Az ujjlenyomat általi azonosítás meggátolása

„Ha nem akarunk harcolni, meggátolhatjuk, hogy ellenfelünk harcba szálljon velünk, mégha alig láthatók is a földön táborhelyünk körvonalai. Csak annyit kell tennünk, hogy valami furcsát és érthetetlent vetünk elébe.”

(Szun Cu: A háború művészete)

Az operációs rendszer ujjlenyomat alapján történő azonosítása az a folyamat, melynek során a távoli géptől visszakapott adatok jellemzői alapján meghatározzuk, milyen operációs rendszert futtat a távoli számítógép. Egyes esetekben ez csak annyit jelent, hogy kapcsolódunk a géphez és elolvassuk a szolgáltatás bejelentkező szövegét, máskor viszont bonyolult, mint például a TCP-kezdőadatok és zászlók (flagek) statisztikai elemzése. A kívülállók képesek általános adatokat megállapítani, mint például a gépen futó operációs rendszer típusát, azáltal, hogy megkeresik a TCP-verem operációs rendszerekre egyedileg jellemző megvalósításának a különbségeit. Egyes esetekben ezek a különbségek nagyon részletes adatokat is elárulhatnak, például az operációs rendszer változatszámát és a processzor típusát.

A gép operációs rendszerének pontos azonosítása révén a támadó jól tájékozott és pontos támadást tud intézni a célgép ellen. Ebben az átmeneti tár túlcsordulásaival teli világban a támadónak talán csak arra az alkalomra van szüksége, hogy pontosan megtudja az operációs rendszer és a processzor típusát. Az olyan programok használatával, mint például a linuxos Netfilter, a rendszergazdák ki tudják kerülni az operációs rendszer ujjlenyomata általi pontos azonosítását, néhány esetben pedig még meg is tudják hamisítani a külső erők által gyűjtött adatokat. Bár ezeket a szokásokat semmiképpen sem szabad megbízható biztonsági megoldásnak tekinteni, néha elbizonytalaníthatja, sőt össze is zavarhatja a betörni szándékozót az, ha célja titokzatos hálózati egységnek látszik.

Bár az ujjlenyomat általi azonosítás elkerülése remekül eltitkolja, milyen operációs rendszert futtat valójában a gép, semmiképpen sem teszi biztonságossá különféle sebezhető pontjain. A biztonsági eljárások és irányelvek célja az, hogy magasabbra emeljék a rendszer tönkretételéhez szükséges szakértelem szintjét –, az eltitkolás csak a valódi célpont elrejtését kísérli meg. Mégha a külvilág úgy is látja, hogy rendszereden a Microsoft IIS5 fut, nem véd meg téged, ha mondjuk a Sendmail egyik sebezhető változatát használod, és egy betörőpalánta (script kiddie) önműködő pásztázója kísérel meg betörni hozzád. Az ujjlenyomat általi felismerés elkerülésének célja a támadások elterelése, nem pedig a megakadályozása.

A felfedezés módszerei

Mielőtt egy lehetséges támadót azzal próbálnánk meg eltántorítani, hogy becsapjuk az operációs rendszerünket illetően, meg kell ismerkednünk az operációs rendszerek ujjlenyomat általi azonosítására használt eszközökkel és eljárásokkal.

A „támadó” kifejezést itt bő értelemben használjuk, és magában foglalja mindazokat, akik megpróbálják azonosítani egy gép ujjlenyomatát, és azokat is, akiknek szándékában állhat, hogy kárt okozzanak egy rendszerben. A biztonság a lépések és az azt követő ellenlépések egymást követő sorozata volt, és az író felfogása szerint mindig is az lesz. Ha megismerkedünk az ilyen támadásokra használt eszközökkel és eljárásokkal, akkor nem csupán a jelenlegi fogásokra tudunk felkészülni és terveket készíteni, de bepillantást nyerünk abba is, mit hozhat a jövő.

Számos nyilvánosan elérhető eszköz van, melynek célja az operációs rendszerek ujjlenyomat általi azonosítása. Úgy tűnik, ezek közül az eszközök közül a legnépszerűbb az nmap (➲ http://www.insecure.org/nmap/index.html), melyet ***Fjodor***, az Insecure.org webhely fenntartója készített. Az Nmap számos módszert használ, hogy megpróbálja felismerni a gép operációs rendszerét hálózati szintről, ezek között vannak kezdetleges megközelítésűek és bonyolultabbak is, melyekhez a TCP/IP-protokoll és a protokollok általános jellemzőinek alapos ismeretére van szükség. Íme, néhány figyelemre méltó módszer az Nmap használta módszerek közül:

- ***FIN-próba*** – ha a támadó gép egy nyitott kapujára olyan csomagot küld, amelyben csupán a FIN-zászló van beállítva, akkor bizonyos, a kérésre válaszoló gépekről adatokat szerezhet be. Ez a viselkedés nincs összhangban az RFC-kkel, ezért nagyon jól jelzi az operációs rendszer fajtáját.
- ***TCP ISN mintavétel*** – A TCP-csomagok ISN- (kezdősorszám) mintavételezése értékes módszer lehet a távoli gépek azonosítására és besorolására. Azáltal, hogy a támadó bizonyos mintákat keres az ISN-ekben, a támadó megalapozott találgatásba bocsátkozhat a gép operációs rendszerét illetően.
- ***ICMP-hibaüzenetek küldése*** – az ICMP- (internetvezérlő-üzenetprotokoll) hibaüzeneteket használva a támadó hasznos adatokra tehet szert a gép válaszai alapján. Különös érdeklődésre számot tartó területek az ellenőrző összegek, a hibaüzenet visszhangjának épsége, és a válaszüzenetekben található TOS- (szolgáltatástípus) mezők.
- ***TCP beállítások*** – minden TCP-veremnek talán a legárulkodóbb jellegzetessége, hogyan kezeli a választható TCP-zászlókat és adatokat. A géphez intézett valóságos kérések és a változó ablak- és szakasz- (segment) méret segítségével meghatározható, milyen operációs rendszert futtat a számítógép, annak alapján, hogy hajlandó-e elfogadni ezeket a választható változókat, vagy válaszol-e rájuk.

Bár az operációs rendszer ujjlenyomat alapján való azonosításának mindezen módszerei csomagszinten működnek, nagy gondot kell fordítani rá, hogy megértsük gépünket a szolgáltatások szintjén. Lehet, hogy a támadó rendszerezi és összehasonlítja a csomagok felépítését, de gyakran csak annyit tesz, hogy lekéri a webkiszolgálótól a HTTP-válaszfejléc ***Server*** (kiszolgáló) mezőjében szereplő értéket. Ha tudjuk, mely szolgáltatások azonosítják magukat készségesen, és ami még fontosabb, ha ismerjük az operációs rendszer felépítését, az további lehetőségeket is megmutat nekünk, amelyek révén távolról adatokat lehet szerezni.

Az ügyfélprogramok csendessége (vagy ennek hiánya) is

remek módszer, hogyan szerezhetünk adatokat egy számítógépről. A többi lehetőségtől eltérően ez a folyamat teljesen passzív is lehet. Annak megfigyelésével, hogyan mutatkozik be egy ügyfélalkalmazás valamely szolgáltatásnak, ésszerű találgatásra vállalkozhatunk az operációs rendszerre és a felépítésre vonatkozóan egyaránt. Az ügyfelek közül többnyire a webböngészők, a levelezőprogramok és az IRC-ügyfelek a leginkább vétkesek. Ha IRC-zünk, és lekérdezzük az egyik felhasználó CTCP-változatát, válaszként pedig a ***mIRC32 v5.81 K.Mardam-Bey***, szöveget kapjuk, akkor alapos okkal gondolhatjuk azt, hogy a számítógépen a Windows operációs rendszer valamelyik változata fut.


1.

Végezetül ott van a gyenge pontok kipróbálása. Bár nem túl tapintatos, azért hasznosnak bizonyulhat a számítógép operációs rendszerének megállapítására. Az operációs rendszerre egyedileg jellemző szolgáltatásmegtagadási (denial-of-service) támadások sorozatának a megindítása által a kívülálló próbálkozhat, és megállapíthatja, sebezhető-e a számítógép. Ezzel meghatározható, milyen besorolású rendszert futtat a gép, általában még a foltok szintjén is. A Windows-közösség hálás lehet Fjodornak és az ujjlenyomat általi azonosítást végző eszközök többi fejlesztőjének, amiért úgy döntöttek, hogy ezt a módszert nem foglalják bele szokásos pásztázási eljárásaik választékába.

Mire jó az ujjlenyomat alapján történő felismerés meggátolása?

Ha eddig elolvastad a cikket, egész biztosan érdekel, miért vállalná magára valaki a bonyodalmakat azért, hogy megakadályozza az operációs rendszer ujjlenyomata általi azonosítást. Jó kérdés! Úgy gondolom, az indítékok személyenként eltérőek. Mindenkinek megvan az oka, amiért el akarja rejteni, vagy éppenséggel nem akarja elrejteni az operációs rendszerét. Vannak, akiknél a fokozott titokzatosság a homályban való meghúzódás és a belső kényelem érzését váltja ki. Hasonlóan azokhoz, akik Telnet bejelentkező képernyőjükről eltávolítják a változatszámot tartalmazó üdvözlőszöveget, de a távoli helyek biztonságos elérésére a Telnetet használják az SSH helyett titokzatos, de szakmai szempontból kevésbé biztonságos. Mások esetében az operációs rendszer eltitkolása lehetővé teszi, hogy minden részletre kiterjedően beállítsák behatolásjelző-rendszerüket (IDS), ugyanis nemcsak azt tudják viszonylag jól, mi jöhet be a hálózatukba, hanem azt is, milyen adatok mehetnek ki belőle (titokzatos, óvatos és remélhetőleg biztonságos). Néhányaknak talán még szükségük is van a biztonságosságra, ugyanis minden hálózat, melyre rácsatlakoznak, ellenséges hálózatnak bizonyulhat; és minél jobban elrejtik az operációs rendszerüket, annál szélesebb lehetőségük nyílik rá, hogy elvégezzék mindenkori feladatukat, anélkül, hogy észrevennék őket (titokzatos, óvatos, biztonságos, és valószínűleg épp a leveleidet olvassa). Végül ott vagyunk mi, akik egyszerűen szórakozásból tesszük ezt, azért, mert képesek vagyunk rá; és mert nem kis örömet találunk abban, hogy be tudjuk csapni a körülöttünk levő ismeretleneket, akik állandóan pásztázókat irányítanak ránk (igen, bűnösek vagyunk a vád tárgyában).

A meggátolás módja

Itt az idő, hogy szerencsét próbáljunk az ujjlenyomat általi azonosítás elkerülésében. Ismerve a számítógépek operációs rendszerének megállapítására használt népszerű eljárásokat, ezeket az elképzeléseket vissza tudjuk fejteni, ami segíteni fog, hogy eltitkoljuk valós egyéniségünket.

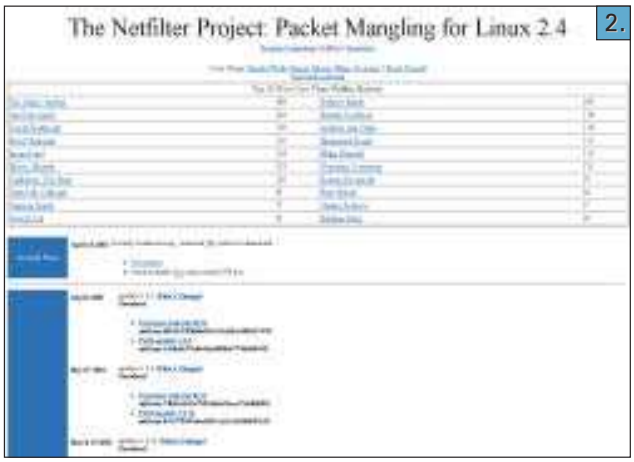
Először is meg kell bizonyosodnunk róla, hogy minden biztonsági folt a helyén van és a rendszer biztonságos. Mint korábban kijelentettem, a titokzatosságra csak azután törekedhetünk, miután gondoskodtunk a biztonságról. Biztosan akadnak, akik nem értenek ezzel egyet, és csupán a titokzatosságra támaszkodnak ahhoz, hogy biztonságossá tegyék rendszerüket, de vajon mire jó a titokzatosság, ha a 33. számú betörőpalánta önműködő programja rendszergazdai jogosultságot szerez a gépeden még ma este? Fogadni merek, hogyha egyszer sikerült rendszergazdai hozzáférést szereznie, akkor nem azzal fog foglalkozni, hogy kitalálja, a Linux melyik változatát futtatod.

Másodszor, meg kell figyelnünk a szolgáltatásainkat. Összhangban vannak azzal az operációs rendszerrel, melynek a használóiként szeretnénk feltüntetni önmagunkat? A legtöbb esetben ez nem okoz nagy gondot, hiszen a Unix-környezetek többsége hasonló vagy ugyanolyan szolgáltatásokat alkalmaz. De ha Windows-gépként vagy éppenséggel Cisco-útválasztóként szeretnéd feltüntetni magad, akkor nem biztos, hogy előnyös, ha látszik, hogy fut az IRC-d. Tegyünk erőfeszítést azért, hogy a szolgáltatásaink összhangban legyenek az álcaként használt számítógéppel.

Ha már a szolgáltatásoknál tartunk, az is jó ötlet, ha a szolgáltatások forráskódját átfésüljük, és megkeressük üdvözlőszövegeit és általános azonosítóit. Néhány árulkodó azonosító jel lehet az ASP-oldalak támogatása vagy az olyan webtartalom, melyet tömörített gzip-formátumban szolgáltatunk. Ismétlem, neked kell megszabnod, milyen szintű rejtőzködést és az álcaként használt géppel való egyezést próbálsz meg elérni.

Ezután azt kell megvizsgálnunk, hogyan jelenik meg gépünk a hálózaton, szembeállítva azzal, ahogyan az álcaként használt gép megjelenik a hálózaton. Ennek megkönnyítésére azt javaslom, tanulmányozzuk a már leírt anyagokat, vagyis azokat a friss ujjlenyomat-állományokat, amelyeket maguk a programok használnak. Időt kell szakítanunk nemcsak annak megállapítására, hogyan válaszol az álcaként használt gép a szokásos kérésekre, hanem arra is, mely egyedi zászlókat támogatja a TCP-ben. A TCP-zászlók hasznos adatokat szolgáltatnak a kívülállók számára, hogy megállapíthassák, milyen operációs rendszert futtatunk. Az ujjlenyomat-állományok nem tartal-

mazzák a gép által adható összes lehetséges választ, hanem csak olyan egyszerű módszereket, melyek megismételhetően működnek. Attól függően, milyen fokú rejtőzködést szeretnénk elérni, érdemes lehet megvizsgálni azokat az ujjlenyomatadatokat is, melyeket az Nmap nem használ (OSPF, OOB, IPv6 stb.). De az is előfordulhat, hogy az alaposság öröménél nagyobb súllyal esnek latba az ezeknek az adatoknak az összegyűjtéséhez szükséges munkával töltött (alvás nélküli, nem álmatlan) éjszakák.



2.

Végül hoznunk kell egy döntést. Cselesek vagy pedig megszállottak leszünk? Ha az utóbbi a választunk, akkor valószínűleg az ügyfélalkalmazásaink meghamisításával folytatjuk. Mint korábban említettem, a gép ügyfélprogramjai hajlamosak mindenféle adatokat kiszolgáltatni (közvetlenül vagy közvetve) a rendszerről. Előző példánkban egy IRC-ügyfél elárulja magáról, hogy Win32-höz készült, de vannak a számítógépek azonosításának kifinomultabb módszerei is, így például, amikor a kimenő levelek fejlécét olvassák el. Megismétlem, minden attól függ, hány alvás nélküli éjszakát vagy hajlandó ráfordítani, hogy rendszered megfeleljen az általad kívánt jellemzőknek.

A lehetséges gondok

Az operációs rendszer ujjlenyomat általi azonosításának elkerülése ugyanolyan, mint a biztonság bármely más nézőpontja: tervezésre, megfelelő megvalósításra, és mindenekfelett megértésre van szükség. Ha a biztonsági irányelveket nem megfelelően ültetik át a gyakorlatba, a rendszer még sebezhetőbbé válhat, mintha egyáltalán nem alkalmaznák őket.
A népszerűség elismeréshez vezet. A számítógépes programok legtöbbjénél az ismertség nagyszerű dolog; a figyelmet ráirányítja kemény munkádra és eltökéltségedre. Az operációs rendszerek ujjlenyomat alapján történő felismerése esetén munkád elismerése ellened dolgozik. Ha népszerű eszközt vagy csomagot használsz, végül fel fogják fedezni sebezhető pontjait és különlegességeit – ez elkerülhetetlen. Ugyanezek a programra jellemző azonosítók képessé tesznek majd másokat, hogy az ujjlenyomat alapján pontosabban azonosítsák ellenlépéseidet, mint magát az operációs rendszert. Csaknem minden operációs rendszer igyekszik véletlenszerűvé tenni TCP ISN-adatsorozatát, így próbálva meggátolni a TCP-eltérítést és a rendszer ellen intézett bonyolultabb támadásokat. Ha az általad választott eltérítési eljárás megpróbálja módosítani a TCP-kezdőadat számait, nagy gondot kell fordítani arra, nehogy alulbecsüljük, illetve leértékeljük ezt a szolgáltatást, és kiszolgáltassuk gépünket az ilyen jellegű támadásoknak. Mint minden a rendszerbe bekerülő számítógépes programcsomag esetén itt is elsődleges fontosságúnak tekintendő az alkalmazás biztonságossága. Az elrejtőzési folyamat része a meglevő szolgáltatások álcázása; a másik része az olyan kód, melynek a forgalom szűrése és a hálózaton folyó forgalmunk álcázása a feladat. Nagy gondot kell fordítani arra, hogy az erre a feladatra használt alkalmazás a jó programozási eljárásoknak és a szigorú ellenőrzésnek köszönhetően megfelelően biztonságos legyen. Csak egyetlen rosszul átgondolt `strcpy()` hívásra van szükség, hogy ebből az előnyből veszélyforrás legyen.
Az egyik korábban említett rejtőzködési mesterkedés az, hogy megváltoztatjuk a program önmagát azonosító üdvözlőszövegét. Elővigyázatosságra van szükség, mert néhány kiegészítő programcsomag ezeket az üdvözlőszövegeket használja, hogy megállapítsa, együttműködik-e a jelenlegi rendszerprogrammal.

Kockázat és előnyök

Most, hogy leszögeztük, a rejtőzködés nem jelent biztonságot, meg kell vizsgálnunk egy másik szempontból is ezt a folyamatot, mégpedig a hatásfokéból. Mivel minden jó rejtőzködési beállítás nagy mennyiségben szűri a forgalmat, igen valószínű, hogy a rendszer teljesítménye megsínyli ezt. Ha a kiszolgálód 10 000 ügyfél weblapjának ad otthont, nagyobb jelentősége van a teljesítménynek, mintha csak fel van állítva egy linuxos géped, melyen a barátaiddal nézegeted a leveleidet és IRC-zel. Rendszergazdaként el kell döntened, melyik a nagyobb előny neked (és felhasználóidnak): a teljesítmény vagy a titoktartás.

Az elmélet bizonyítéka

Azért, hogy szemléltessem, mennyire használható és viszonylag egyszerű az ujjlenyomat általi azonosítás, mellékeltem egy Linuxon, kicsi felhasználói területen futó mintaalkalmazást (OSFPE), mely a Netfilter rendszermagmodult használja (lásd az 1. listát a 20. CD-n). Az operációs rendszer ujjlenyomat általi azonosításának elkerülése egyre hatékonyabb lesz az olyan programok használatával, mint például a Linux alatti Netfilter (lásd még előző írásunkat). Egymás után bukkannak fel a hasonló módosítások és alkalmazások; BSD-ben ezt a feladatot az IP Filterrel és viszonylag kis mennyiségű kóddal lehet megoldani. A Windowst használók (akik messze a legnagyobb hátrányban vannak ezen a területen) fokozatosan fedezik fel a módszereket, melyekkel beburkolhatják TCP/IP-rendszerű adatforgalmukat, és a win32 alatti Libpcap megszületésével elfoghatják és meghamisíthatják a csomagokban küldött saját válaszukat.

A Netfilterről röviden

A Netfilter, ahogy szerzője mondja, „keretprogram a csomagok keveréséhez”. Érdekesen hangzik, nemde? A Netfilter összekapcsolódik a Linux rendszermaggal (pontosabban a 2.4.x és afölötti változatokkal), és mindegyik protokollhoz kezelőket jegyezhet be. Ha a megfelelő szabályok helyükön vannak, ezek a kezelők elfogják a meghatározott szabályoknak megfelelő bejövő és kimenő hálózati forgalmat. A csomagok ezután feldolgozásra kerülnek és NF_DROP jelzést kapnak, ha el kell ejteni, NF_ACCEPT jelzést pedig, ha a csomagot rendesen kell kezelni a veremben, végül NF_QUEUE jelzést kapnak, ha a csomagot sorba kell állítani, hogy fel lehessen dolgozni a felhasználói területen. Ha a csomagot a felhasználói területen való feldolgozásra sorba állítják, az `ip_queue` beállítja a sorba; ezután bármely, a felhasználói területen futó program

aszinkron módon feldolgozhatja a csomagokat, amely bejegyezte magát az ilyen fajta csomagokra. Amikor ezek az alkalmazások kivesznek egy csomagot a sorból, képesek módosítani, elfogadni vagy elutasítani azt. Ha a csomagot elfogadják, továbbadják a következő olyan alkalmazásnak, amely bejegyezte magát az ilyen csomagokra. Ha a csomag NF_DROP jelzést kap, akkor elejtésre kerül, és a szóban forgó csomag feldolgozása abbamarad. A Netfiltert használva a felhasználói területen futó alkalmazások gyakorlatilag magszinten ellenőrizhetik a hálózati forgalmat.

IP Tables

Az IP Tables olyan alkalmazás, mely a Netfilterhez kapcsolódik, és általa beállíthatók, megtekinthetők és eltávolíthatók a rendszer szűrésére használt szabályok. Azért említem meg itt az IP Tablest, mert az elmélet bizonyítására szolgáló kódunk fejlesztésekor úgy véltük, jobb, ha a felhasználókat a szabályok beállítására az IP Tables program használatával ismertetjük meg, mintha magával az alkalmazással dolgoztatnánk fel a szabályokat. Így az embereknek módjuk nyílik jobban megismerni, mi történik a sorba állított csomaggal.

A mi módszerünk

A Netfilter modulok és az IP Tables szabályfeldolgozó program használatával be tudtuk állítani a szabályokat, melyek elfogták a bejövő UDP-, TCP- és ICMP-csomagokat. A bejövő csomagtól és a küldő géptől függően vagy megengedjük, hogy szokványos módon férjenek hozzá rendszerünkhöz, vagy pedig Windows-gép látszatát keltő válaszokat ötlünk ki az Nmap operációs rendszer ujjlenyomat általi azonosításra használt bejegyzései alapján. Íme ez az ujjlenyomat, amelyet megpróbálunk lemásolni, és egy rövid leírás arról, hogyan értük el a célunkat:

```
TSeq(Class=TD|RI%gcd=1|2|3|4|5|8|A|14|1E|28|
➥5A%SI=<1F4)
T1(DF=Y%W=2017|16D0|860|869F%ACK=S++%Flags
➥=AS%Ops=M|MNWNNT)
T2(Resp=Y%DF=N%W=0%ACK=S%Flags=AR%Ops=)
T3(Resp=Y%DF=Y%W=0%ACK=O%Flags=AR%Ops=)
T4(DF=N%W=0%ACK=S++|O%Flags=R%Ops=)
T5(DF=N%W=0%ACK=S++%Flags=AR%Ops=)
T6(DF=N%W=0%ACK=S++|O%Flags=R%Ops=)
T7(DF=N%W=0%ACK=S++%Flags=AR%Ops=)
PU(DF=N%TOS=0%IPLEN=38%RIPTL=148%RID=E%RIPCK=E
➥%UCK=E%ULEN=134%DAT=E)
```

Az első sorból kiderül, hogy az időponttól függő (TD) TCP-műveletsorozatot kell felépítenünk, vagy pedig egy olyat, mely véletlenszerűen növekszik (RI), és az egyes lépések egyenlők, de nem nagyobbak, mint 0x1F4 (500). Ezt tulajdonképpen elég egyszerű volt elérni, vagy jobban mondva lemásolni. Először is elfogtuk a bejövő csomagot, vettük a TCP-sorozat számát, és létrehoztunk egy véletlenszerűnek tűnő számot 1 és 500 között. Ez megfelelt az ujjlenyomattal szemben támasztott mindkét követelménynek, a véletlenszerű növekedésre és a legnagyobb közös osztóra vonatkozónak is.
Ezután felbontottuk a csomagokra vonatkozó próbákat (T1-T7) és mindegyik esetre létrehoztunk nekik megfelelő eseteket a TCP-kezelőnkben. Ezek mind nagyon egyértelműek voltak, és egyszerűen szükségessé teszik, hogy a gép adott módon válaszoljon a különféle nyitott és zárt kapukra érkező kérésekre. A részletes tesztek és a beállítások alaposabb feldolgozása megtalálható Fjodornak az operációs rendszerek távoli azonosításáról írt tanulmányában.
Ezután megalkottuk az UDP „nem elérhető kapu”-típusú UDP-kérésre adandó válaszunkat. Az Nmap ekkor annyit tesz, hogy a gép egyik zárt kapujára küld egy UDP-csomagot és várja az ICMP „nem elérhető kapu”-csomag formájában érkező választ. A „nem elérhető kapu” ICMP-csomagok egyszerűen annyit mondanak a küldő gépnek, hogy az UDP-csomagot nem sikerült eljuttatni arra a kapura, amelyre szerették volna, mert azon nincs működő UDP-szolgáltatás. Bizonyos hálózatokon ezeket az üzeneteket sohasem küldik vissza, mert az útválasztó eldobja őket. Annak érdekében, hogy alkalmazkodjunk az ujjlenyomathoz, igyekeztünk visszaküldeni azt, amire számítottak.
Végül, gépünk bizonyos kapuiról kis ráadásként Syn-Ack csomagokat küldtünk vissza arra az esetre, ha véletlenül pásztázással megvizsgálnák, nyitva vannak-e ezek a TCP-kapuk. Ezenkívül nem küldtünk vissza semmilyen választ bizonyos UDP-kapuknál, melyeknél azt a látszatot szerettük volna kelteni, hogy nyitva vannak a gépünkön (mint korábban megállapítottuk, csak a zárt UDP-kapuk küldenek vissza „nem elérhető kapu” üzenetet). Amikor gépünk pásztázása véget ér, úgy kell látszódnia, mintha a 135. és 139. TCP-kapu, valamint a 135., 137., és 138. UDP-kapu nyitva lenne. Ha gépünket az ujjlenyomata alapján megpróbálják azonosítani, bizonyosan megfelelünk a fenti ujjlenyomatnak és „Windows NT4, Windows 95, Windows 98” gépnek látszunk.
Végül az elmélet bizonyítására használt kód pontosan ezt jelenti: rövid programrészlet, melynek célja, egy elképzelés bizonyítása. Kíméljük meg magunkat a gondoktól, és ne futtassuk létfontosságú eszközökön. Próbáljuk ki, tanuljunk belőle, módosítsuk, hasznosítsuk, de ne támaszkodjunk rá! Igyekeztem biztonságosan és jól olvashatóan (ez vitatható) megalkotni a kódot, de semmit nem ígérhetek.

Köszönetnyilvánítás

Hálás köszönetem ***Rex Warren***-nek a kemény munkájáért, és amiért ennek a tanulmánynak és a mintakódnak az elkészítésében segített, Fjodornak azért, hogy megengedte nehéz munkája gyümölcsének felhasználását, és hogy ilyen nagyszerű biztonsági eszközt készített. Köszönet ***Dan Kurc***-nak kódom átolvasásáért és azért, hogy szüleményemet csúnyának nevezte (hé, ez az első C-ben készült programom), és ***Sir Dystic***-nek a C nyelv használatához nyújtott támogatását, valamint hála ***Courtnee***-nek.

Rob Beck jelenleg biztonsági rendszerek tervezőjeként dolgozik az @stake Security Consulting szolgálatnál, szakterülete a Windows operációs rendszerekbe, illetve alkalmazásokba való behatolás felmérése, és a biztonságos felépítés megtervezése.

Kapcsolódó címek

Fjodor tanulmánya az operációs rendszerek távoli felismeréséről ➲ http://www.insecure.org/nmap/nmap-fingerprinting-article.html (*1. kép*) Rendkívül tanulságosnak találtam ezt a tanulmányt az operációs rendszerek ujjlenyomat általi azonosítása témakörében. Netfilter magmodulok ➲ http://netfilter.samba.org (*2. kép*)