika

1. Bevezetés, alapfogalmak

1.1 Az informatikai biztonság

Az informatikai elemek védelme érdekében a megelőző és javító eljárások összessége, a jogosulatlan hozzáféréseken vagy használaton keresztül érkező támadásokkal szemben. Nincs teljes - 100%-os biztonság, ezért az adott rendszer biztonságát az un. tudatos kockázatvállalással jellemezhetjük. A védekezőknek minden lehetséges támadási pontot védenie kell, míg a támadónak elég egyetlen támadható pontot találnia, ezért a biztonság egyben folyamatos és konzisztens éberséget jelent a rendszer minden pontján.

1.2 Az adatvédelem és adatbiztonság

Az adatvédelem az egyén adatainak a védelme, míg az adatbiztonság az adatok fizikai és logikai biztonságát jelenti a különböző fenyegetettségekkel szemben.

1.3 Kockázatelemzés és kockázatkezelés

Az informatikai rendszerek támadásoknak vannak kitéve. A támadások különböző forrásból a sebezhetőségeket keresve igyekeznek célt érni. A védekezések a támadások ellen a sebezhetőségek számának mielőbbi csökkentését jelentik. A fenyegetettségek és a védekezések mértékétől függ az adott rendszer kitétsége. A kockázatelemzés a fenyegetettségek bekövetkezési valószínűségét és a várható kárt méri fel, és ehhez rendeli a védekezési szintek vagy megoldások költségét. Az elemző munka alapján lehet egyes kockázatokat kezelni (védekezési eljárással) vagy felvállalni, de a végeredmény tudatos lesz. Az elemzéseket időszakosan újra el kell végezni egyrészt a védekezés hatékonyságának felmérésére, másrészt az esetleges új fenyegetettségek felfedezésére.

1. ábra Az informatikai biztonság elemei és összefüggései [1]

1.4 A linux

A Linux egy Unix szerű operációs rendszer, amely a linux kernelmagot használja. A szabad szoftverek és a nyílt forráskódú programok egyik legismertebb példája. A Linux-rendszermagot Linus Torvalds 1991-ben kezdte el fejleszteni. A hétköznapi nyelvben gyakran a teljes operációs rendszerre utalnak vele, amely a

Linux-rendszermagra és az 1983-ban, Richard Matthew Stallman vezetésével indult GNU projekt keretében született programokra épül. A Linux pontosabb neve ebben az értelemben **GNU/Linux**.

1.5 A GNU/linux részei

Kernel, feltelepített szoftverek, és azok beállításai, a felhasználói interfész (gyakran shell), a fájlrendszer, a függvénykönyvtárak és a fordító. A GNU/linux hardening keretein belül általában az első öt rész biztonsági kérdéseivel foglalkozunk.

2. Mennyire biztonságos a linux?

Ha az operációs rendszereket össze kell hasonlítani, sokszor halljuk, hogy a Linux mennyire biztonságos. Amikor a Linux biztonságosságáról beszélgetünk, folyton a biztonsági problémák, megoldások sokaságáról esik szó. Akkor most a Linux biztonságos vagy nem? A válasz attól függ, milyen Linux disztribúciót és abban alkalmazható megoldásokat használunk, illetve mennyire szeretnénk biztonságosnak a rendszerünket.

2.1. Főbb biztonsági kockázatok

A Linux biztonsági hibáit jellegük szerint alapvetően két csoportba sorolhatjuk. Az első csoportba a helyi biztonság kérdések, a másodikba a távoli (hálózati) biztonsági hozzáférések kérdései tartoznak. A helyi biztonság kérdése, azaz a szerverünkre bejelentkezett felhasználók, vagy a felhasználók nevében futó alkalmazások kordában tartása is annyira fontos, mint a hálózati biztonságé. A szerverünkre történő betörés nagy része a helyi folyamatokon keresztül történik, mert a biztonsági hibák a legtöbb esetben egy program nem várt – általában számunkra helytelen működése – vagy általunk elvégzett rossz konfigurációk okozzák.

2.2. Védekezési lehetőségek

Az első lépés a kockázati tényezők felmérése, és csak a második a kockázatok csökkentése. A mindenféle hibából eredő kockázatokat jelentősen csökkenthetjük, ha a felesleges jogokat megvonjuk a programoktól, valamint a működésüket kontrolláljuk. Erre alkalmas megoldások többek között a PaX [2], a GrSecurity [3], az RSBAC [4], a LIDS [5], a SELinux [6], stb. Sajnos ezeket nem lehet egyszerre feltelepíteni rendszerünkre, mert ezek közül számos megoldás kizárja vagy értelmetlenné teszi a másikat, illetve tartalmazza annak funkcióját. Egy rendszeren természetesen ezeken kívül alkalmazha-

tunk, sőt kell is alkalmaznunk olyan technikákat, amelyek megnehezíthetik rendszerünk kiismerését, még akkor is, ha hatásuk átfedik egymást. Az adatok lopásvédelme már egy érdekesebb kérdés.

3. Hozzáférés-vezérlési modellek Linuxon

Régen úgy tűnt hogy kiváló módszer a hozzáférés-vezérlésre az hogy az erőforrásokat jelképező objektumokat (file), ellátják olyan attribútumokkal, amikből kiderül hogy ki férhet ahhoz hozzá, és milyen műveletet végezhet az egyes fájlokon. Természetesen valakinek képesnek kell lennie arra a műveletre hogy ezeket a jogosultságokat beállítsa. Ilyenkor az erőforrás tulajdonosára van bízva, hogy ki nek milyen jogosultsága lesz az erőforráshoz. Az ilyen módszereket nevezzük diszkrecionális hozzáférési modelleknek. Azok a modelleket, amelyek azon alapulnak, hogy az adatoknak és a felhasználóknak címkéi vannak, amik megmondják hogy az adat mennyire titkos, illetve milyen jellegű, és a felhasználó mennyire titkos és milyen jellegű adatokhoz férhet hozzá kötelező hozzáférés-vezérlésű (MAC, Mandatory Access Control) modelleknek hívjuk. Ezek a modellek olyan szabályokat definiálnak, amiket az informatikai rendszer minden elemének és felhasználójának be kell tartania.

3.1 Diszkrecionális hozzáférési modellek

A diszkrecionális hozzáférési modellekben általában a felhasználók és a felhasználói csoportok alapján osztanak ki jogokat.

3.1.1 Szabványos unix hozzáférés-vezérlés

A Linux/Unix rendszereken hagyományosan a felhasználó-csoport-bárki más alapú hozzáférés-vezérlési modellt implementálják. Ez a modell egy bonyolultabb igényeket támaztó rendszeren gyakran nem elégséges. Egy állományhoz egy felhasználó tartozik, aki egyúttal annak tulajdonosa is, és egy csoport, valamint mindenki mást. A műveletek is kevesen vannak, név szerint írás, olvasás, végrehajtás.

3.1.2 Hozzáférési listák

Mint a legtöbb komoly Unix változaton, Linuxon is lehetséges ennél komolyabb hozzáférés-vezérlést megvalósítani. Másik nagyon elterjedt diszkrecionális hozzáférés-vezérlési módszer a hozzáférési listák, ACL-ek (Access Control List) használata[7][8]. Az ACL-eket is a felhasználó és felhasználói csoport alapján osztjuk ki, de egy állományhoz több szabály is tartozhat, és nem csak olyan szabályt osztunk ki ami megad egy hozzáférést, hanem olyan is ami azt megvonja. Az ACL típusú rendszereken öröklődő szabályokat is létrehozhatunk, ekkor egy könyvtár hozzáférési jogai öröklődnek az abban lévő file-okra is. Az írás, olvasás, végrehajtás műveleteken felül további műveletek is definiáltak.

3.1.3 Hálózati hozzáférés-vezérlés

Amikor hálózati hozzáférés-vezérlésről beszélünk, a felhasználót nem csak felhasználónév vagy csoport, hanem az általa használt gép IP címe alapján is azonosíthatjuk. Léteznek olyan hálózati alkalmazások, melyek saját hozzáférés-vezérléssel rendelkeznek, illetve használják a fájlrendszer hozzáférés-vezérlését is. A szabványosnak tekinthető hálózati hozzáférés-vezérlési módszer unixokon a tcpwrapper [9] használata, amely a hosts.allow és hosts.deny állomány alapján dönti el hogy egy adott

IP cím a szolgáltatáshoz hozzáférhet-e. Az alacsonyabb szinten végzett hozzáférés-vezérlés a csomag-szűrés. A csomagszűrés mint hozzáférés-vezérlési módszer szegényes eszköztárral rendelkezik, mivel alacsony szinten dolgozik, nagyon kevés információja van a hálózati szolgáltatásokról, és nagyon kevés műveletet ismer.

3.1.4 A diszkrecionális hozzáférés-vezérlési módszerek hátrányai

A diszkrecionális hozzáférés-vezérlési módszerek a trójaiak, a rosszindulatú felhasználók és a rosszul megírt applikációk ellen nem véd. Vegyük például azt az esetet, amikor András elkészít egy titkos dokumentumot. A dokumentum annyira titkos, hogy csak Béla láthatja, Cecília, a titkárnő már nem. András beállítja a dokumentum hozzáférési jogait úgy hogy csak Béla láthassa. De Béla nyúl, és a dokumentumról készít egy másolatot Cecília részére, hogy helyette ő végezze el a feladatot. Igen ám, de Cecília a versenytárs cég felbérelt ügynöke, és elküldi a dokumentumot az konkurenciának. Ilyen és hasonló esetekre kivédésére találták ki a kötelező hozzáférés-vezérlési modelleket.

3.2 Kötelező hozzáférés-vezérlési modellek

A kötelező hozzáférés-vezérlés (MAC, Mandatory Access Control) olyan szabályokat definiálnak, amiket az informatikai rendszer minden elemének és felhasználójának be kell tartania. Ezek a modellek azon alapulnak, hogy az adatokra és a felhasználókra címkéket ragasztunk, amik megmondják hogy az adat mennyire titkos, illetve milyen jellegű, és a felhasználó mennyire titkos és milyen jellegű adatokhoz férhet hozzá. Kötelező hozzáférés-vezérlési modell többféle létezik, attól függően hogy az adatoknak melyik tulajdonságát (bizalmasság, vagy integritás) kell jobban védeni, illetve hogy milyen lehetőségeket nyújt a használt informatikai rendszer.

4. Összefoglaló

A dolgozatban igyekeztünk kitérni a fontosabb alapfogalmak bevezetésére és az alapvető modellek főbb jellemzőinek rövid ismertetésére. A tömörségből látható, hogy a téma sokrétűbb, mint ahogy elsőre gondolnánk; és összetettebb, mint ahogy kinéz.

Irodalom

- [1] Kincses Zoltán: *Biztonsági kérdések a korszerű hálózati adatátvitelben és üzemeltetésben*; Doktori értekezés Budapest, 2005
- [2] <http://pax.grsecurity.net>
- [3] <http://www.grsecurity.net>
- [4] Rule Set Based Access Control (RSBAC) for Linux (<http://www.rsbac.org>)
- [5] LIDS: Linux Intrusion Detection System: <http://www.lids.org>
- [6] NSA Security-enhanced Linux <http://www.nsa.gov/research/selinux>
- [7] Trusted unix working group (trusix) rationale for selecting access control list features for the unix (r) system (<http://nucua.unomaha.edu/dspace/documents/NCSC-TG-020-A.pdf>)
- [8] Extended attributes and ACL for Linux (<http://acl.bestbits.at>)
- [9] W.Z. Venema, „TCP WRAPPER, network monitoring, access control and booby traps”, Mathematics and Computing Science, 2000

Fried Zoltán

Munkahely: Unicredit Leasing Kft.
Cím: 1016, Magyarország, Budapest, Hegyalja út 7-13.
Telefon: 06-20-4155548
E-mail: kolozsvar@deirf.hu