Biztonsági megoldások tűzfalak segítségével

Csubák Dániel[1], Kiss Attila[2]

[1] csuby@caesar.elte.hu
ELTE IK, Információs Rendszerek Tanszék,
Balabit IT Security Kft.
[2] kiss@inf.elte.hu
ELTE IK, Információs Rendszerek Tanszék

Absztrakt. A tűzfal célja a számítástechnikában annak biztosítása, hogy a hálózaton keresztül egy adott számítógépbe ne történhessen illetéktelen behatolás. Először áttekintjük az olyan alapvető fogalmakat, mint például a tűzfalak típusai és fajtái, átjárók, zónák, szolgáltatások, stb. A továbbiakban a tűzfalakat célzó, károsító támadások néhány fajtája, menete kerül majd ismertetésre. Természetesen az üzemeltetés szempontjából lényeges, hogy a hiba kockázatát minimálisra csökkentsék, így a támadások tárgyalása után bemutatásra kerülnek olyan lehetőségek és eszközök, melyekkel a felhasználó a betörésekről időben értesülhet, azokat elkerülheti, illetve a tűzfal működését ellenőrizheti, optimalizálhatja.

1. Bevezetés

Az információs társadalom életének rengeteg területén egyre több számítógép, illetve internethez való kapcsolódási képességgel rendelkező eszköz jelent meg, mely a számos előny mellett különféle veszélyforrásokat is magában hordoz. Ahogyan a támadható eszközök száma növekszik, úgy növekszik az igény a védelmet nyújtó alkalmazások, így a tűzfalak iránt is.

Mindemellett, ahogyan a vállalatok hálózatainak mérete is egyre növekszik, illetve egyre több szolgáltatást (pl.: web, e-mail) nyújtanak, melyeknek kívülről – azaz nem a vállalat belső hálózatából – is elérhetőnek kell lennie, a biztonsági kockázatok számukra is megnövekednek. A kritikus vállalati hálózatok védelmére határvédelmi eszközökként tűzfalakat használnak, melyek kontrollálják, felügyelik és esetlegesen szűrik, vagy monitorozzák a vállalat rendszerébe belépő, és abból kimenő adatforgalmat.

A továbbiakban a 2. fejezet megismerteti az olvasót a témában szükséges alapvető fogalmakkal, illetve a tűzfalak típusaival, fajtáival egy egyszerű példahálózat felépítésének bemutatásán keresztül. A 3. fejezet a határvédelmi pontokat veszélyeztető támadásokra irányítja a figyelmet, kitérve ismertebb módszerekre, míg a 4. fejezet az üzemeltető, a rendszergazda eszköztárának néhány elemét mutatja be, mellyel a betörésekről értesülhet, elkerülheti azok egy részét, illetve optimálisabbá teheti a tűzfal működését, illetve kurrens szakcikkek által prezentált eszközöket ismertet, végül az 5. fejezet röviden összefoglalja a korábbiakat.

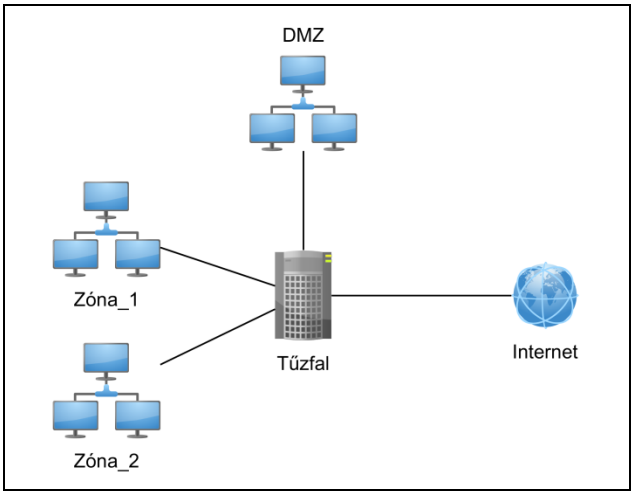
2. Alapvető fogalmak

Az által, hogy a vállalatok szolgáltatásaik egy részét a külvilág számára is elérhetővé kívánták tenni, saját hálózati felépítésükben egy új elem jelent meg, mely biztonsági kockázatot tartalmazott, hiszen a védett hálózat egy részéhez széleskörű hozzáférést nyújtottak. Egy ilyen változás szemléletváltást hozott magával a biztonsági kockázat kiküszöbölésének, az esetleges veszteségek csökkentésének érdekében.

Szükségessé vált, hogy a vállalatok elkülönítsék hálózatuk azon részét, melyet a külvilág számára is hozzáférhetővé kívántak tenni, illetve azon alhálózataikat, melyeket kizárólag belső használatra terveztek, kritikus információkat tartalmaztak. Mindemellett, olyan céllal, hogy a vállalatban megjelenő felelősségi körök, és hozzáférési jogosultságok elkülönülése, illetve az egyes alszervezetek logikai egységei is elkülönüljenek, további csoportosításokat vezettek be.

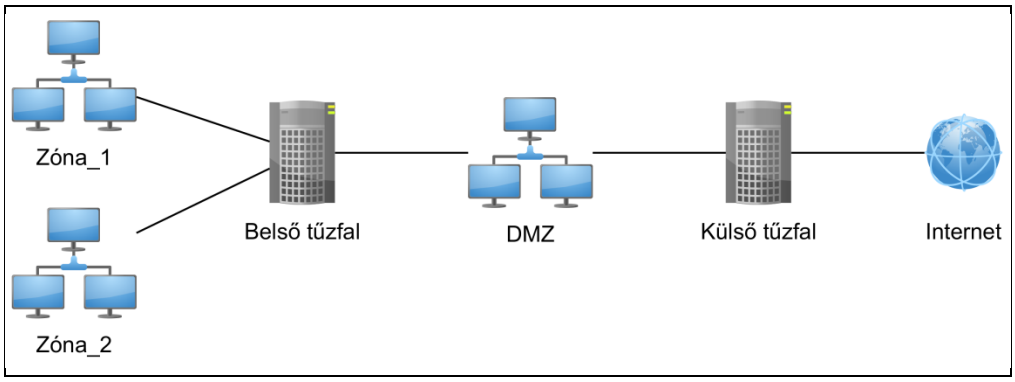
Zónáknak hívjuk egy hálózat egy jól elkülöníthető, a többi alhálózattól teljesen diszjunkt alhálózatát. Egy vállalat zónaszerkezetét, zónáinak hierarchiáját általában valamilyen fizikai, logikai, szervezeti csoportosítás alapján készítik el. Ezen struktúra lényege, hogy a felelősségi körök mentén szeparálja a szervezet résztvevőit, és a tűzfal üzemeltetője számára segítséget nyújtson a logikai csoportok kialakításához, melyek alapján később egyik zónából egy másikba engedélyezi, vagy tiltja a hozzáférést. Például: egy cég fejlesztési osztályán dolgozó munkatárs számítógépéről nyilvánvalóan nem indokolt, sőt, kockázatot hordoz az esetleges hozzáférés a könyvelésen dolgozó munkatársak számítógépeihez.

A fentebb említett, kívülről elérhető alhálózat neve Demilitarizált Zóna (továbbiakban: DMZ), mely kialakításának célja, hogy a vállalat belső hálózatáról szigorúan leválasztva tartsa azokat a szolgáltatásokat nyújtó számítógépeket, melyek kívülről is elérhetőek, ez által pedig könnyebben támadhatóak.



1. ábra: Egyszerű hálózati felépítés, melyben a DMZ ugyanahhoz a tűzfalhoz kapcsolódik, mint a többi zóna, csak másik hálózati interfészen.

A DMZ-t általában ugyanúgy, ahogyan a céges hálózat többi részét, tűzfallal védjük, azonban az valamilyen szinten elkülönített a belső hálózatot védő eszköztől. A 1. ábrán látható egy vállalati hálózat, illetve a DMZ, melyeket ugyanaz a tűzfal véd, különböző hálózati interfészeket használva az egyes zónákhoz. Ezen megoldásban bár megjelenik az elkülönítettség, mégis idővel elavulttá vált, hiszen a belső hálózatot védő tűzfalgéphez a támadó a DMZ-n keresztül hozzá tudott férni.



2. ábra: Egyszerű hálózati felépítés, melyben a DMZ-t csak a külső-, míg a belső hálózatot a belső tűzfal is védi.

Napjainkban elterjedtebb a 2. ábrán láthatóhoz hasonló hálózati kialakítás. Ebben az esetben a DMZ-t egy külső tűzfal védi a „külvilágtól", azonban mivel annak szolgáltatásai kívülre is elérhetőek, egy második, belső tűzfal kerül elhelyezésre a hálózat kritikus zónái – például: fejlesztés, pénzügy, könyvelés – és a DMZ közé, mely egy újabb biztonsági réteget vezet be. A két tűzfal konfigurációja, de előfordulhat, hogy a rajtuk futó alkalmazás is különbözik, hogy ez által a támadó, aki átjutott a külső tűzfalon, ugyanazzal a technikával csak kevesebb eséllyel legyen képes a belső hálózathoz is hozzáférni, míg a belső hálózatból csak a tűzfalon keresztül lehet elérni a DMZ által nyújtott szolgáltatások bármelyikét.

2.1. A tűzfalak fajtái, típusai

A tűzfalakat a vállalat hálózatában ellátott feladatuk alapján, illetve funkcióik szerint is tudjuk csoportosítani.

Az első kategóriában három csoport alakítható ki, melyek közül kettő már a 2. ábra kapcsán említésre került. Ezek:

- külső tűzfal: a belső hálózatot csak részben választja el az internettől. Jellemzően a DMZ és a külvilág közé helyezzük el.
- belső tűzfal: a belső hálózat védelmi szempontból kritikus részét választja el a többitől. Jellemzően a DMZ és a helyi hálózat kritikus részei közé helyezzük el.
- személyi tűzfal: egy adott hosztra elhelyezett szolgáltatás, melyre jelen tanulmányban nem térünk ki.

A funkcionalitás szerinti csoportosítás sokkal többrétű, mely által a tűzfalak fejlődésének lépéseit is végigkövethetjük. Ezek a továbbiakban kerülnek részletezésre.

2.1.1. Csomagszűrő tűzfal

Az egyik legrégebbi típus, mely manapság a tűzfalak alapfunkciója. Lényege, hogy az adatcsomagokat a cél- és forrás IP-cím és port alapján szűrhetjük, azaz jellemzően az OSI modell [5] alsóbb rétegeivel foglalkozik. A szűrés feltételeit egy szabályrendszer (rule set) definiálja, amelyben meghatározzuk, hogy az egy-egy feltételhez illeszkedő adatcsomagokkal mi történjen. Jellemzően két opció létezik – vagy átengedjük a csomagot, vagy blokkoljuk, eldobjuk, mely esetben nem történik visszajelzés, csupán a kapcsolat nem jön létre.

Minden csomag esetén a tűzfal egyetlen szabályt választ, ami meghatározza a döntést, azonban erre több módszer is létezik. Az úgynevezett first match (első találat) az első illeszkedő

szabályt, míg a best match (legjobb találat) valamilyen heurisztika alapján a legjobbnak ítélt szabályt választja majd. Az alább látható szöveges leírás first match alapján a 192.168.1.1-re érkező HTTP forgalmat blokkolni fogja (mivel ez a 192.168.1.0/24 alhálózat része, és azon szabály az első), míg best match alapján a tűzfal várhatóan felismeri, hogy a második szabály specifikusabb, szűkebb alhálózatra vonatkozik, így engedélyezi a forgalmat. First match-et használó tűzfalak esetén kiemelt fontosságú, hogy a példában látotthoz hasonló „elfedés" ne történjen, mert ez által a működés el fog térni az elképzelttől.

```
http forgalom tiltása a 192.168.1.0/24 alhálózatba
http forgalom engedélyezése a 192.168.1.1/32 alhálózatba
```

Ilyen típusú tűzfal például a Linux operációs rendszerben található iptables [6].

2.1.2. Állapotmegőrző tűzfal

A csomagszűrő némiképpen „továbbfejlesztett" változata, ami hálózati kapcsolatok állapotát tartja nyilván, és ezek egyes jellemzőt képes elemezni. Ezen vizsgálat a be- és kiküldött adatcsomagokat tekinti a fentebbi jellemzők tükrében és dinamikus táblákban – úgynevezett állapottáblákban – tárolja azokat, majd az összegyűjtött adatok alapján elemzi a csomagokat, így nem szükséges, hogy a tűzfal üzemeltetője bonyolult szabályrendszereket tartson karban – a tűzfal az engedélyezett kapcsolathoz tartozó adatcsomagokat továbbengedi, míg a többit eldobja. Azaz az állapottábla segít felismerni a kapcsolatok közötti összefüggéseket, így a kapcsolat kiépülése után a tűzfal ennek a segítségével ismeri fel, hogy az adatforgalom valós igényeket elégít-e ki, és amennyiben a kliens a külső rendszertől olyan csomagokat kap, melyeket nem kért, a tűzfal blokkolja azokat. Természetesen szabályrendszert ebben az esetben is kell építeni, azonban az valamivel egyszerűbb.

Ilyen típusú tűzfal volt például Check PointFireWall-1, azonban az ilyen működés mára alapfunkciónak tekinthető.

2.1.3. Alkalmazás-szintű tűzfal

Az ilyen típusú tűzfalak az OSI modell magasabb rétegeivel is foglalkoznak, a forgalomhoz tartozó jellemzők mellett a szolgáltatások adatait és a hálózati csomagok tartalmát is figyelik, kontrollálják a be-, illetve kimenetet és a hozzáférést az egyes alkalmazásokhoz, szolgáltatásokhoz.

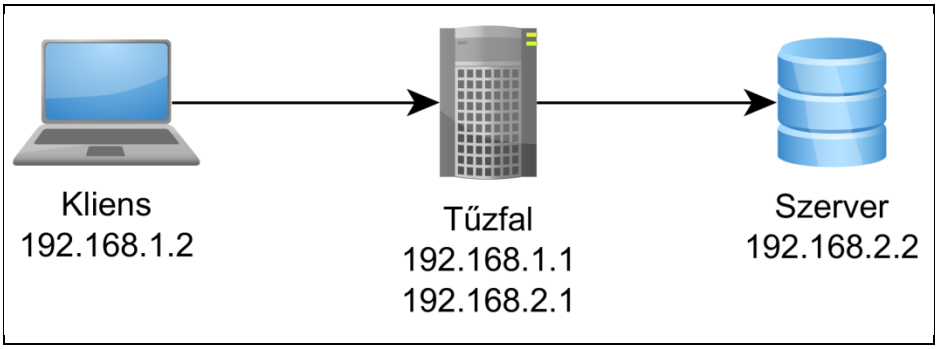
Speciális típusa a web alkalmazás tűzfal (Web Application Firewall, a továbbiakban WAF), mely egy olyan berendezés, szerver modul, vagy szűrő, amely szabályok egy halmazát alkalmazza HTTP kapcsolatokra, ez által az általános támadások, sebezhetőségek (például SQL injection, vagy cross site scripting) kínál védelmet. Természetesen ugyanúgy, ahogyan más esetekben, a szabályrendszer helyes beállítása itt is kritikus fontosságú.

Applikációs-szintű tűzfal például a magyar fejlesztésű Zorp GPL [7].

2.1.4. Proxy tűzfal

Ezen típusú tűzfalak a kliens és a szerver közötti kapcsolat kialakításáért felelősek, azaz amennyiben egy kapcsolódás engedélyezett a tűzfal kapcsolódik a szerverhez, így a kliens nem direkt módon, hanem ezen keresztül tud kommunikálni a szerverrel. Ebben az esetben megoldható, hogy a szerver ne láthassa a kliens IP címét, így kevesebb információval fog rendelkezni a belső hálózat felépítéséről. A csomagok helyes továbbításáról a tűzfalnak kell gondoskodnia. Lényegében a proxy tűzfal „közvetítőként" funkcionál a kapcsolat két végpontja között: feldolgozza a kliens kéréseit és azokkal egyenértékű kéréseket küld a szerver felé, a válaszokkal pedig hasonló módon jár el.

Legegyszerűbb esete csupán annyit foglal magába, hogy a tűzfal egy köztes pont minden kifelé menő kapcsolatban, így elfedvén a belső hálózatot, azonban a proxy-k igazi előnye abban rejlik, hogy a megfelelő protokollok (például HTTP, FTP, SMTP) ismeretével rendelkező dedikált proxy például az adatfolyam csomagjaiban módosításokat képes végrehajtani (például a titkos vállalati adatokat a kliens kimenő kéréseiből egy hash-értékkel helyettesíti, hogy az ne kerülhessen ki a „külvilágba"), mintákat kereshet azokban, amik kártevőkre is utalhatnak, elemzésre adatokat gyűjthet belőle, stb. A tűzfalon természetesen több dedikált proxy is futhat párhuzamosan, mivel ezek lényegében különálló programok, melyek a beérkező csomagokkal foglalkoznak. Mindemellett a proxy tűzfalak általában használnak valamilyen tartalmi gyorsítótárat (cache-t), így némely esetetben csökkentheti a kifelé menő forgalmat.



3. ábra: Proxy esetén a szerverhez megérkező csomagban a küldő címe 192.168.1.2 (a kliens IP címe), míg anonymus proxy esetén 192.168.2.1 (a tűzfal szerver oldali hálózati interfészének címe) lesz.

Két típusa létezik, melyek közötti különbséget a 3. ábra szemlélteti. Az első típus, melyre általában proxy-ként hivatkozunk nem fedi el a kliens IP címét a szerver elől, míg az anonymus proxy ezt a fentebb leírt módon megteszi.

A modern alkalmazás-szintű tűzfalak (így az ott példaként felhozott is) általában rendelkezik integrált proxy megoldásokkal, vagy lehetőséget adnak arra, hogy külső proxy-kat integrálhassunk, így használva azok funkcionalitását.

2.1.5. Tartalomszűrő tűzfal

A tűzfal tartalomszűrők segítségével elemezheti, kiértékelheti egy kapcsolat adatait. Jellemzően vagy eleve integrált megoldásokat kínálnak az egyes alkalmazások, vagy lehetőséget biztosítsanak arra, hogy külső programot használjunk fel ilyen célra.

A tartalomszűrés tipikus esetei:

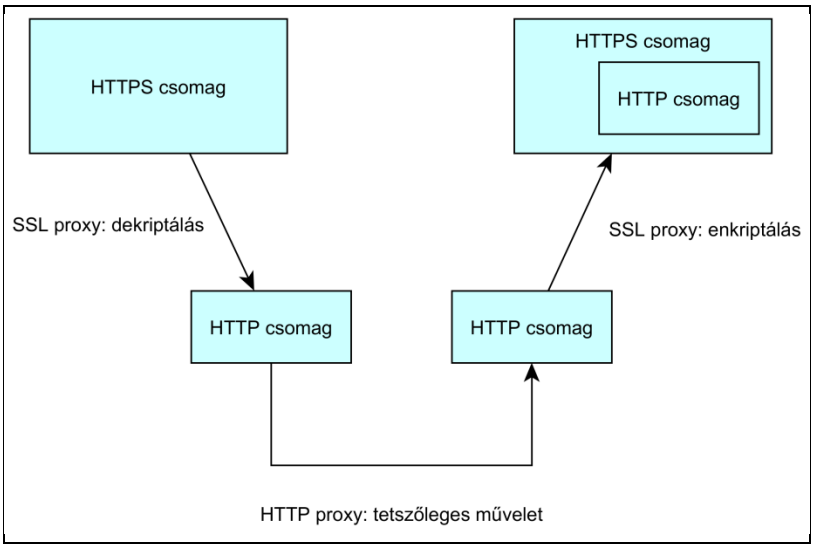
- URL-szűrés (URL-filtering)
- vírusszűrés
- HTTP oldalak szövegének figyelembe vételével egyes oldalak blokkolása
- kulcsszavak alapján egyes, nem kívánt weboldalak tiltása
- bizalmas, kritikus céginformációk kiszűrése
- ActiveX és JavaScript kiszűrése

2.2. Hibrid megoldások, modularitás

Fentebb a teljesség igénye nélkül felsorolt funkciók közül a modern tűzfalak többet is magukba foglalnak, így hibrid megoldásoknak tekinthetőek. Könnyen elképzelhető például alkalmazás-

szintű tűzfal, mely integrált proxy-kkal dolgozik, melyek valamilyen tartalomszűrő funkciót is ellátnak.

A modularitás az egyes funkciók elkülönítését, egymásba ágyazhatóságát jelenti. Ez alatt annyit kell érteni, hogy például a HTTPS protokoll elemzéséhez nem elegendő egy proxy, hanem – mivel maga a protokoll is SSL használatával titkosított HTTP forgalmat jelent – két, egymásba ágyazott proxy alkalmazásával lehet mindezt megoldani. Ahogyan az a 4. ábrán, szemléletesen látszik, az SSL-t értelmező proxy felelős a dekriptálásért, míg a HTTP proxy képes az ez után előállt HTTP adatokat elemezni. Az elemzés után az egymásba ágyazott proxy-k előállítják a továbbítandó csomagot, mely ilyen formában fogja majd elhagyni a tűzfalat.



4. ábra: Egymásba ágyazott proxy-k esetén a működést szemléltető példa.

3. Tűzfal elleni támadások

A tűzfalakat célzó támadások célja általában a védelmi rendszeren történő átjutás, hogy ez által a lezárt belső hálózatban tárolt kritikus információk a támadó eltulajdoníthassa, a hálózatban lévő számítógépeken módosítást hajtson végre, vagy azokat további céljaira felhasználhassa.

A támadásnak általában több fázisa van, ezek jellemzően a felderítés és a védelmi rendszeren történő áthatolás.

3.1. Felderítés

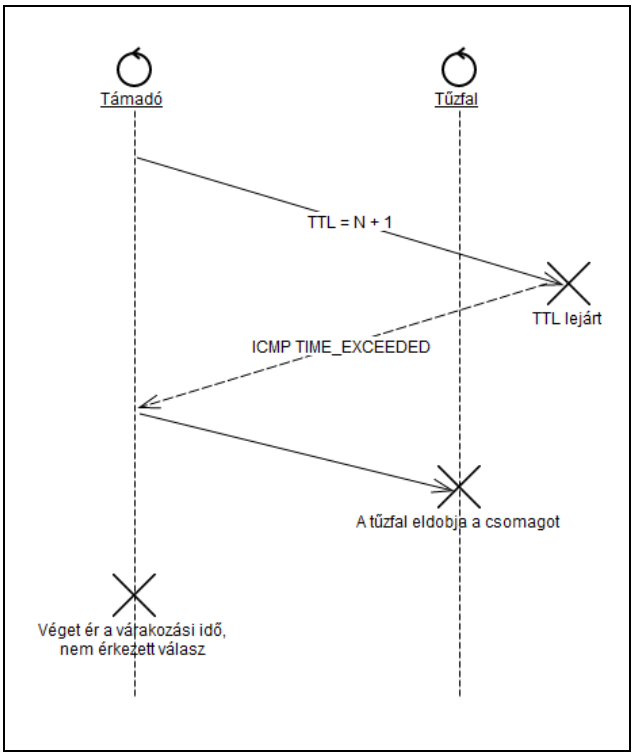
A támadó célja a felderítéssel többrétű dolog. Egyrészt elképzelhető, hogy a belső hálózat topológiai felépítését szeretné megismerni, másrészt lehetséges, hogy a tűzfal szabályrendszerét, annak működését szeretné kideríteni. Utóbbi az, ami a tanulmány szempontjából lényegesebb.

Különböző portszkennelési technikák alkalmazásával a támadó képes felderíteni, hogy az adott hálózati eszközön milyen alkalmazások, szolgáltatások futnak, azonosíthatja a zárt portokat. Tűzfalak esetén egy ilyen vizsgálattal a támadó a nyitott portokat tudja meghatározni.

Portszkennelés egyszerűen végezhető az nmap [8] program segítségével.

A tűzfalszabályok felderítésének célja, hogy a támadó megtudja, hogy milyen forgalom megy át a tűzfalon, illetve hogy mik a szűrt portok. Ezen célra az egyik leghatékonyabb módszer

a [9]-ban bemutatott és [10]-ben összefoglalt FireWalk technika. Ennek lényege – melyet a 5. ábra szemléltet –, hogy a tűzfal távolságának (hány ugrásra, hop-ra van a támadótól) meghatározása után – legyen ez a távolság N – a csomag Time-to-Live (továbbiakban TTL) mezőjét N+1-re állítva hajt végre szkennelést a támadó egy adott porton, majd a tűzfal által adott válaszból következtetést von le az érvényben lévő szabályra. Amennyiben a tűzfal átengedte az adott csomagot, a támadó a TTL beállítás miatt egy ICMP TIME-EXCEEDED [12] üzenetet fog visszakapni, míg ellenkező esetben nem érkezik majd válasz. Ezen módszer a FireWalk technikák elvi alapja, melynek [9]-ban bemutatott módosításait, változatait alkalmazzák leginkább a gyakorlatban.



5. ábra: FireWalk szemléltetése: az első küldés esetén van tűzfalszabály, ami elfogadja a csomagot, így amikor a TTL elfogy, a megfelelő ICMP üzenet érkezik válaszul. Ellenkező esetben a tűzfal válasz nélkül eldobja a csomagot.

3.2. Áthatolás a védelmi rendszeren

A tűzfal szabályainak felderítése után a támadónak több lehetősége van a folytatásra. [10]-ben ezen technikák részletesen kifejtésre kerültek, így ebben tanulmányban csak felsoroljuk őket.

- IP tördelés
- forrás routing
- forrás-port hamisítás
- TCP, IP nem használt mezők, illetve illegális értékek
- támadás a tűzfal operációs rendszere ellen

4. Az üzemeltetőt segítő eszközök, lehetőségek

A tűzfal üzemeltetetését végző szakembernek, rendszergazdának eszközök, módszerek sokasága áll rendelkezésére mind abban, hogy a tűzfal egyes támadásokkal szemben védett, vagy kevésbé sebezhető legyen, mind azon tekintetben, hogy valamilyen szempontból optimálisabban működjön.

Jelen tanulmányban a teljesség igénye nélkül a monitoring, a reporting (jelentéskészítés) jelentősségét, lehetőségét mutatjuk be, illetve a fentebb bemutatott alapvető FireWalk technika elleni védekezési módot ismertetünk. Mindezek mellett kapcsolódó szakcikkekben ismertetett módszereket, illetve lehetőségeket prezentálunk.

Fontos hangsúlyozni, hogy pontosan úgy, ahogyan a támadások elleni védekezés fejlődik, a támadók, a támadási módszerek is fejlődnek, változnak. Az IT biztonság területén nem lehet mindenre felkészülni, nincsen tökéletes védettség (hiszen ha IT szempontból még teljesen védett is lenne a hálózatunk, az emberi tényezőt akkor sem tudjuk teljességgel kizárni), azonban azt különböző módszerekkel, folyamatos felülvizsgálatokkal, szoftvereink biztonsági frissítéseinek telepítésével közelíteni tudjuk.

4.1. Monitoring

A monitorozás által a tűzfal jelenlegi állapotáról kaphatunk információkat, illetve, amennyiben a monitorozást végző alkalmazás valahol eltárolja adatainkat, különféle, a múltban történt eseményekkel kapcsolatban is információkat szolgáltathat számunkra.

A monitorozást végző alkalmazás lehet a tűzfalba integrálva, vagy attól különálló programként, mely különféle szenzorokkal kapcsolódik tűzfalunkhoz, hogy adatokat gyűjtsön róla.

A monitorozás során például az alábbi információkat kaphatjuk meg:

- be-, és kimenő kapcsolatok száma
- be-, és kimenő forgalom nagysága
- amennyiben a tűzfal több összekapcsolt számítógépből áll, ezek állapota
- a rendszerünk terheltsége
- elindult proxy példányok száma

Mindemellett a monitorozást biztosító alkalmazások általában képesek a tűzfalról kinyert adatokat grafikonokon szemléltetni, melyek dinamikusan változnak, így ettől már csak egy lépés az, hogy amennyiben egy érték meghalad egy előre definiált küszöbértéket (például kiugró forgalom keletkezik), a monitorozásért felelős program figyelmeztessen minket erre.

4.2. Reporting

A jelentéskészítés a tűzfalon, illetve a hálózatunkban egy adott időintervallumban (általában napi-, heti-, illetve havi reportokra van szükségünk) történt eseményeket ismerteti, szemlélteti. Célja kettős, mivel az üzemeltetésnek nyújtott segítség mellett a menedzsment számára is hasznos információkat tartalmaz.

A jelentések készítésének első lépése általában a tűzfal által készített naplóbejegyzések (logok) megfelelő módon történő összegyűjtése. A tűzfalak a rajtuk keresztül menő forgalom függvényében képesek rengeteg, naponta akár több gigabájtnyi log-ot is előállítani, így ezek összegyűjtése közben, vagy közvetlenül az után fontos ezek előfeldolgozása, hogy a jelentéskészítő alkalmazás már csak az ő szempontjából ténylegesen hasznos adatokkal kelljen hogy foglalkozzon.

A jelentések például az alábbi adatokat tartalmazhatják:

- egyes alhálózatok, zónák adatforgalma
- egyes hosztok adatforgalma
- népszerű célcímek
- levelezési statisztikák
- szabálysértések száma

A jelentések alapján a tűzfal üzemeltetője következtetéseket vonhat le például azt illetően, hogy mely zónák forgalma nagyobb, így azokon megfelelően el tudja osztani a terhelést, vagy láthatja, hogy mely felhasználók követik el a legtöbb szabálysértést.

A jelentések esetén esszenciális fontosságú, hogy azok formailag megfelelőek és könnyen áttekinthetőek legyenek, hiszen lényegük pont abban rejlik, hogy az üzemeltető számára megkönnyítik a naplóbejegyzésekből történő információkinyerést.

4.3. Védekezés a példaként mutatott FireWalk módszer ellen

A 3.1. alfejezetben a tűzfalszabályok felderítésére bemutatott FireWalk technika alapja egy ICMP TIME_EXCEEDED üzenet beérkezése a támadóhoz abban az esetben, ha az általa küldött csomagot a tűzfal átengedte. Ez alapján világos, hogy ezen módszer ellen úgy tudunk védekezni, ha a tűzfalon blokkoljuk az ilyen típusú ICMP csomagok kilépését a védett hálózatból, hiszen ebben az esetben a támadó semmikor nem kapja meg a választ, ami számára megkülönbözteti a blokkolt, illetve az átengedett csomagot.

Fontos megjegyezni, hogy a FireWalk technika rendelkezik továbbfejlesztett változatokkal, ezen módszer csak az alapértelmezett verzió ellen effektív.

4.4. Kapcsolódó munkák módszerei

Ebben az alfejezetben két módszert mutatunk be röviden. Az egyik a különböző szoftversebezhetőségekkel szemben lehet az üzemeltető segítségére, míg a másik a tűzfal szabályrendszerének offline módon történő kiértékelésében, elemzésében használható.

A [13]-ben bemutatott MulVAL egy Datalog alapú sebezhetőség elemző (vulnerability analyzer). Segítségével megtudhatjuk, hogy milyen szoftversebezhetőségek találhatóak meg a rendszerünkben, és ezek milyen esetleges következményekkel járhatnak egy támadás esetén.

Az eszköz először egy CVE adatbázisból - mely publikusan bejelentett és dokumentált szoftversebezhetőségeket tartalmaz - frissíti saját adatbázisát, majd vizsgálatokat végez az általunk megadott hosztokon. Ezen vizsgálatok eredményeképpen kiderül, hogy milyen sebezhetőségek vannak jelen a rendszerünk egyes számítógépein. Következő lépésként az eszköz manuális adatbevitelt is igénybe véve elkészíti a hálózatunk modelljét, illetve a tűzfalszabályaink alapján az engedélyezett, és tiltott kapcsolatok listáját, míg végül az összegyűjtött adatokat összevetve kikövetkezteti, hogy milyen hozzáférés kiterjesztési, illetve szolgáltatáskieséssel járó támadás az, ami tisztán a szoftverhibákat kihasználva érinti azt.

A MulVAL koncepciója alapján is láthatjuk, hogy szoftvereink biztonsági frissítéseit érdemes minél korábban telepítenünk, ha nem szeretnénk különféle, ezek hiányából fakadó veszélynek kitenni belső hálózatunkat. Az elmúlt fél évben több olyan szoftversebezhetőség is napvilágra került, mely komoly problémákat okozhatott azon rendszerekben, melyek a biztonsági frissítéseket csak késve telepítették, vegyük csak példának a HeartBleed-et, vagy a ShellShock-ot melyeket [14], illetve [15] ismertet részletesebben.

A [16]-ben bemutatásra került alkalmazások offline módon képesek elemezni a tűzfalunk szabályrendszerét, és azt összevetve a hálózat felépítésével átlátható jelentéseket képesek készíteni arról, hogy milyen szolgáltatást honnan és hová engedélyezünk. Ennek a jelentőssége abban

rejlik, hogy a tűtfalak konfigurációs nyelve emberek számára gyakran kevéssé, vagy csak nehézkesen érthető meg ránézésre, míg egy jól összefoglalt jelentés könnyedén áttekinthető.

Az eszközök közül csak a másodikkal, az FA-val (Firewall Analyzer) foglalkozunk a tanulmány során, ez a másik bemutatott prototípus továbbfejlesztett verziója. A FA először a tűzfal routing táblája alapján létrehozza a hálózati topológia modelljét, majd a tűzfal konfigurációs fájlját, mely a szabályrendszert tartalmazza értelmezi. Mindezek után különböző általános lekérdezéseket futtat le a kialakult kapcsolati modelleken, melyek alapján ki tudja deríteni, hogy hol, milyen port van nyitva, milyen szolgáltatásokat engedélyezünk.

Az FA-val kapcsolatban fontos megjegyezni, hogy nagy előnye, hogy nem szükséges tényleges kapcsolatokat kiépítenünk ahhoz, hogy a tűzfal szabályrendszerét tesztelni tudjuk, így az kevesebb erőforrás-igénnyel végezhető, mintha infrastruktúrát kellene építenünk a teszteléshez.

5. Összegzés

A tanulmány során bemutattuk a tűzfalak kapcsán felmerülő alapfogalmakat, melyeket példákkal illusztráltunk. A teljesség igénye nélkül áttekintettük a tűzfalak fajtáit, kiemelve a fontosabb szempontokat, funkciókat.

Kitértünk a tűzfallal védett hálózatokat, illetve magát a tűzfalat célzó támadások egy lehetséges menetére, illetve annak céljaira. Ezen támadást két részre, felderítésre és a védelmen történő áthatolásra bontottuk fel.

Végezetül prezentáltunk néhány eszközt az üzemeltető, illetve a rendszergazda tárházából, melyek segítségével felügyelhetik, optimalizálhatják a tűzfal működését. Ennek keretein belül bemutattuk a monitoringot és a jelentéskészítést, illetve kapcsolódó szakcikkek anyagaiból ismertettünk olyan megoldásokat, melyek segítségünkre lehetnek különböző sebezhetőségek, hibák kiküszöbölésében.

Irodalom

1. *Wikipedia: DMZ (computing)* (elérés dátuma: 2014.11.03.) http://en.wikipedia.org/wiki/DMZ_%28computing%29
2. *Wikipedia: Demilitarizált zóna (informatika)* (elérés dátuma: 2014.11.03.) http://hu.wikipedia.org/wiki/Demilitariz%C3%A1lt_z%C3%B3na_%28informatika%29
3. *Wikipedia: Firewall (computing)* (elérés dátuma: 2014.11.03.) http://en.wikipedia.org/wiki/Firewall_%28computing%29
4. *Wikipedia: Tűzfal (számítástechnika)* (elérés dátuma: 2014.11.03.) http://hu.wikipedia.org/wiki/T%C5%B1zfal_%28sz%C3%A1m%C3%ADt%C3%A1stechnika%29
5. *Wikipedia: OSI model* (elérés dátuma: 2014.11.03.) http://en.wikipedia.org/wiki/OSI_model
6. *Netfilter – iptables project* (elérés dátuma: 2014.11.03.) http://www.netfilter.org/projects/iptables/index.html
7. *Zorp GPL* (elérés dátuma: 2014.11.03.) http://www.balabit.com/hu/network-security/zorp-gpl
8. *nmap* (elérés dátuma: 2014.11.03.) http://nmap.org/
9. István Szabó: *Tűzfal szabályok felderítése*. In: Híradástechnika, LXI. Évfolyam 2006/5., ISSN *0018-2028: 10-14.*
10. Zoltán Papp: A *számítógép-hálózatok tűzfalainak támadása*. In: Hadmérnök, 7/2 (2012): 335-341.

11. *Wikipedia: Firewalk (computing)* (elérés dátuma: 2014.11.03.)
http://en.wikipedia.org/wiki/Firewalk_%28computing%29

12. *RFC 792: Internet Control Message Protocol* (elérés dátuma: 2014.11.03.)
https://www.ietf.org/rfc/rfc792.txt

13. Ou, Xinming, Sudhakar Govindavajhala, and Andrew W. Appel. *MulVAL: A Logic-based Network Security Analyzer*. USENIX Security. 2005.

14. *The Heartbleed bug* (elérés dátuma: 2014.11.03.)
http://heartbleed.com/

15. *Wikipedia: Shellshock (software bug)* (elérés dátuma: 2014.11.03.)
http://en.wikipedia.org/wiki/Shellshock_%28software_bug%29

16. Mayer, Alain, Avishai Wool, and Elisha Ziskind. *Offline firewall analysis*. In: International Journal of Information Security 5.3 (2006): 125-144.