© Kiskapu Kft. Minden jog fenntartva

Behatolásérzékelő rendszerek

Pedro a különböző behatolásérzékelő rendszereket (IDS) veszi szemügyre, és bemutatja, hogyan is azonosítják a támadásokat.

Vajon Ön biztonságosnak tartja számítógép-hálózatát? Tudja-e, mi történik a hálózatán éppen ebben a pillanatban? Valamikor réges-régen éltek olyan hálózati rendszergazdák, akik hálózati biztonsági gondjaikra a megoldást egy egyszerű tűzfalban látták. A legutóbbi néhány évben szemtanúi lehettünk annak, hogy ez a szemlélet már a múlté. A hálózati rendszergazdáknak a furcsa jelenségekről szinte valós időben tájékoztató riasztóeszköz iránti igénye az IDS-rendszerek megjelenését eredményezte. Írásunkban az IDS-rendszerek válfajait és alapelveit mutatjuk be: a kiszolgálóalapú behatolásérzékelő rendszert (HBIDS), a hálózati behatolásérzékelő rendszert (NIDS), és egy új alapelveken nyugvó vegyes behatolásérzékelő rendszert (h-IDS). Megvizsgáljuk, hogyan értelmezhetjük, keletkezett adatokat, miként hozhatunk létre elektronikus aláírásokat a behatolásokat jelző adatmintákat, valamint a Linux-rendszeren működő behatolásérzékelő rendszereket, amilyen például a NIDS Snort-kezdeményezés.

Mit is takar a behatolásérzékelő rendszer elnevezés?

Az IDS olyan program, amely képes észlelni a szokásostól eltérő, a gépet, sőt a hálózatot is tönkretevő csomagokat és tevékenykedést. Az első IDS-rendszer még kiszolgálóalapú volt, a piacot hatalmába kerítő rendszer mégis az NIDS (hálózati behatolásérzékelő) lett. Általában mindig van egy készülék vagy program, amelyet hol érzékelőnek, hol ügynöknek neveznek. Esetenként egy vagy két hálózati kártyával felszerelt és válogatás nélküli üzemmódban dolgozik. Más szóval azt mondhatjuk, hogy ez az összes hálózati kártyára érkező csomagot befogja, és nem csak az egy bizonyos IP-címre érkezőket. Ily módon az IDS az összes hálózatba belépő csomagot ellenőrzi, megvizsgálja, hogy tartalmaznak-e gyanús karakterfüzéreket, azután dönt a megfelelő válaszlépésről: a tűzfallal folytatott párbeszéd közben új szabályokat hozhat létre az adott IP-cím kizárására, lapozási kérést vagy elektronikus üzenetet küldhet a biztonságért felelős rendszergazdának stb. Az NDIS-sel kapcsolatban fontos kérdés, hogy hová kell az érzékelőt telepítenünk: a tűzfalon belülre vagy kívülre? De hol is van az a legalkalmasabb hely? Hosszan lehetne erről vitatkozni, hiszen mindkét nézőpontnak akadnak támogatói, de kétségkívül a tűzfalon belülre és kívülre is telepített rendszer a legjobb választás.

Az IDS-ek válfajai és alapelvei

Az IDS alaposabb megértéséhez mindenekelőtt tudnunk kell, hogyan is működik. Két alapvető IDS-fajta létezik: az elektronikus aláírás, és az eltérésalapú modell. Az elektronikus aláírás avagy használatbeli visszaélést leíró modell a legáltalánosabban alkalmazott IDS-modell. Az ujjlenyomatok, illetve aláírások olyan minták, amelyek azonosítják a támadásokat a csomagban levő különféle lehetőségek alapján, úgymint a feladó címe, a célgép címe, a küldő és fogadó gép kapui, a zászlók és a hasznos teher és egyéb jellemzők alapján. Az ujjlenyomatok ezen gyűjteménye együttesen tudásadatbázist alkot, amelyet az IDS az összes csomaglehetőség összehasonlítására felhasznál, és ellenőrzi, vajon megegyezik-e valamelyik tárolt mintával. Az alábbiakban majd kitérünk a Nimda-féreg lenyomatára a Snort IDS-ben.

Az eltérésalapú modell a hálózaton belül a szokványostól eltérő viselkedést próbálja felismerni. Ennek megfelelően elsőként ki kell tapasztalnia, hogyan is működik a hálózat, azután a különböző mintákat fel kell ismernie, végül valamiféle üzenetet kell küldenie a konzolnak vagy az érzékelőnek. Az ilyenfajta IDS legnagyobb hátránya, hogy nem tudni, a hálózat a tanulási szakaszban vajon felmutatta-e az összes lehetséges viselkedésformát vagy sem. Ez a módszer ezért kezdetben nagyszámú téves riasztást eredményezhet.

A téves riasztások az IDS-től érkező jelzések, amelyek támadóról tájékoztatnak, ám valójában csak egy be nem állított változóról van szó, vagy például egy alkalmazás a szokásostól eltérő kapura küldött csomagot, és még mindig nincs szó valamiféle hátsó ajtóról. A nehézség elhárítása: a rendszergazdának időről időre figyelemmel kell kísérnie az IDS által küldött figyelmeztetéseket, majd el kell végeznie a rendszer finomhangolását.

A kiszolgálóalapú behatolásérzékelő rendszerek általában kiszolgálógépeken futnak, és csak olyan eseményeket észlelnek, amelyek kizárólag a programnak otthont adó gépre vonatkoznak. A HBIDS-rendszer fő célja a gép megóvása a külső változtatásoktól és a rossz szándékú lekérdezésektől. Az efféle IDS-ek fontosságát igazolják azoknak a rosszindulatú támadásoknak az elszaporodása, amelyek célja a weboldalak megrongálása, illetve támadócsomagok (rootkit) telepítése a számítógépekre, hogy onnan újabb gépeket fertőzzenek meg. A támadócsomagok a kalóz által a megrongálandó gépre telepített csomagok, amelyek hátsó ajtó nyitására alkalmas programcsomagokat tartalmaznak, naplóállományokat törölnek, hogy leplezzék magukat, vagy a `ps` és `nestat`-hoz hasonló parancsokat cserélik le, elrejtik a démonokat, vagy éppen kaput nyitnak meg.

Emellett a HBIDS szolgáltatásai között szerepel a támadások még megtörténtük előtti érzékelése is.

Tripwire

A Tripwire a Linux számára fejlesztett HBIDS-rendszerek egyik remek példája (lásd Linuxvilág 2001. szeptember, 34. oldal). A Tripwire HBIDS-rendszerként azonosítható, mivel állományépség-vizsgáló iránti igényünket elégíti ki. A Tripwire programmal a felhasználó meghatározhatja, a beállítóállományban pedig ki is jelölheti, hogy mely állománycsoportokat szeretné megvédeni a módosulástól, majd a Tripwire ezen állományok és állományjellemzők ellenőrzőösszegét fogja használni. Bármilyen változás esetén figyelmeztetést küld a rendszergazdának. A programmal „gyárilag” szállított beállítóállomány jó kiindulási alap, de a vakriasztások számának mérséklése érdekében a felhasználó nem mulaszthatja el a program beállítását. Fordítsunk különös figyel-

met a naplóállományokra. Nincs értelme felvenni őket a védendő állományok közé, hiszen tudjuk, hogy bármilyen esemény – mint amilyen a bejelentkezés is (login) – hatására növekednek. A Tripwire a `cron` ütemeződémonnal együtt használható különösen jól. A felhasználók ilyen módon önműködővé tehetik az ellenőrzési folyamatot, és előírhatják, hogy hol és mikor fusson.

PortSentry

A PortSentry – lásd ***Anthony Cinelli***-nek a PortSentry-t bemutató webhelyét a Linux Journal oldalán (➲ http://www.linuxjournal.com/article?sid=4751) találjuk – a Psionic Software cég Abacus-kezdeményezésének része, amely „az internetközösség számára ingyenes kiszolgálóalapú biztonsági és betörésvédelmi eszközkészlet létrehozását tűzte ki célul". A HBIDS-rendszerek egyik fontos fajtája ez, mivel képes a kiszolgálóknak címzett csomagok érzékelésére, és TCP-burkolókkal és az IP Tablesszel együtt használható. Ez az érzékelési mód hasznos, mivel a kapupásztázás gyakran a támadások előhírnöke. A PortSentry képes TCP- és UDP-pásztázások érzékelésére, kimutatva azokat a kiszolgálógépeket, ahol a jelzett szolgáltatás éppen a lekérdezett kapuban fut. A következő lépés a foltok és frissítések ellenőrzése, sőt, ha szükséges, még elérésvezérlő listákat (ACL) is létre kell hozni a TCP-burkolókkal, hogy a pásztázó kiszolgálóval mindenféle jövőbeni kapcsolatot megszakítsunk. Ezenkívül a tűzfalon is létrehozhat szabályokat, vagyis például egy olyat, hogy az IP Tables minden, a pásztázó géptől származó csomagot ejtsen el. Az alábbiakban egy PortSentry-bejegyzést mutatunk be a `syslog` rendszernaplóból:

```
Dec 9 03:03:17 mobile portsentry: [701]:
➥attackalert:
   TCP  SYN / Normal scan from host:
   200.185.61.132 / 200.185.61.132 to TCP
➥port: 111
Dec 9 03:03:17 mobile portsentry: [701]:
➥attackalert:
   Host 200.185.61.132 has been blocked via
➥wrappers
   With string "ALL: 200.185.61.132"
Dec 9 03:03:17 mobile portsentry: [701]:
➥attackalert:
   Host 200.185.61.132 has been blocked via
➥dropped
   Route using command "/sbin/iptables -I
   INPUT -s 200.185.61.132 -j DROP"
```

Swatch

A Swatch olyan rendszernapló-figyelő program, amely értesíti a rendszergazdát, amint a rendszernapló-állományban – például a */var/log/messages*-ben – felbukkan valamilyen előre megadott karakterlánc. A következő példában létrehoztam egy a Sort beállítására szolgáló egyszerű állományt, és a `snort`, valamint a `portsentry` elnevezések figyelését írtam elő. Azt is beállítottam, hogy találat esetén az eredményeket különböző színnel jelenítse meg, és a gép ekkor hangjelzést is adjon:

```
watchfor /snort/
echo red
bell
watchfor /portsentry/
echo blue
bell
```

Találat esetére a Swatchnak előírhatom, hogy hová küldjön üzenetet vagy milyen parancsot hajtson végre. Az előbbi Swatch-beállítóállomány az alábbi üzenetek megjelenését eredményezte:

```
Dec 9 03:22:53 flamengo snort[3268]:
➥[1:1256:2]
    WEB - IIS CodeRed v2 root.exe access
➥[Classification:
    Web Application Attack]: [Priority: 1]:
    {TCP} 200.31.36.11:253 ->
200.204.68.154:80
Dec 9 03:22:53 mobile portsentry [701]:
attackalert:
     TCP SYN / Normal scan from host:
     200.185.61.132 / 200.185.61.132 to TCP
port: 111
```

LIDS

A LIDS rövidítés a Linux behatolásérzékelő rendszert jelöli, amely a Linuxot – a telepített rendszerfoltok révén – kivételes szolgáltatásokkal ruházza fel.

A szolgáltatások között megtalálható az állománysértetlenség- és a folyamatvédelem, valamint a kapupásztázás-érzékelés. Az első kettő részletesebb magyarázatot igényel. Az állomány- és a folyamatvédelem még a rendszergazdával szemben is működik. Ez olyankor bizonyul rendkívül hasznosnak, amikor a kalóz egy rendszerbiztonsági rést – amilyen mondjuk egy átmeneti tár túlcsordulása – kihasználva megkaparinthatná a rendszergazdai jogosultságokat és bármit szabadon megtehetne, tehát akár támadócsomagokat is telepíthetne, naplóállományokat módosíthatna, vagy HTML-oldalakat törölhetne stb. Ezekkel a szolgáltatásokkal ACL-eket határozhatunk meg, amelyek szabályozhatják az állományokhoz való hozzáférést, jelszavakat tartalmazhatnak ezek olvasásához, illetve módosításához, a nem kívánt változások pedig elkerülhetők, származzanak azok meg nem hatalmazott felhasználótól vagy magától a rendszergazdától. Ugyanez igaz a folyamatokra is, mert ez a rendszer megakadályozza a bináris állományok, illetve a démonok megváltoztatását is. E megoldás további kellemes szolgáltatása, hogy a rendszermag területén kapupásztázást képes végezni.

NIDS

A hálózati behatolásérzékelő rendszerek az IDS-ek azon fajtái, amelyek képesek a hálózatot érintő támadásokat érzékelni. Fontos vitatéma, hogy hová is kellene telepíteni őket. Lehet olyan hálózati elrendezést találni, amelyben a tűzfal előtt találjuk őket, és elképzelhető olyan is, amikor mögötte. Mint már említettem, mindkettő mellett jó érveket lehet felsorakoztatni: csak a helyi igények dönthetik el, mikor melyik szükséges. Ezekben a példákban a nyílt forrású Snortot fogom használni.

Snort

Marty Roesch fejlesztette a Snortot, és jelenleg már ezernél is több szabály alapján érzékeli a támadásokat az egyszerű pásztázásoktól kezdve egészen a legújabb fajta `ssh CRC32` biztonsági résig – részletesebben lásd ***Nalneesh Guar*** Snort: IDS tervezése az Ön vállalkozása számára című írását a Linux Journal honlapján (➲ http://www.linuxjournal.com/article.php?sid=4668). A Snort óriási előnye, hogy az igényeknek megfelelően képes új szabályt alkotni. Ezzel szemben más IDS-szállítóknál általában meg kell várni a legfrissebb csomagok megjelenését, a támadások nyomán a rendszer testreszabható és új lenyomatok

© Kiskapu Kft. Minden jog fenntartva

© Kiskapu Kft. Minden jog fenntartva

készíthetők. A fenti esetre kiváló példa volt a `wu-ftpd` esete 2001 decemberében. Mindössze néhány órával a biztonsági rés nyilvánosságra hozása után a Snort-szűrő már hozzáférhető volt a biztonsági levelezőlistákon. A Snort képes együttműködni a tűzfalakkal is, vagyis például a Check Point FireWall-1-gyel, az OPSEC lehetőség felhasználásával, vagy a bedolgozó programok segítségével, hogy együttműködjön a Linux IP Tables programjával.
Amellett, hogy a Snort hatalmas ujjlenyomat-adatbázissal rendelkezik, és főként a használatbeli visszaélésleíró modellen alapszik, bétaszolgáltatásként az eltérésalapú modellt is felvonultatja. E szolgáltatás, amelynek a SPADE keresztnevet adták, az összegyűjtött adatok statisztikai elemzését adja, és megkísérli meghatározni a normális viselkedés jellegét. Amint az a nyílt forrású alkalmazásoknál lenni szokott, számos kapcsolódó alkalmazással használhatjuk.
Kellemes alkalmazás a Silicon Defense cégtől a SnortSnarf, amely a Snort által összegyűjtött adatokból HTML formátumú jelentéseket készít. A Snort egyetlen hálózati kártyával is megelégszik. Más NIDS-rendszerektől eltérően, amelyek két hálózati kártyát igényelnek – egyet az adatgyűjtéshez, egy másikat pedig a rendszergazdai felület számára – a Snort mindössze egy hálózati kártyával is dolgozni tud csomagválogatás nélküli üzemmódban, és a felügyeleti feladatot is képes ellátni, felvéve és frissítve az új szabályokat.

Vegyes IDS

Újabban egyre népszerűbbé válik egy másfajta elgondolás: a vegyes behatolásérzékelő rendszer. ***Marcus Ranum***, a Network Flight Recorder (NFR) alapítója úgy véli, hogy „a vegyes IDS-rendszerek a HBIDS- és NIDS-rendszerek erejének és gyengéinek érdekes vonalát képviselik". Ez azt jelenti, hogy ebben a rendszerben a NIDS-rendszer számos jellemvonása felbukkan, ám ezúttal gépenkénti alapon. Ebből több előny is származik, minthogy a kiszolgálóra irányuló behatolási kísérleteket egyediként próbálja azonosítani; az általa elemzett csomagokban csak a célgép IP-címét tartalmazó csomagok fordulnak elő. Az ilyen fajta IDS-nek nagy hátránya, hogy minden egyes kiszolgálógépre telepíteni kell.

Prelude IDS

A Prelude is a vegyes rendszerek csoportjába tartozik. Két részre osztható: a Prelude NID elnevezésű érzékelőre, amely a csomagok befogásáért és elemzéséért felelős, és a jelentéskészítőkiszolgálóra, amelyet az érzékelő a behatolási kísérlet jelzésére használ. A Prelude egy érdekes jellemvonása külön említést érdemel: képes szabályokat kiolvasni a Snort IDS-ből, más szóval kész futtatható szabálygyűjteménnyel rendelkezik. Webhelyéről bármelyik Prelude IDS-ről képes a szabályokat leolvasni. A Prelude-öt a fürtözési elvet figyelembe véve építették, ez a magyarázata annak, hogy a jelentéskészítő-kiszolgálót egy másik gépre telepítették, amely képes a begyűjtött adatokat felhasználóbarát formába önteni, például HTML-állománnyá alakítani.

A lenyomatok értelmezése és létrehozása

Mint már a cikk fentebbi részéből kiderült, a lenyomatok tulajdonképpen támadási minták. Fontos, hogy megértsük, miképpen működnek. Szükség esetén, vagy egy új biztonsági rés felfedezésekor magunk is létrehozhatunk ilyeneket. Példáink megvilágítják, hogyan kezeli a Snort az elektronikus ujjlenyomatokat. 2001 második felében új és nagyon hatékony férgek jelentek meg a Világhálón: Code Red, Code Red II és a Nimda. A támadások idején a Linux-felhasználók – jómagam is ezek egyike voltam – roppant szerencsésnek érezték magukat, mivel a férgek főleg a Microsoft Internet Information Serverehez kapcsolódtak. Ezeknek a férgeknek eltérő volt a mintája, például a Microsoft IIS-en keresztül megpróbálta elérni a ***cmd.exe*** állományt. Ennek tudatában könnyű volt megalkotni a Nimda Snort-szabályt, amint azt „Az IDS-ek válfajai és alapelvei" című részben már olvashattuk:

```
alert tcp: $EXTERNAL_NET any ->
➥$HTTP_SERVERS:80
msg: "WEB-IIS cmd.exe access"
content: "cmd.exe"; nocase
classtype:web-application-attack; sid:1002;
➥rev:2;)
```

Rendben, és mit jelent mindez? A Snort-értékek valójában két csoportba sorolt értékek sorozatát alkotják, amellyel a Snort figyelmét ráirányíthatjuk bizonyos dolgokra. Az első rész a szabályfejléc, és az első zárójelig mindenféle megtalálható benne. Az első érték azt mondja meg, mi kell tenni, ha a csomagegyezést talál. Az `alarm` üzenetriasztást fog végezni, és naplózza a csomagot. A második érték jelzi a Snortnak, hogy milyen protokollt szeretnénk használni, jelen esetben csak TCP-t. A következő öt érték mutatja a feladó IP-címét, a csomag irányát, a címzett IP-címét és adatkapuját. Ilyen módon biztosak lehetünk afelől, hogy a hálózatunkon kívül eső címről, a 80-as kapun – a webkiszolgálók rendszerint éppen ezen a kapun figyelnek – érkező csomagokat a belső szabálylehetőségek alapján fogják ellenőrizni. A szabálylehetőségek szakaszriasztási üzeneteket és a csomagok ellenőrizendő részéről adatokat tartalmaz. E vizsgálat eredményének ismeretében a megfelelő tevékenységet hajtja végre.
A példánkban szereplő szabálylehetőségek:

- `msg: WEB-IIS cmd.exe access` – üzenet, a riasztás leírása.
- `flag: A+"` (zászló) – logikai műveletjel, a csomagban levő összes zászló ellenőrzése.
- `content: cmd.exe` (tartalom) – beállítja a hasznos terhet.
- `nocase:` – a lehetőség megengedi a kis- és nagybetűk megkülönböztetésének figyelmen kívül hagyását.
- `classtype: web-application-attck` (osztálytípus: webalkalmazást érő támadás).
- `sid:1002` – `sid`, azaz Snort-azonosító.
- `rev: 2` – a szabály változatszáma.

További érdekességek

Marcus Ranum: Jelentés a behatolásérzékelő rendszerekről (IDS) ➲ http://www.intrusiondefense.net
A hálózati behatolásérzékeléssel kapcsolatos témában Stephen Northcutt, Judy Novak és Donald McLachlan: An Analyst's Handbook (A rendszerelemző kézikönyve, 2. kiadás), New Riders, 2000.
A SANS Intézet honlapja
➲ http://www.sans.org/infosecFAQ
A Silicon Defense cég webhelye
➲ http://www.silicondefense.com
A Snort webhelye ➲ http://www.snort.org

A Snort felhasználói kézikönyvben harmincnál is több lehetőség található a felhasználói igények kielégítésére. Úgy véli, túlságosan bonyolult? Á, dehogy! Tegyünk egy próbát, és jelezzük szabállyal a hálózatról pornográf oldalakra történő elérési kísérletet:

```
alert tcp: $EXTERNAL_NET any ->
➥$HTTP_SERVERS:80
msg: "Web porn access attempt"
content: "Free porn"; nocase;flags:A+);
```

A létrejött adatok elemzése

Egy szolgáltatáshoz tartozó kapupásztázás olyan, akár a `portmap` (111-es kapu), amelyről közismert, hogy számos biztonsági rést rejt magában, amit a PortSentry biztosan észrevenne.

```
Dec 9 03:03:17 flamengo portsentry [701]:
➥attackalert:
    TCP SYN / Normal scan from host:
     200.185.61.132 / 200.185.61.132 to TCP
     ➥port:111
```

A naplóállományok értelmezési képessége kulcsfontosságú, hogy a behatolást vizsgáló vagy a biztonsági szakember pontosan tudja, mi a teendője egy adott helyzetben. A PortSentry-ből származó fenti riasztás a `syslog` rendszernapló állományból származik. Ez a riasztás azt állítja, hogy december 9-én 3:03-kor a ***flamengo*** nevű kiszolgálógép, amelyre a PortSentry telepítve van, egy SYN-zászlós normális kapupásztázást talált a 111-es TCP-kapun, amely rendszerint a `portmap` szolgáltatást futtatja, ezúttal a `200.185.61.132`-es címről.

Összegzés

A tűzfal elsődleges biztonsági elem a hálózatban, azonban képtelen már megnyitott szolgáltatásokra irányuló támadást érzékelni, mint például egy DNS- vagy egy webkiszolgálóra irányuló támadást.
Az IDS egyedüli biztonsági elemként nem fogja megoldani gondjainkat: amennyiben elvégzi rendszerünk testreszabását, segíteni fog a figyelmeztetések helyes felismerésében, ha a hálózatban szokatlan tevékenység zajlik, vagy ha a kiszolgálóhoz vagy a hálózathoz illetéktelen hozzáférési kísérlet történne. Ezekkel az adatokkal – a behatolási IP-címmel együtt – már fel lehet keresni a rendszergazdát, és tájékoztatni lehet (a többi felhasználót is), hogy mi zajlik a hálózaton.

Linux Journal május, 97. szám

Pedro Bueno
(bueno@ieee.org) korábban a Lucent Technologies adattechnikai tervezőmérnöke, jelenleg az Open Communications Security biztonsági kérdésekkel foglalkozó mérnöke. Önkéntesként résztvesz a Best Linux-változat fejlesztésében. Kedvenc időtöltése a foci mellett a Snort által létrehozott riasztások megfejtése.

© Kiskapu Kft. Minden jog fenntartva