ÁROP – 2.2.21 Tudásalapú közszolgálati előmenetel

ISBN 978-615-5491-59-7

Biztonsági tesztelés a gyakorlatban

Tihanyi Norbert, Vargha Gergely, Frész Ferenc

Nemzeti Közszolgálati Egyetem

Budapest, 2014

Tartalomjegyzék

1. Bevezető a tananyaghoz

Információbiztonsági problémák, különböző kockázati besorolású incidensek nap, mint nap előfordulnak az összetett informatikai rendszerekben, melyek azonosítására, kezelésére, kiküszöbölésére, a keletkezett károk elhárítására, csillapítására jelentős anyagi és emberi erőforrások kerülnek felhasználásra. E költségek egy része kiküszöbölhetetlen és az informatikai rendszerek működésének sajátosságai alapján mindig is jelen lesznek a rendszerekben, azonban van a teljes információbiztonságra fordított költségeknek egy igen jelentős része, mely feleslegesen terheli a szervezetek, fejlesztési projektek költségvetését. Ezek tipikusan olyan, az informatikai rendszerekbe beágyazott alrendszerek tervezési és implementálási hiányosságaira vezethetőek vissza, melyek jó része a megtervezett, időben végrehajtott és lépésről lépésre lekövetett biztonsági tesztelésekkel megelőzhetőek lehetnének.

A felszínre bukkanó problémák utólagos kezelése a költségvonzat tekintetében akár több nagyságrendi eltérést is mutathat a megalapozott tervezés, fejlesztés és implementálás erőforrásigényeihez képest, de ami talán még ennél is fontosabb az az, hogy az éles működés alatt véletlenszerűen bekövetkező incidensek, működési zavarok az alapvető elvárásként megfogalmazható informatikai rendszer rendelkezésre állásának folytonosságát megszakíthatják, a rendszer működését megbéníthatják, akár a teljes rendszer vonatkozásában.

Látható tehát, hogy a biztonsági tesztelések elhagyása, vagy felületes elvégzése súlyos következményekkel járhat adott szervezet, fejlesztési projekt vagy akár termék vonatkozásában, mind költség oldalon mind az információ biztonság tekintetében. Felmerül a kérdés, hogy akkor általában miért nem fordítanak kellő erőforrást a biztonsági tesztelések elvégzésére?

A tananyag kísérletet tesz a kérdésre adott válasz kifejtésére, azonban a válasz megfogalmazásakor fontos figyelembe venni, hogy a biztonsági tesztelések rendszeres és tervezett elvégzése önmagában nem oldja meg az informatikai rendszerek információ biztonsági problémáit. Az azonban kijelenthető, hogy a biztonsági tesztelések rendszerének megfelelő kialakítása és fenntartása hozzásegítheti mind a rendszereket fenntartó szervezeteket, mind a rendszerek üzemeltetésében résztvevő szakembereket, továbbá a felelős

döntéshozókat abban, hogy adott rendszer biztonsági szintje, rendelkezésre állása külön, jelentősebb pénzügyi erőforrás bevonása nélkül meghatározó mértékben javulhasson.

A tananyag célja többrétű. Egyrészt megpróbálja a fent nevezett probléma eredetét meghatározni, annak érdekében, hogy hatékonyan segítse ezzel a képzésben résztvevő szakemberek biztonsági tesztelésekkel kapcsolatos feladatainak szervezeti szintű beazonosítását. Másrészt a tananyag kiemelten foglalkozik a szervezeteknél elvégzett biztonsági tesztelések értelmezésével és felhasználásával annak érdekében, hogy támogassa a felelős biztonsági vezetőket a tárgykörrel kapcsolatos döntéshozatali folyamatokban. A tapasztalatok az mutatják, hogy a biztonsági vezetők a leghatékonyabban a biztonsági tesztelések közül is a sérülékenység vizsgálatok elrendelésével tudják elősegíteni az adott szervezet elvárt információbiztonsági szintjének elérését, így a tananyag gyakorlati része erre a témakörre helyezi a hangsúlyt, legyen szó akár sérülékenység vizsgálat megrendeléséről, a vizsgálatok lefolytatásáról, vagy akár a vizsgálatok eredménye alapján az intézkedési tervek végrehajtásáról.

2. Biztonsági tesztelésről általában

2.1. Bevezetés

A mind összetettebb rendszerek, a folyamatosan megkurtított költségvetések, az egyre szűkebb határidők, a bürokratikus pályázati lehetőségek a legtöbb esetben már az indulás előtt lehetetlen helyzet elé állítják az informatikai projekteket. Ennek az általános trendnek az egyik hozadéka, hogy a fejlesztési projektek során a biztonságra fordított erőforrásokat kispórolják a projektből, hiszen ez látszólag nem befolyásolja a majdan elkészülő rendszer rendelkezésre állását. Ez a téves, kártékony és felelőtlen döntéshozatali eljárás azonban – ahogy számtalan gyakorlati példa is igazolja – gyenge vagy elégtelen biztonsági szintű végtermékeket eredményez. Mindezt összevetve azzal az alapvetéssel, hogy egy rendszer biztonsági szintje megegyezik a leggyengébb elemének biztonsági szintjével, továbbgondolható, hogy milyen biztonsági kockázatokat hordoz magában az, ha egy ilyen terméket integrálnak egy szervezet informatikai rendszerébe.

Ilyen környezetben a tesztelések, és azon belül is a biztonsági tesztelések rendszerének kialakítása és folyamatos elvégzése, kiemelt jelentőségű feladata a szervezet információbiztonságért felelős döntéshozóinak, munkatársainak.

2.2. Fejlesztési problémák

Különböző informatikai fejlesztési projektekben a technológiai robbanás, a soha nem látott mértékű fejlesztések, az új irányvonalak, a piac diktálta eszeveszett tempó hatására óhatatlanul felborul a minőség, költség, határidő hármasának egyensúlya, legalább valamely szempontrendszer sérülni fog és megjelenik az úgynevezett szoftverkrízis jelensége, mely a projektek sikertelenségéhez vezet az alábbi tünetek kíséretében:

- a fejlesztés a tervezettnél drágábban készül el,
- a fejlesztés a tervezetnél hosszabb idő alatt készül el,
- a fejlesztés nem az igényeknek megfelelően készül el,
- a fejlesztés eredményterméke rossz minőségű / rossz hatásfokú / nehezen karbantartható lesz,
- a fejlesztés eredménytermékének használata anyagi / környezeti / egészségügyi kárhoz vezet,
- a fejlesztés eredményterméke átadásra sem kerül

Fent nevezett problémák közül a biztonsági tesztelés az információbiztonsági hiányosságok kiküszöbölésében, illetve a károkozás megelőzésében tudja hatékonyan támogatni a fejlesztést. A többi esetben a problémák megelőzését leginkább a helyesen megválasztott és következetesen végrehajtott fejlesztési módszertan (pl.: V-modell, prototípus modell, RAD alkalmazás fejlesztés, agilis szoftverfejlesztés stb.) segítheti elő.

2.3. Biztonsági tesztelés a fejlesztés különböző életciklusaiban

A biztonsági tesztelések célja, hogy a fejlesztés különböző életciklusaiban felderíthető információbiztonsági hiányosságok azonosításra és mielőbbi javításra kerülhessenek. Az informatikai fejlesztések életciklusait alapvetően 6 jól elkülöníthető csoportra oszthatjuk, melyek az alábbiak:

1. *Tervezési szakasz:* a fejleszteni kívánt alkalmazás a különböző előzetes specifikációk alapján megtervezésre kerül, amivel párhuzamosan az információbiztonság szemszögéből a biztonsági intézkedések azonosítása és specifikációja is megtörténik. Ebben a fázisban a biztonsági tesztelés a dokumentációkban meghatározott folyamatok, követelmények információbiztonsági vonatkozású megfelelőségét hivatott biztosítani.
2. *Fejlesztési szakasz:* az elfogadott tervek alapján az informatikai rendszer fejlesztése megtörténik. A fejlesztési folyamat mérföldköveinél a fejlesztés típusától függően érdemes lehet a biztonsági teszteléseket elvégezni az elkészült rendszerelem vonatkozásában, hogy a fejlesztés során a rendszerbe kódolt hibák a lehető legkorábbi fázisban azonosításra és javításra kerülhessenek.
3. *Tesztelési szakasz:* az elkészült alkalmazás tesztelése, esetleges hibák felderítése, hibajavítások elvégzése, felkészülés az éles üzemre való átállásra. Ebben a szakaszban a teljes rendszer funkcionális tesztelése mellett a biztonsági tesztelések is végrehajtásra kerülnek, de még nem az éles környezetben.
4. *Implementációs szakasz:* az elkészült alkalmazás beüzemelése, valamit az üzembe állításhoz köthető biztonsági intézkedések (szervezési és technikai intézkedések) megvalósítása. A biztonsági teszteléseket fontos elvégezni az éles rendszeren is, mivel számos esetben az éles- és a tesztrendszer biztonsági szempontból releváns paraméterekben különbözhet.

5. *Fenntartási szakasz:* a fejlesztés eredményeként előállt alkalmazás folyamatos üzemeltetése mellett szükséges az előre meghatározott biztonsági intézkedések, tesztesetek napi szintű alkalmazása, visszacsatolása, a rendszer ellenőrzése. A biztonsági tesztelések tehát ebben a fázisban annak teljes élettartama alatt ciklikusan elvégzésre kerülnek.
6. *Nyugdíjazás:* a rendszer leállításra kerül, mely során az előzetes terveknek megfelelően gondoskodni kell a rendszerben tárolt adatok megsemmisítéséről, vagy éppen mentéséről, esetleg új rendszerbe való migráció esetén adatkonverzióról és/vagy adattisztításról. A biztonsági tesztelés képes lehet a nyugdíjazás során meghatározott folyamtok elvégzésének megfelelőségét ellenőrizni.

2.4. Hibajavítás költségvonzata

A különböző fázisokban a biztonsági tesztelések során feltárt hibák javításának költségigénye jelentős eltérést mutathat. Általánosságban kijelenthető, hogy minél korábban azonosításra kerül egy adott hiba, a javításának költségvonzata szignifikánsan kisebb lesz. Így - ahogy sok felmérés és tanulmány is igazolja - a biztonsági tesztelések által a megfelelő szakaszban feltárt és javított hibákra fordított erőforrás, mely az egyedi biztonsági tesztelések tervezéséből, elvégzéséből és a hibajavítás folyamatából tevődik össze, a későbbiek során sokszorosan megtérül.

Ezt azonban nem feltétlenül lehet számszerűsíteni, hiszen egy meg nem történt incidens, adatszivárgás értékét a legritkább esetben szokták nyereségként elkönyvelni, míg egy súlyos információbiztonsági incidens egyértelmű és számszerűsített veszteségként jelentkezik. Ez az oka többek között annak, hogy a döntéshozók egész addig nem feltétlenül támogatják a biztonsági tesztelések rendszerét, míg nem érintettek a problémában személyesen, és nem értik meg egyértelműen a tevékenység fontosságát, súlyát. Ez az a szemlélet, melynek megváltoztatásában leginkább a biztonsági tudatosságot erősítő képzésekkel lehet eredményeket elérni.

Az Amerikai Szabványügyi Hivatal (NIST) által publikált tanulmány alapján a hibajavítások költségei a különböző fejlesztési fázisokban az alábbiak szerint alakulnak:

Életciklus megnevezése	Információbiztonság érvényesítésének költségvonzata
Tervezési szakasz	alapértelmezett, 1x költségigény
Fejlesztési szakasz	6.5x költségvonzat
Tesztelési szakasz	15x költségvonzat
Fenntartási szakasz	100x költségvonzat

Az összesítésből egyértelműen látszik, hogy az információbiztonság szempontrendszerének alapjait, ezen belül a biztonsági tesztelések rendszerét elengedhetetlen a tervezési szakaszban rögzíteni annak érdekében, hogy a későbbi problémák megelőzhetőek, minimalizálhatóak legyenek, ugyanakkor egy hiba feltárása és kijavítása még egy későbbi projekt fázisban is megtérülhet ahhoz a veszteséghez képest, amit egy rendszer kompromittáció okozni képes.

2.5. Sérülékenységek kialakulása

A biztonsági tesztelések eredményei alapján a leggyakrabban előforduló sérülékenységek statisztikáiból beazonosítható az a három terület, melyekre érdemes mind a biztonsági tesztelés, mind a fejlesztés folyamán kiemelt figyelmet fordítani: *infrastruktúra szint, operációs rendszer szint* és *alkalmazás szint*. Amennyiben alábontjuk ezeket a szinteket és ismerjük az adott területek leggyakoribb hiányosságait, hatékonyabb tervezési és tesztelési folyamatokat tudunk összeállítani, melyek során az adott hiányosság, sérülékenység ellenőrzésre, tesztelésre kerülhet.

A következőkben bemutatjuk a 3 érintett szint sérülékenységek vonatkozásában leggyakrabban érintett összetevőit:

2.5.1. Infrastruktúra szint

A jelenlegi komplex informatikai rendszerek esetén, infrastruktúra szinten számtalan olyan pontot azonosíthatunk, melyek jelentős rést üthetnek a rendszer védelmén, adott esetben kiszolgáltatva az informatikai rendszert egy teljes, rendszerszintű kompromittációnak. Az infrastruktúra szint biztonságának alapjait a többi területhez hasonlóan a tervezési szakaszban szükséges lefektetni, rögzíteni. Az alábbi felsorolás a legfontosabb területeket mutatja be, melyek ellenőrzése, illetve adott esetben biztonsági tesztelése kiemelten fontos:

2.5.1.1. Adatszivárgás megakadályozása

- partnerek és beszállítók biztonsági minősítése
- gyári beállítások módosítása
- hálózati megosztások feltérképezhetőségének tiltása
- a teszt, a fejlesztési és az éles környezetek elkülönítése
- adatszivárgást elősegítő konfigurációk megszüntetése

2.5.1.2. Hálózati hozzáférésvédelem kialakítása

- külső és belső hálózati határvédelem kialakítása
- terheléselosztó rendszerek használata
- adminisztratív területek elérhetőségének korlátozása
- adminisztratív, felhasználói és szerver oldali virtuális helyi hálózatok kialakítása
- Egyirányból érkező viharszerű lekérdezések csillapítása

2.5.1.3. További szükséges intézkedések

- minősített eszközök használata
- megfelelő minőségű titkosított kommunikációs csatornák használata
- szükségtelen protokollok tiltása
- nem rejtjelezett protokollok használatának mellőzése
- felesleges szolgáltatások leállítása
- megfelelő autentikáció és autorizáció használata
- megfelelő minőségű jelszavak kikényszerítése
- log menedszment használata
- nem hitelesített (anonim) elérések tiltása
- ciklikus sérülékenység vizsgálatok végzése, eredményeinek felhasználása

2.5.2. Operációs rendszer szint

Az operációs rendszer kiválasztása tekintetében elképzelhető, hogy alapvetően nem az információbiztonsági szempontok fogják eldönteni az választott rendszer típusát, azonban amennyiben előbbi eldöntésre kerül, az alábbi intézkedések figyelembe vétele szintén megkerülhetetlen részét képezi a tudatos információbiztonsági felkészülésnek és a biztonsági teszteléseknek:

- operációs rendszer és komponenseinek frissen tartása (patch mmenedzsment)
- munkaállomások és szerverek beállításainak időszakos felülvizsgálata, megerősítése
- hozzáférési azonosítók titkosított formában való tárolása
- megfelelő autentikáció és autorizáció használata
- megfelelő minőségű jelszavak kikényszerítése
- nem hitelesített (anonim) bejelentkezések tiltása
- adatvédelmi technikák bevezetése (PGP, DRM, stb.)
- backup management kialakítása
- beépített felhasználók tiltása, egyedi felhasználónevek alkalmazása
- vírus- és kártevőprogram elleni védelem kialakítása
- gyári beállítások módosítása
- szükségtelen protokollok tiltása
- felesleges szolgáltatások leállítása
- adminisztratív felületek elérhetőségének korlátozása
- ciklikus sérülékenység vizsgálatok végzése, eredményeinek felhasználása

2.5.3. Alkalmazás szint

Alkalmazás szinten az információbiztonságot szervesen befolyásoló területek nagyon szerteágazóak. A nem megfelelő tervezés, fejlesztés vagy implementálás ebben az esetben is könnyedén teljes, rendszerszintű kompromittációhoz vezethet, mely elhárításának költségvonzata felesleges, megelőzhető, ugyanakkor jelentős terhet róhat az informatikai projektre. Fontos az alábbi, általános szempontok megkövetelése és alkalmazása:

2.5.3.1. Többrétegű architektúra kialakítása

- megjelenítési réteg,
- távoli elérést kiszolgáló réteg,
- alkalmazás réteg,
- adatbázis réteg.

2.5.3.2. A szoftveres környezetnek megfelelő bemeneti paraméter ellenőrzése (input validation)

- engedélyezett karakterek listája
- adattípus ellenőrzés

- formátum vagy kép ellenőrzés
- fájl meglevőségének ellenőrzése
- hash összesítő ellenőrzés
- adatkorlát ellenőrzés
- mennyiségi ellenőrzés
- számossági ellenőrzés
- tartomány ellenőrzés
- ellenőrző szám vagy számsor használata
- logikai hiba ellenőrzés
- jelenlét ellenőrzés
- konzisztencia ellenőrzés
- összesítő ellenőrzés
- keresztplatformos konzisztencia ellenőrzés
- hivatkozás integritás
- helyesírás ellenőrzés
- egyediség ellenőrzés

2.5.3.3. További szükséges intézkedések

- webes könyvtár és fájl indexálhatóság tiltása
- tartalomszűrő rendszer használata
- hozzáférési azonosítók titkosított formában való tárolása
- megfelelő autentikáció és autorizáció használata
- megfelelő minőségű jelszavak kikényszerítése
- nem hitelesített (anonim) bejelentkezések tiltása
- kriptográfiai technikák alkalmazása
- backup management kialakítása
- beépített felhasználók tiltása, egyedi felhasználónevek alkalmazása
- gyári beállítások módosítása
- adminisztratív felületek elérhetőségének korlátozása
- log management használata
- ciklikus sérülékenység vizsgálatok végzése, eredményeinek felhasználása

2.6. Biztonsági tesztelés hatóköre

Fontos különbséget tenni a tesztelés és a biztonsági tesztelés között. Míg maga az általános tesztelési folyamat a CIA modellt (bizalmasság; sértetlenség; rendelkezésre állás) alapul véve inkább a rendelkezésre állásra fókuszál, addig a biztonsági tesztelések elsősorban a bizalmasság és a sértetlenség feltételrendszerének ellenőrzését segítik elő, ugyanakkor nehéz éles és egyértelmű határt húzni a két tesztfolyamat között, mert például létfontosságú rendszerek esetén maga a rendelkezésre állás is kiemelt biztonsági kérdés, így a biztonsági teszteknek erre is ki kell terjedniük. Összefoglalva, biztonsági tesztelésnek nevezhetünk minden olyan tevékenységet, mely adott informatikai rendszer információbiztonsági megfelelőségét hivatott ellenőrizni.

3. Biztonsági tesztelés megjelenési formái

3.1. Bevezetés

Ahogy az előző fejezetben láthattuk a biztonsági teszteléseket különböző fejlesztési fázisokhoz lehet kötni, de csoportosíthatjuk a tesztelés tárgya szerint is. Jelen fejezet a biztonsági tesztelés területeit a tesztelendő elem típusa alapján csoportosítva mutatja be.

3.2. Dokumentáció biztonsági tesztelés

Jellemzően még a különböző informatikai fejlesztések előkészítő fázisaiban előálló felhasználói, fejlesztői és információbiztonsági specifikációkat és egyéb dokumentumokat ajánlott információbiztonsági szempontból felülvizsgálni, illetve az elvárt biztonsági követelményekkel egybevetni. Fontos megjegyezni, hogy ajánlott az információbiztonsággal kapcsolatos kérdéseket, szabályozásokat külön információbiztonsági specifikációban rögzíteni.

3.3. Fizikai biztonsági tesztelés

A fizikai biztonsági eseményekről elmondható, hogy általában az alacsony valószínűségű események közé sorolhatóak, azonban az esemény bekövetkeztekor a kárérték aránya kifejezetten magas lehet. Mindehhez párosul, hogy a szimulációjuk elég nehézkes és az is magas költségekkel jár. A biztonsági tesztelések alapvető feladata ezen a területen is a megelőzés, annak érdekében, hogy káresemény ne következhessen be.

Az informatikai fizikai biztonság témakörének egy lehetséges csoportosítása az alábbi, melyet figyelembe vehetünk a biztonsági tesztek, ellenőrzések tervezésekor:

Fizikai biztonság: biztonsági tesztelések során leginkább a fizikai biztonságot megvalósító eszközök kerülnek tesztelésre, ilyenek tipikusan a zárak, lakatok, biztonsági ajtók. A fizikai biztonság érinti az alábbi területeket:

- passzív tűzvédelem (pl.: tűzgátló válaszfalak, tűzgátló ajtók)
- páratartalom szabályozás
- vízbetörés elleni védelem
- betörés/illetéktelen hozzáférés elleni védelem

Technikai biztonság: biztonsági tesztelések során leginkább a biztonságos energiaellátást biztosító berendezések (pl.: dízel generátorok, szünetmentes áramellátást biztosító berendezések), illetve a tűz és füstjelző berendezések tesztelésére szokott sor kerülni:

- biztonságos energiaellátás és elosztás
- géptermi és technikai területek hűtése
- automatikus üzemfelügyelet
- aktív tűzvédelem (pl.: tűzjelzők, automatikus oltóberendezések)
- kábelmenedzsment
- kommunikációs hálózat fizikai biztonsága

Logikai biztonság: érdemes tesztelni, hogy külső személy, vagy belső felhasználó milyen jellegű biztonsági zónákat képes elérni, milyen a környezet dokumentáltsága, a biztonsági események rögzítése, illetve például papír alapú dokumentumok, adathordozók, milyen csatornákon juthatnak ki a rendszerből:

- hozzáférés szabályozás
- integrált elektronikus védelmi (behatolás, beléptető és CCTV) rendszerek
- védelmi rendszerek naplóállományainak elemzése
- fizikai és IT környezet változáskezelés
- karbantartás és hibaelhárítás menedzsment

3.4. Hardver biztonsági tesztelés

Az informatikai rendszerek infrastrukturális alapjait hardver eszközök biztosítják, így az informatikai rendszerek döntő többségében a hardver eszközök beszerzése megkerülhetetlen. Új hardver megrendelése esetén az eszköz a legtöbb esetben gyári beállításokkal érkezik, a szállító nem állítja be a szervezet specifikus paramétereket, a megrendelő azonban sok esetben azt várja, hogy kulcsrakész rendszert kap. Ebben az általánosnak mondható elvárás rendszerben a megrendelő és a szállító elvárásai nem találkoznak. Ebben az esetben a biztonsági teszteléseknek mindenképp ki kell terjedniük az adott hardver elemek biztonsági tesztelésére a gyári beállításokkal kapcsolatban.

Előfordul olyan eset is, amikor a szállító rögzíti, hogy a megrendelő nem módosíthatja adott hardverelem beálltásait, mert például elvész a garancia. Ebben az esetben a biztonsági teszteket sem tudjuk elvégezni a klasszikus értelemben, de a szerződő feleknek minden, a hardver biztonságával kapcsolatos kérdést szabályoznia kell az együttműködési

megállapodásokban, a szolgáltatói szerződésekben, vagy éppen a üzemeltetési lánc definiálásakor.

Sajnos a példák azt mutatják, hogy a biztonsági beállítások, módosítások adott esetben elvégzésre kerülnek az informatikai eszközben még az eszköz beérkezését megelőzően.

3.5. Szoftver biztonsági tesztelés

Szoftver biztonsági tesztelésén általában a szoftver forráskódjának célirányos átvizsgálását értjük, mely megelőző tevékenységként számos előnyt biztosít:

- csökkenti a szoftverfejlesztés és karbantartás költségeit
- segít a forráskód értelmezésében, karbantarthatóságában
- információval szolgál a termék minőségéről
- a jobb minőségű kód kevesebb tesztelést igényel
- felfedezhetőek, majd javíthatóak a biztonsági rések az éles indulás előtt

Alapvetően beszélhetünk statikus és dinamikus tesztelésről, melyekkel más-más típusú hibák fedezhetőek fel.

Statikus tesztelés folyamán a szoftver forráskódja kerül átvizsgálásra, melybe beletartozik a dokumentáció felülvizsgálata is. A statikus tesztelésen belül is megkülönböztethetünk úgynevezett *felülvizsgálatot*, mely a kód illetve a dokumentáció elemzésének manuális folyamatát jelenti, illetve *statikus elemzést,* amely során a kód illetve a dokumentáció elemzése automatikus eszközök felhasználásával történik.

A statikus tesztelési eljárás nem követeli meg a tesztelendő rendszer futtatását, sőt még a teljes forráskód állomány együttes megléte sem alapkövetelmény. A statikus tesztelési technikák előnye, hogy már korai fejlesztési szakaszban alkalmazhatóak, akkor, amikor még nem áll rendelkezésre futtatható verzió, így a tesztelések során megállapított hibákat hamarabb vissza lehet táplálni a rendszerbe, csökkentve a későbbi fejlesztési költségeket illetve növelve a rendszerek biztonságát a biztonsági rések feltárásával és javításával.

Dinamikus tesztelés esetén a szoftver futásidőben kerül tesztelésre, vagyis leghamarabb akkor végezhető el, amikor a teljes rendszer összeállt és a forráskódok lefordításra kerültek. A legelterjedtebb dinamikus tesztelési módszertanok a specifikáció alapú, a struktúra alapú és a gyakorlat alapú tesztelési technikák.

Adott funkciót megvalósító szoftver termék megrendelése esetén általában 2 lehetőség adódik:

- egyedi fejlesztés
- dobozos termék

Egyedi fejlesztés esetén a biztonsági tesztek elvégzése vagy a fejlesztői oldalon történik meg a fejlesztő csapat mérnökeinek bevonásával, vagy független, kódaudit tesztelésre specializálódott szolgáltató bevonásával. Ezek eredményeiről a megrendelőnek is célszerű tudnia. Később, a termék átadását követően lehet kérni a fejlesztőtől független biztonsági bevizsgálást a termékre vonatkozóan, ezt nevezzük általában etikus hacking eljárásnak, mely során a sérülékenység vizsgálat befejezését követően, annak eredményeként előállt és a szerződésben meghatározott kockázati szintet elérő hiányosságokat a megrendelőnek javítania kell. Ezen felül elengedhetetlen a szerződésben a forráskód tulajdonjogának kérdését is rögzíteni.

Dobozos termék esetén előfordulhat, hogy nem lesz lehetőség kódaudit elvégzésére, mivel a fejlesztő fenntarthatja a jogot a zárt forráskódra. Ebben az esetben is elvégezhető az audit black-box módszerrel, ahol a forráskód, illetve dokumentáció nem áll rendelkezésére a tesztelőknek.

3.6. Hálózat biztonsági tesztelés

A hálózati biztonsági tesztek célja megmutatni, hogy a hálózat részeként funkcionáló eszközről, vagy idegen eszköz belső hálózathoz való csatlakoztatása esetén milyen mértékű rendszerkompromittáció érhető el. A hálózati tesztek kiterjednek mind a vezetékes, mind a vezeték nélküli, valamint a telekommunikációs hálózatokra, beleértve a hálózatban található aktív eszközöket is.

A hálózati vizsgálatok esetében a sérülékenységek döntő többségét az alábbi problémák idézik elő, így ezek biztonsági tesztelése minden esetben javasolt:

- *alapértelmezett gyári beállítások:* hálózati eszközök tekintetében gyakran előfordul, hogy a gyári beállítások nem kerülnek megváltoztatásra, így az adott eszközök elérhetőek, átkonfigurálhatóak
- *hálózati megosztások elérése, feltérképezhetősége:* akár jogosultság nélkül, akár olyan felhasználói jogosultsággal, mely felhasználói hozzáférés nem indokolt (számos esetben webalkalmazások home könyvtárai is elérhetőek, módosíthatóak)

- *nem megfelelő hálózati végponti védelem:* a végpontokra csatlakoztatott eszközök hálózati kommunikációját az informatikai rendszer nem akadályozza meg, például DHCP szolgáltatás kiosztja az IP címet és az egyéb szükséges beállítások is automatikusan konfigurálásra kerülnek (DNS, Default Gateway, stb.).
- *nem kellően kialakított hálózati szegmentáció:* alapvető probléma szokott lenni, hogy a felhasználói, az adminisztratív és a szerver szegmensek átjárhatóak. A felhasználók által használt szolgáltatások elérése természetesen fontos, de ezeken kívül máshoz nem szabad hozzáférést biztosítani, mert az felesleges biztonsági kockázat hordoz magában
- *közbeékelődéses (Man in the Middle) támadás lehetősége: Két fél közötti kommunikáció kompromittációja olyan* módon, hogy a támadó a kommunikációs csatornát eltérítve mindkét fél számára a másik félnek adja ki magát. A támadás során tetszőleges belső felhasználó kommunikációja lehallgathatóvá válhat, így a hálózat biztonsági tesztelésekor szükséges ellenőrizni, hogy a hálózat tartalmaz-e MAC Flooding, IP cím, MAC cím és DHCP szerverhamisítás elleni védelmet.
- *titkosítás nélküli protokollok használata:* minden esetben javasolt a titkosítatlan protokollok titkosított párját alkalmazni (FTP helyett SFTP, HTTP helyett HTTPS, stb)
- *felesleges szolgáltatások:* az új technológiák, eszközök rendszerbe való integrálása elviszi a fókuszt a régebbi rendszerelemekről, így azok használata fokozatosan megszűnik, de a szolgáltatás sok esetben benne marad a rendszerben, ami az idő múlásával egyre több sérülékenységet tartalmaz majd. Ugyanez a probléma felvetődik új rendszerelemek integrálása után is, amikor a gyári beállítású, de nem használt szolgáltatások elérhetőek. Ezeket a szolgáltatásokat le kell állítani, feleslegesen növelik a biztonsági kockázatot.
- *demilitarizált zóna átjárhatósága:* a biztonsági tesztelések folyamán fontos ellenőrizni, hogy a külső hálózat és a védett hálózat védelmi szintje között elhelyezkedő DMZ kellő védelmet nyújt-e a külső hálózat kompromittálódása esetén
- *WiFi eszközök biztonsági beállításai:* a WiFi eszközök gyári beállítása nem, vagy csak nagyon alacsony szintű biztonsági beállításokat tartalmaz, melyek megváltoztatása elengedhetetlen (SSID megváltoztatása, SSID szórásának kikapcsolása, gyári jelszavak megváltoztatása, lehetőség esetén MAC szűrés bekapcsolása, DHCP kikapcsolása)

3.7. Rendszer biztonsági tesztelés

Egy teljes informatikai rendszer biztonsági tesztelésén a fejezetben felsorolt tesztek együttes alkalmazását érthetjük kiegészülve az alábbiakkal, melyek a különböző rendszerelemek együttes viselkedésére lehetnek hatással:

- *Social engeneering:* az emberi hiszékenységen alapuló tesztek képesek rámutatni az adott szervezet munkatársainak információbiztonsági tudatossági szintjére, mely az egyik legnagyobb biztonsági kockázatot magában foglaló faktor (humán faktor). Az információbiztonsági tudatosság megerősítését csak a többszintű, célzott információbiztonsági tudatossági képzések elvégzésével lehet elérni, melyek az alábbi célcsoportokat érintik:
    - felhasználók
    - üzemeltetők
    - fejlesztők
    - döntéshozók
- *IT biztonsági szabályzók:* fontos, hogy a különböző rendszerszintű szabályzók a gyakorlatban alkalmazhatóak legyenek, így ezek tesztelése is részét képezik a teljes rendszer biztonsági tesztjének:
    - BCP (Business Continuity Plans - Üzletmenet-folytonossági terv)
    - DRP (Disaster Recovery Plans - Informatikai katasztrófa elhárítási terv)
    - SLA (Service Level Agreement - **S**zolgáltatási szint szerződés)

4. Sérülékenység faktorok

4.1. Bevezetés

Ahhoz, hogy egy sérülékenység vizsgálati jelentést megfelelően tudjunk értelmezni, elengedhetetlen a sérülékenységi faktorok bemutatása, azok áttanulmányozása. Mielőtt áttekintjük a sérülékenység vizsgálati módszertant, áttekintjük az alapvető sérülékenységi faktor osztályokat.

Egy vizsgálati jelentés feldolgozásakor az egyik legfontosabb dolog, ha valamilyen ismeretlen sérülékenységgel találkozunk, hogy sikeresen tudjuk behatárolni, hogy a hiba melyik nagy sérülékenységi osztályba tartozik. Ha ezt a képességet megfelelően elsajátítjuk, akkor egy új sérülékenység megjelenését könnyen tudjuk kategorizálni, amely később segítséget nyújt ahhoz, hogy a hibák javítását megfelelően tudjuk menedzselni.

4.2. Sérülékenység faktorok meghatározása

Alapvetően a sérülékenységi faktorok közül 8 nagyobb csoportot különböztetünk meg. Ezen csoportokba az előforduló sérülékenységek összes megjelenése besorolható. A sérülékenység típusokat több, illetve kevesebb részre is fel lehet bontani, kizárólag arra kell figyelni, hogy minden egyes sérülékenység le legyen fedve. A Nemzeti Biztonsági Felügyelet által is használt sérülékenységi faktorok osztályozása a következő:

4.2.1. Information Disclosure (információ kitakarás)

Minden olyan típusú hiba ide tartozik, amely valamilyen többletinformációt szolgáltat egy támadónak, amelyet felhasználhat egy későbbi támadáshoz. Általában az ilyen típusú hibák önmagukban nem kihasználhatóak. Kihasználásukhoz, valamilyen egyéb másik sérülékenységet felhasználva kerülhet sor. Jó példa erre az operációs rendszerek, illetve szoftverek verziószámának, típusának kiszivárgása. Ha egy támadó a szerver válaszaiból pontosan meg tudja határozni, hogy milyen típusú, illetve verziójú operációs rendszer fut a szerveren, akkor célzottan tud különböző ismert sérülékenységeket kihasználni az operációs rendszer ellen. Információ kitakarás kategóriába tartozik egy web alkalmazás könyvtárstruktúrájának listázhatósága is. Amennyiben feltérképezhető, hogy pontosan milyen fájlok, programok találhatóak a szerveren, akkor azokat elemezve, újabb sérülékenységek tárhatóak fel. Például ha egy honlapon a /cgi-bin/upload/ könyvár listázása során találunk egy

upload_file_FX345.php nevű file-t amellyel a webhelyre bárki feltölthet adatokat, akkor ezzel a teljes szerver kompromittálhatóvá válhat. Amennyiben a könyvtár listázhatóság nem lett volna kivitelezhető, akkor a támadó nem talált volna rá az upload_file_FX345.php fájlra sem. Az ilyen és ehhez hasonló többletinformációk kiszivárgásának elhárításával számos támadást lehet megelőzni.

*4.2.2. Brute Forcing (*Tömeges Viharszerű lekérdezések, „nyers erő" alkalmazása)

Ezen sérülékenység faktor nagyon sok helyen fordulhat elő, és rengeteg támadási forma épülhet a kihasználására. Bármilyen alkalmazásról vagy protokollról legyen is szó, közös tulajdonsága a sérülékenységi faktornak, hogy nagy mennyiségű kérést lehet intézni egy adott szolgáltatásra vonatkozóan, anélkül, hogy bármilyen ellenintézkedés történne. Az egyik legkézenfekvőbb és legelterjedtebb sérülékenység a különböző login felületeknél tapasztalható. Több próbálkozás után a rendszer nem lassítja, korlátozza a próbálkozások számát. Egy rosszindulatú felhasználó így rengeteg felhasználónév/jelszó párost tud kipróbálni. Egy másik kézen fekvő probléma, ha egy webes szolgáltatás teljes könyvtár struktúrája tömeges lekérdezéssel végigpróbálgatható. Előfordulhat, hogy olyan konfigurációs állományokhoz fér hozzá a támadó, amelyet felhasználva a teljes rendszer kompromittálhatóvá válik. Összességében, minden olyan tömegesen végrehajtott lekérdezés ebbe a kategóriába tartozik, amelyeket a rendszer ellen fordíthat egy támadó.

4.2.3. Phishing (adathalászat)

Az ebbe a sérülékenységi faktorba tartozó hibák közös tulajdonsága, hogy felhasználóktól próbálnak adatokat szerezni illetéktelen módon. Ennek egyik jó példája az e-mailen történő adatszerzés, vagy a különböző honlapok utánzásával végrehajtott adatlopás. Az egyik legelterjedtebb példa, hogy rosszindulatú felhasználók lemásolnak pénzintézeti beléptető felületeket, és e-mailben arra kérik az ügyfelet, hogy jelentkezzen be a hamis online felületen, megszerezve így a felhasználók személyes adatait. Minden olyan támadási kísérlet, amely valamilyen megtévesztésen alapuló technikával adatot próbál szerezni, ebbe a kategóriába tartozik. Egy előrepreparált honlap, illetve egy XSS támadás összekombinálásával, eredményes támadás intézhető a felhasználók nagy többsége ellen.

4.2.4. Input Validation (bemeneti adat ellenőrzése)

Ebbe a kategóriába tartoznak a különböző kód beszúrásos támadások. A bemeneti mezők nem megfelelő ellenőrzése különböző kártékony kódok beillesztését teszik lehetővé. Ide tartoznak többek között az SQL-injection típusú hibák, vagy az XSS típusú támadások is (Ezekről bővebben később olvashatunk).

4.2.5. Password management (jelszó kezelés)

Ebbe a kategóriába tartozik minden olyan típusú sérülékenység, amelyek jelszókezelési problémákkal hozható összefüggésbe. A jelszavak hosszának és komplexitásának minőségi kritériumai egy jó példa erre a sérülékenység faktorra. Például egy 4 karakter hosszú, csak kisbetűt tartalmazó jelszó önmagában rejti a veszélyt, így ezek kijavítása nélkülözhetetlen. Ide tartoznak továbbá a különböző rossz jelszólenyomat (hash) készítési és jelszó titkosítási technikák is. Amennyiben egy vizsgálat során valahol elavult lenyomat (hash) készítési algoritmussal találkozunk (pl: MD5), a sérülékenység ebbe a kategóriába lesz besorolva. Minden olyan probléma, ahol a jelszó kezelésén, tárolásán kell változtatni szintén ebbe a kategóriába fog esni.

4.2.6. Factory defaults (gyári beállítások)

A gyári beállítások önmagában hordozzák a veszélyeket. Legyen az web alkalmazás, hálózati eszköz, vagy bármilyen más hardver vagy szoftver elem, amennyiben gyári beállításokkal kerül üzembe helyezésre, az kockázatot rejt magában. Ilyen formán egy CISCO router admin/admin jelszóval való használata gyári beállítás használatára utal. (Természetesen ez a sérülékenység egyben a jelszókezelési sérülékenységi faktorba is tartozhat). A gyári beállításokat bárki megnézheti egy adott eszközhöz, vagy szoftverhez, így hozzáférhet a menedzsment felülethez, átkonfigurálva a hálózati elemet. A gyári beállítások problémája nem csak jelszavakhoz, hanem egyéb más technikai paraméterhez is kapcsolódhat. Jó példa lehet erre egy hálózati nyomtató rossz konfigurációja. Ha például az SNMP community sztringet a gyári beállításokon hagyva üzemeltetjük (public/private), akkor egy rosszindulatú felhasználó távolról is képes az eszközünket menedzselni, átkonfigurálni.

4.2.7. Access Control management (hozzáférés szabályozás)

Ebbe a kategóriába tartozik minden olyan sérülékenység, amely valamilyen rosszul beállított hozzáférés problémához vezethető vissza. Jó példa erre egy rosszul konfigurált VLAN,

amelyben minden felhasználó képes csatlakozni a szerverhez. Egy jól beállított hálózaton a szerver menedzselését kizárólag rendszergazdák végezhetik, így felesleges mindenki számára hozzáférést biztosítani a szerverhez. Szintén ebbe a kategóriába tartozik például egy nem megfelelő port-security-val ellátott hálózat. Amennyiben illetéktelen eszköz csatlakoztatható egy belső hálózathoz, úgy egy rosszindulatú felhasználó támadásokat hajthat végre a hálózaton.

4.2.8. Misconfiguration (konfigurációs hiba)

Ebbe a kategóriába több dolog tartozhat. Általában minden olyan problémát ide sorolunk, ami valamilyen téves konfiguráción, vagy konfiguráció hiányán alapszik. A gyári beálltásoknál említett problémák, egy az egyben itt is alkalmazhatóak. Ide tartoznak továbbá például a rosszul konfigurált NETBIOS megosztások is, amelyek következtében olyan könyvtárak vállnak olvashatóvá, írhatóvá, melyek egy átlag felhasználó számára általában nem elérhetőek. Az egyik legalapvetőbb hiba, a rossz SSL tanusítványok használata is ide sorolható, vagy a rosszul beállított hálózati protokollok titkosítatlan kommunikációja.

4.3. OWASP TOP 10

Mára az egyik legnépszerűbb internetes szolgáltatás a World Wide Web (WWW), amelyet nap, mint nap felhasználók milliói vesznek igénybe. A honlapokon megjelenő információ nagy részét általában valamilyen háttéradatbázisban tárolják, így nem csupán egyetlen szolgáltatásról, hanem több szolgáltatás összekapcsolásáról lehet beszélni. Az ilyen típusú weblapokat komplex rendszereknek hívjuk. Az összetett és komplex honlapok megjelenésével egyidejűleg jelentek meg új, a komplex honlapokra jellemző sérülékenységek. Nem meglepő tehát az a tény, hogy a legtöbb sérülékenység a webes alrendszer szolgáltatásaiban találhatóak, így mára a támadások nagy részét a WWW és az ahhoz kapcsolódó háttérszolgáltatások ellen indítják. Ebben a fejezetben megpróbáljuk összefoglalni azon vezető sérülékenységek listáját, amelyek a leggyakoriabbak a webes szolgáltatásokban.

Az OWASP (Open Web Applications Security Project) minden évben közzéteszi a webes alkalmazásokban előforduló leggyakoribb sérülékenységeket. Az OWASP TOP 10 sérülékenységek a következőek:

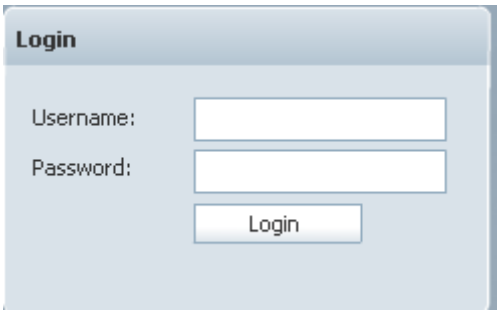
Osztályozás	Sérülékenység megnevezése
A1	Beszúrásos típusú támadások
A2	Hibás hitelesítés és sessionkezelés
A3	Cross-Site Scripting (XSS)
A4	Nem biztonságos direkt objektumhivatkozás
A5	Helytelen biztonsági beállítások
A6	Érzékeny adat nem megfelelő védelme
A7	Helyetelen URL és függvény validáció
A8	Cross-Site Request Forgery (CSRF)
A9	Hibás beépülő komponensek használata
A10	Nem ellenőrzött átirányítások és továbbítások

Az elmúlt évek tapasztalatai azt mutatják, hogy a leggyakoribb webes hibák az SQL-injection és XSS sérülékenységek. Bár ezeknek a hibáknak a kihasználása erősen technikai jellegű, mégis néhány példán bemutatva elengedhetetlen, hogy áttekintsük őket, mert a jelentések többsége valamelyik hibát nagy valószínűséggel tartalmazni fogja. Továbbá fontos tisztázni, hogy egy ilyen jellegű hibánál az elhárítást a mi oldalunkon kell kezdeni, vagy esetleg egy külső fejlesztő bevonása is szükségessé válik.

4.3.1. SQL-injection

Az SQL-injection sérülékenység napjaink egyik legelterjedtebb támadása olyan weblapok, alkalmazások ellen, amelyek adatbázist használnak. Az SQL-injection sérülékenység során egy rosszindulatú felhasználó képes elérni a háttéradatbázist, melynek kihasználásával egy támadó megszerezhet adatbázisban tárolt információkat, illetve módosíthatja azokat, futtatható állományokat tölthet fel a szerverre, és fájlokat olvashat ki a szerverről. Az alábbi néhány példán keresztül vizsgáljuk meg közelebbről, hogyan is néz ki egy ilyen típusú sérülékenység:

Nézzük az alábbi beléptető felületet:



A HTML forrásban megtalálhatjuk a login.php-t amely a beléptetésért felelős:

```
</head>
<body id="login" onload="document.getElementById('admin_name').focus()">
<form name="login"
 action="http://Proba/admin/login.php?AdminID=81e80fc6510e70dd750a513800
 " method = "POST">
  <fieldset>
    <legend>Admin Login</legend>
```

A login.php-ban az alábbi SQL lekérdezés található:

$q = mysql_query(„select tbd from users WHERE username = ’$username’ AND pass = ’$pass’”);
if (mysql_num_rows($q)>1) {Sikeres beléptetés esetén végrehajtandó forráskód}

Átlagos felhasználóként megadott felhasználónév/jelszó: **admin/admin**

A login.php-ben az SQL lekérdezés a következőképpen alakul:

$q = mysql_query("select tbd from users WHERE username = 'admin' AND pass = 'admin'");

Az alábbi példában a felhasználónév: admin és a jelszó: admin volt. Ha az SQL lekérdezés igaz és a felhasználónév/jelszó páros megtalálható volt az adatbázisban, akkor sikeresen beléptettük a felhasználót.

Rosszindulatú felhasználóként megadott felhasználónév/jelszó : **admin/ ' OR 'a'='a**

A lekérdezés ekkor így alakul:

$q = mysql_query("select tbd from users WHERE username = 'admin' AND pass = ' ' OR 'a'='a')

Az alábbi lekérdezés mindig igaz lesz az 'OR 'a'='a' feltétel miatt. Ebben az esetben tehát mindegy mi a jelszó, a feltétel mindig igaz lesz így az alkalmazás be fog minket engedni a védett oldalra. Ez egy jó példa egy tipikus SQL-injection támadásra. Egy rosszindulatú felhasználó SQL-parancsokat adott át az adatbázisnak az alkalmazáson keresztül. Az alkalmazás nem megfelelően szűrte a beviteli mezőket, így volt lehetőség megadni a jelszó

mezőben az ' OR 'a'='a SQL parancsot. Másik gyakori SQL-injection támadás az, hogy az URL-mezőben átadott paramétereket manipulálja egy rosszindulatú felhasználó.

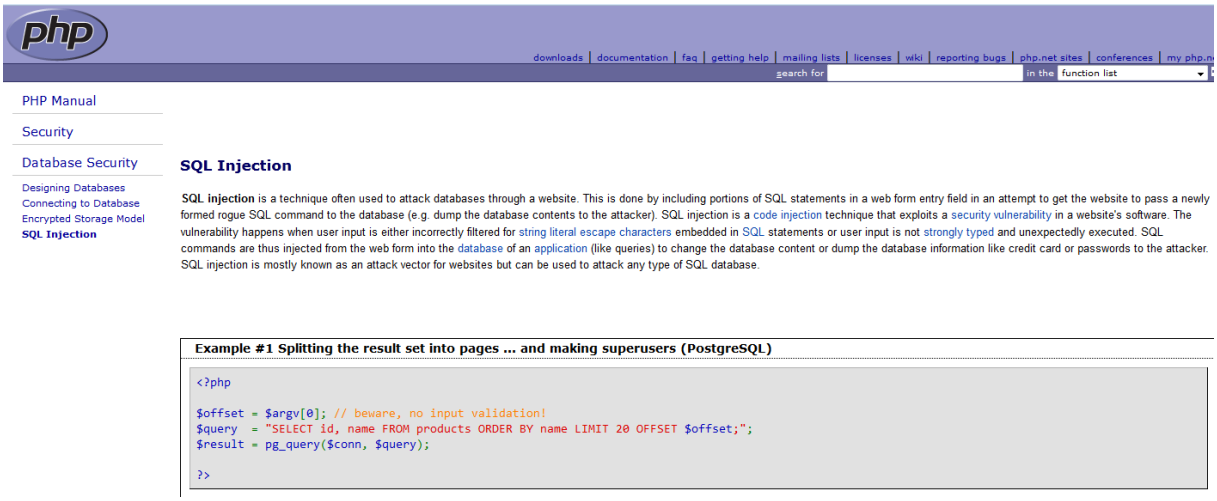
Nézzük az alábbi linket: *http://www.aldozat.hu/login.jsp?ReleaseID=4614*

A honlapon különböző tartalmú cikkeket jelenítenek meg adatbázisból. Ebben az esetben a böngészőnk a 4614-es számú cikket fogja az adatbázisból lekérdezni, és megjeleníteni. Ez egy nagyon elterjedt megoldás és a legtöbb honlapon alkalmazzák. Amennyiben az alkalmazás nem megfelelően szűri a beviteli mezőket, (jelen esetben a ReleaseID paramétert) akkor lehetőség nyílik további SQL-parancsokat kiadni a hibás paraméteren keresztül.

A hiba teszteléséhez hívjuk meg a következő linket:

http://www.victim.com/login.jsp?ReleaseID=4614 AND 1=1

A kimenet a következő:



Most az alábbi linket próbáljuk meg lefuttatni:

http://www.victim.com/login.jsp?ReleaseID=4614 AND 1=2

A kimenet a következő:

A különbség szembetűnő a két lekérdezés között. Az első változatban az AND 1=1 nem befolyásolja a lekérdezést, mert a feltétel mindig igaz, így a cikk megjelenik a böngészőnkben. (Jól látható, hogy megjelent az SQL-injection leírás ami a 4614-es adatbázis cikk volt). A második lekérdezésnél az AND 1=2 minden esetben hamis, így a teljes lekérdezés is hamissá változik, így nem jelenik meg semmilyen cikk a böngészőben, mert nem található az adatbázisban a feltételnek megfelelő cikk. Ez egy bizonyíték arra, hogy az adatbázis lefuttatja az általunk beinjektált kódot, tehát az SQL-injection típusú támadás lehetősége jelen van a rendszerben.

Napjainkban ez az egyik legveszélyesebb és legelterjedtebb, de mégis a legegyszerűbb támadási forma. Egy rosszindulatú felhasználó képes lehet letölteni a teljes adatbázist, vagy akár módosítani annak tartalmát.

Megoldási javaslat:

A javaslat nem teljesen egyértelmű. Minden esetben külön-külön meg kell vizsgálni, hogy hogyan előzhető meg az illetéktelen adatbevitel. Első megoldásként megfelelő lehet az olyan adatbázis specifikus szavak szűrése és eltávolítása az input paraméterekből, amelyeket az adatbázis értelmezni tud. Pl: (SELECT, AND, OR, FROM , UNION, stb..) Továbbá gondoskodni kell a felhasználói adatbevitel megfelelő ellenőrzéséről, mind az URL-ek, mind az űrlapok, mind a legördülő menük esetében. Ajánlott továbbá az alábbi karakterek teljes szűrése:

Felhasználói bemenetekből szűrendő karakterek listája			
|	pipe sign	\'	backslash-escaped apostrophe
&	ampersand sign	\"	backslash-escaped quotation mark
;	semicolon sign	<>	triangular parenthesis
$	dollar sign	()	parenthesis
%	percent sign	+	plus sign
@	at sign	CR	Carriage return, ASCII 0x0d
'	single apostrophe	LF	Line feed, ASCII 0x0a
"	quotation mark	,	comma sign
|	pipe sign	\	backslash
&	ampersand sign		

Természetesen előfordul olyan eset, amikor az input paraméterben kötelezően át kell tudnunk adni speciális karaktereket, vagy speciális szavakat. Ebben az esetben egyedi megoldásokat kell alkalmazni, és alkalmazás szinten összeállítani úgy a lekérdezéseket, hogy azok ne tegyék lehetővé az illetéktelen adatbázis elérést.

Minden esetben meg kell győződni, hogy az összes bemeneti paraméter ellenőrzése megtörtént-e, és az nem tartalmaz-e kártékony kódot, illetve az nem módosítja az eredeti lekérdezést. Erre nem létezik egyszerű és mindig működő technika. Ez függ az adatbázis típusától a programozási nyelvtől és az általunk megvalósított alkalmazáslogikától. Fontos megjegyezni, hogy megfelelő védelmet nyújt továbbá a megfelelő programozási technikák alkalmazása, amelyek nélkülözhetetlen elemei egy biztonságos rendszer tervezésének.

Mit tegyünk egy ilyen hiba láttán?

Az előzőekben bemutatott példák segítségével egy rosszindulatú felhasználó képes akár a teljes adatbázist eltulajdonítani. Amennyiben egy általunk megrendelt sérülékenység vizsgálati jelentésben SQL-injection hibáról olvasunk, az első és legfontosabb dolog, hogy beazonosítsuk, hogy a hiba javítását a mi oldalunkon kell eszközölni, vagy esetleg külső céget, beszállítót kell igénybe venni. Vizsgáljuk meg, hogy milyen alkalmazásban találták az SQL-injection hibát. Ha például nyilvánosan letölthető alkalmazásban, mint például egy ingyenes

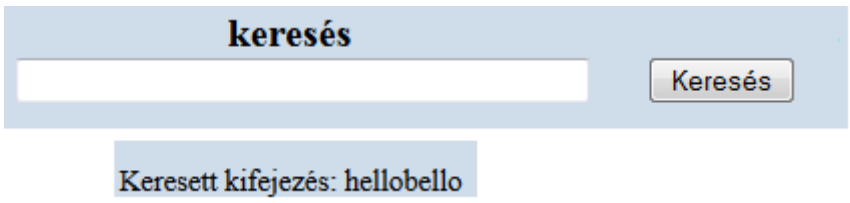
tartalomkezelő weboldal, amilyen például a Joomla vagy Wordpress, akkor a problémát mi oldalunkon kell orvosolni. Meg kell nézni, hogy elérhető-e frissítés, hibajavítás az adott sérülékenységhez, és a lehető legújabb verzióra frissíteni az alkalmazást. Ha egy olyan alkalmazásban találtak ilyen hibát, amit valamilyen külső cégtől egyedileg vásároltunk, akkor a hiba javítását nem a mi oldalunkon kell elkezdeni. Vegyük fel a kapcsolatot a gyártóval, és kérjük a hiba azonnali javítását. Fontos mérlegelni, hogy mennyire kritikus az alkalmazás, amelyben a hibát találták. Szükség esetén kapcsoljuk le a szolgáltatást, amíg a hibát megfelelően el nem hárítjuk.

4.3.2. Cross-site Scripting (XSS) sérülékenység

A Cross Site Scripting (XSS) sérülékenység napjaink másik legelterjedtebb sérülékenysége az SQL-injection típusú sérülékenységek mellett. Az XSS alapú támadások lényege, hogy a támadó futtatható scriptet (Pl: HTML kód, Java kód) képes input mezőkön, URL-eken keresztül beinjektálni a webes alkalmazásba, amely hatással lesz az alkalmazás megjelenésére, és egyéb tulajdonságaira. Az XSS sérülékenységeknek 2 fő csoportját különböztetjük meg.

4.3.2.1. Nem-tárolt XSS

Az XSS ezen fajtája a leggyakoribb és legelterjedtebb. Akkor fordul elő, amikor a kliens által szolgáltatott adatot a szerver-oldali futtatókörnyezet közvetlenül felhasználja a válasz előállításához. Ha ellenőrizetlen és HTML kódolás nélküli kliens-adat kerül be a szerverre, akkor így lehetővé válik, hogy a kliens-oldali kód a dinamikus lapba bekerüljön. Klasszikus példa erre a site keresőmotorja lehet: a keresett kifejezések közé HTML speciális karaktereket írnak be, és ha a keresőmotor válaszában visszaküldi a kódolatlan kereső stringet, akkor jön létre az XSS.

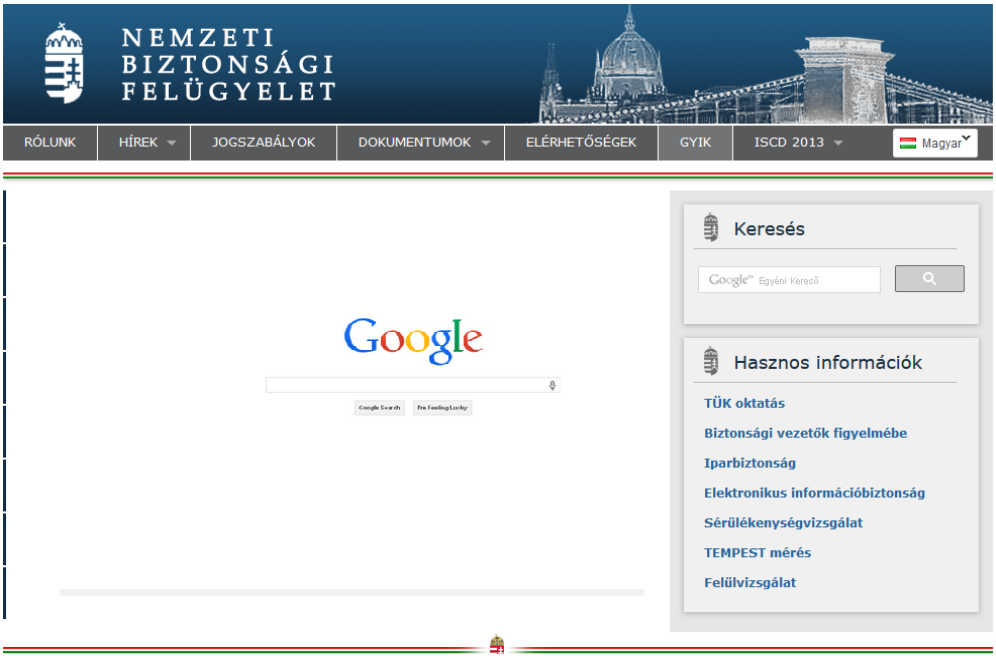


Amennyiben a kereső mezőbe értelmezhető kód kerül (Pl: <script>alert(”/XSS/”);</script>) az aktív részévé válhat a honlapnak, így a kliens oldalon megjelenik az /XSS/ message-box.

A támadás kizárólag kliens oldalon fut le, így a látszólagos módosítások nem hajtódnak végre a szerveren. A támadás kizárólag a célzott személyeknél jelenik meg. Az alábbi példán egy sérülékeny weblapot láthatunk, amelyben a kereső mező XSS sérülékenységet tartalmaz.

www.nbf.hu/search?q=<iframe src="[http://www.google.com/">

Az XSS eredményeképpen egy beágyazott iframe-n keresztül egy teljesen más honlap jeleníthető meg az áldozat számítógépén. Megjegyzés: Az alábbi kép csak illusztráció. A Nemzeti Biztonsági Felügyelet honlapja nem tartalmaz XSS sérülékenységet. Egy előre preparált honlap segítségével egy hasonló kinézetű honlap készíthető, amelynek segítségével félrevezető információk jeleníthetőek meg az áldozat böngészőjében.



Webes űrlapok és URL-ek inputadatainak nem megfelelő szűrése esetén egy rosszindulatú felhasználó kliens oldalon számos támadási formát hajthat végre:

- módosíthatja a weboldal megjelenését, tartalmát, ezáltal eléri, hogy a sérülékeny webszerver egy rosszindulatú felhasználó által meghatározott tartalmat jelenítsen meg a kliensoldali böngészőkben
- átirányíthatja a kliens böngészőt egy másik oldalra a felhasználó tudta nélkül
- ellophatja vagy módosíthatja egy felhasználó session azonosítóját és cookie-ait, így a rosszindulatú felhasználó képes lehet a célpont személyazonosságát meghamisítani és belépni a jelszóval védett területre, hozzáférve annak személyes adataihoz, beállításaihoz, tranzakcióihoz.

4.3.2.2. Tárolt típusú XSS

Az XSS ezen fajtája a legveszélyesebb. Ez a sebezhetőség akkor lép fel, ha a felhasználó által a web-alkalmazás felé továbbított adatait a szerver állandóan tárolja (adatbázisban, fájl rendszerben vagy más helyen), és később a felhasználóknak weblap formájában olvashatóvá teszi HTML kódolás nélkül. Klasszikus példa erre az online üzenet-felület. A szerver adatbázisban tárolja el a beinjektált kódot, így a honlapot böngésző összes felhasználó kliens oldalán végrehajtódik az injektált kód.

Megoldási javaslat:

Az XSS sebezhetőség elkerülésének első feltétele az, hogy minden speciális HTML karaktert kódoljanak.

Kódolás nélkül: <script>alert('xss');</script>

Kódolással: <script>alert('xss');</script>

A pirossal jelölt rész egy kliens-oldali scriptet mutat, a zöld pedig ugyanazt kódolva. A kódolt verzió egy böngészőben literálként fog megjelenni, és nem használható a HTML tag-ek speciális jelentése. Ezzel meg lehet előzni azt, hogy a HTML lapra egy speciális scriptet szúrjanak be.

Az összes input mezőt megfelelő védelemmel és szűrővel kell ellátni, ahhoz, hogy az XSS hibákat kiküszöböljük. Arra is figyeljünk oda, hogy a szűrés ne csak kliens oldalon, hanem

szerver oldalon is történjen meg. Tervezéskor figyeljünk oda, hogy az input mezőkből visszaírt szövegek ne közvetlen épüljenek be a forráskódba, hanem megfelelő technikával, jelenítsük azt meg úgy, hogy azt a böngésző ne futtatható kódként értelmezze.

5. Tesztelési módszertan

5.1. Bevezető

Az alábbi fejezet betekintést nyújt a sérülékenység vizsgálat módszertanába. A fejezet fő célja, hogy az informatikai biztonságért felelős vezetők átfogó képet kapjanak a módszertan lényegéről, illetve értelmezni tudjanak egy vizsgálat során létrejött sérülékenység vizsgálati jelentést. A fejezet végén életből vett példákon keresztül tekintjük át a legelterjedtebb sérülékenységeket. Nem a sérülékenységek kihasználásán és technikai végrehajtásán lesz a hangsúly, hanem, hogy a kimeneteket és eredményeket megfelelően legyünk képesek értelmezni. Továbbá segítséget próbálunk adni ahhoz, hogy egy felelős vezető eldönthesse, hogy a jelentésben szereplő sérülékenységek javítását mennyi időn belül, milyen erőforrás bevonásával lehet elhárítani.

5.2. Sérülékenység vizsgálat módszertana

A módszertan áttanulmányozása segít az informatikai rendszerek sérülékenység vizsgálatának értelmezésében, a sérülékenységek hatékony elhárításában. A sérülékenység vizsgálat célja megmutatni, hogy adott rendszer adott időpillanatban különböző irányokból milyen mértékig kompromittálható. Az alábbi táblázat a sérülékenység vizsgálat módszertani mátrixát mutatja:

	Jogosultság			
Irányultság		**Blackbox**	**Greybox**	**Whitebox**
	Külső			
	Web			
	Belső			
	Wi-Fi			
	GPRS / 3G			
	Social			

A mátrix kétféle felosztásban rendszerezi a sérülékenység vizsgálat módszertanát:

5.2.1. Jogosultság

Black-box: jogosultság nélküli vizsgálat

Ebben a fázisban a vizsgálatot minden előzetes információ nélkül végezzük el. Ekkor a vizsgálat célja, hogy megmutassuk, hogy egy rosszindulatú felhasználó milyen információt képes szerezni a hálózatról úgy, hogy semmilyen hozzáférése nincs a hálózathoz. A vizsgálat ezen szakasza rávilágít azokra a hibákra, amelyeket kihasználva felhasználó szintű, vagy akár rendszergazdai szintű hozzáférés szerezhető. A black-box vizsgálati módszertan van a legközelebb a rosszindulatú felhasználók által végzett tevékenységhez. A vizsgálat ezen szakaszában feltárt sérülékenységek világítanak rá azon hibákra, amelyeket egy esetleges hacker támadás során a támadó kihasználhat.

Grey-box: vizsgálat regisztrált felhasználói jogosultsággal

A vizsgálat ezen szakaszán felhasználó szintű jogosultsággal tárjuk fel a rendszer esetleges sérülékenységeit. A vizsgálat célja megmutatni, hogy a hálózatban regisztrált aktív felhasználó képes-e rendszergazdai jogosultságot szerezni.

White-box: vizsgálat adminisztrátori jogosultsággal

A white-box audit jellegű vizsgálat, melynek célja megfelelőségi listák alapján a rendszer állapotának ellenőrzése. Mivel teljes hozzáférésünk van a rendszerhez, ezért a cél nem a kompromittáció, hanem az olyan rejtett hibák feltárása, amely a grey-box és black-box vizsgálat során nem tártunk fel. Az eredmények kiértékelése után pontosan látható, hogy tételesen mennyi és milyen besorolású sérülékenység található a rendszerben.

Miért lehet szükség a white-box vizsgálatra?

„Rendszergazdai jogosultsággal mindent fel lehet törni.”

Vegyük az alábbi nagyon egyszerű, de gyakori példát. A vizsgálat során nem sikerült kompromittálni egy rendszert, és nem sikerült érvényes felhasználónevet/jelszót szerezni. Ez a tény nem jelenti azt, hogy a rendszer teljes mértékben biztonságos. A White-box vizsgálat során a dokumentumok és a forráskódok áttekintése után amennyiben azt tapasztaljuk, hogy a megengedett minimum jelszóhossz 3 karakter, akkor azt mindenképpen sérülékenységnek kell feltüntetni. (Ez egy olyan tipikus sérülékenység, amely nem feltétlenül jön ki a black-box és grey-box vizsgálat során). Számos, ehhez hasonló probléma derülhet ki a white-box audit során.

5.2.2. Irányultság

5.2.2.1. Internet felőli, külső sérülékenység vizsgálat

- Interneten fellelhető, publikus adatbázisokban való szabad keresés (pl. az internetes kereső portálokon végrehajtott "szabad szavas" keresés; whois adatbázisokban végrehajtott két utas lekérdezés; vállalat nevére, IP tartományára és e-mail címekre vonatkozó keresések)
- Célzott információgyűjtés a hálózati struktúráról, a telepített hardver- és szoftverelemekről, illetve a hálózati forgalomról (pl. az Interneten elérhető vállalati e-mailek, levelezőlisták adatai alapján az e-mail fejlécek kiértékelése, vállalati web portálon elhelyezett tartalom forrásának elemzése)
- Az elérhető számítógépek szolgáltatásainak, sebezhetőségének feltérképezése
- A fentiekben szerzett információk elemzése alapján további kézi vizsgálatok és betörési kísérletek elvégzése

5.2.2.2. Webes alkalmazások sérülékenység vizsgálata

Automatizált vizsgálatok:

Első lépésként automatizált célszoftverek kerülnek alkalmazásra, melyek eredményei megmutatják az ismert problémákat, mintázatokat és útvonalakat a webes alkalmazások biztonsági beállításaiban.Az automatizált felderítés elsődleges célja, hogy azonosítsa a webes alkalmazások kompromittálhatóságát az ismert sérülékenységek alapján.

Vizsgált sérülékenységi területek:

- Ismert sérülékenységek
- Alapbeállítások
- Cross Site Scripting
- SQL Injection
- Információ "kitakarás"
- Könyvtár indexek
- Brute Forcing
- Denial of Service támadások
- Parancsok futtathatósága
- Puffer túlcsordítás
- Server oldali meghívható állományok (Includes)

Kézi vizsgálatok:

A kézi vizsgálatok esetében a vizsgálatot végző személy értelmezi a kliens felé adott válaszokat, és befolyásolhatja a szervernek küldött adatokat. A kézi vizsgálat legnagyobb előnye a vizsgált rendszerre vonatkozó precíz, specializált végrehajtás. A kézi vizsgálatok képesek olyan hibákra rávilágítani, amelyre az automatizált vizsgálatok nem voltak képesek. A kézi végrehajtás lassabb, mint az általánosabb, automatizált eszközök, így a webes alkalmazások sérülékenység vizsgálatában az ún. hibrid vizsgálatok elvégzése vezet a legjobb eredményre.

Vizsgált sérülékenységi területek:

- Hozzáférési jogosultság problémák
- Hozzáférés kontroll problémák
- Session (munkafolyamat) státusz problémák
- Speciális felhasználói állományok
- Átmeneti állományok
- Megbízható kapcsolatok problémái
- Adatbázis problémák

5.2.2.3. Belső sérülékenység vizsgálat

Az informatikai rendszer sérülékenység vizsgálata a dolgozók által használt irodákból, black-box, grey-box és white-box módszerrel történik. A helyszíni vizsgálatok első fázisában (black-box) a hálózati csatlakozásoktól eltekintve a célrendszerekhez semmilyen hozzáférési jogosultság nem áll rendelkezésre.

Alkalmazott módszerek ismertetése:

Jogosultság nélküli sérülékenység vizsgálat (black-box):

- Célzott információgyűjtés a hálózati struktúráról, a telepített hardver- és szoftverelemekről, illetve a hálózati forgalomról:
    - Hálózati IP címek kiosztásának felderítése (DHCP / DNS / router broadcast scan, ping scans)
    - Hozzáférés kontroll problémák
    - Port Security beállítások felderítése, a védelmi mechanizmusok kikerülésének tesztelése

    - Hálózaton elérhető számítógépek felderítése (Ping scan, NetBIOS scan, Port scan)
- Az elérhető számítógépek szolgáltatásainak, sebezhetőségének feltérképezése célszoftverek segítségével (A futtatott célszoftvereket olyan paraméterekkel futtatjuk, hogy azok a lehető legkisebb mértékű fennakadást okozzák a rendszerek működésében).
    - Hálózaton elérhető számítógépek szolgáltatásainak feltérképezése (Application Scans)
    - Hálózati forgalom lehallgatása és elemzése (közbeékelődéses - Man in the Middle - ARP Poisoning technikával)
    - Jelszavak és azonosítók lehallgatása, kigyűjtése (LAN Manager, NTLM és cleartext jelszavak, lenyomatok)
    - Gyűjtött jelszavak feltörése, a jelszavak erősségének megállapítása
    - Identity Theft (a megszerzett account-ok felhasználása más felhasználói hozzáférés megszerzése érdekében, információgyűjtés)
- A fentiekben szerzett információk elemzése alapján további, kézi vizsgálatok és betörési kísérletek

Jogosultsággal végzett vizsgálatok (grey-box, white-box):

- Az elérhető számítógépek szolgáltatásainak, sebezhetőségének feltérképezése célszoftverek segítségével (A futtatott célszoftvereket olyan paraméterekkel futtatjuk, hogy azok ne okozhassanak fennakadást az rendszerek működésében).
    - A szervereken, munkaállomásokon futó operációs rendszerek, adatbázis kezelők és főbb alkalmazások biztonsági szintjének vizsgálata
    - A gyártók által kiadott biztonsági javítócsomagok naprakész állapotának ellenőrzése a rendszer elemein
    - A jelszó-követelmények és a gyűjtött kódolt jelszavak visszafejthetőségének vizsgálata

5.2.2.4. Vezeték nélküli hálózat (Wi-Fi, 3G, ...) sérülékenység vizsgálata

Alkalmazott módszerek ismertetése:

- WLAN hozzáférési pontok és Ad-Hoc kapcsolódási pontok keresése, feltérképezése
- IEEE 802.11a/b/g alapú WLAN forgalom lehallgatása és a gyűjtött anyag elemzése

- Titkosítási eljárások elemzése és a titkosítási kulcsok visszafejthetőségének ellenőrzése
- Az elérhető vezeték nélküli hálózatok sebezhetőségének vizsgálata célszoftverek segítségével
- Hamis wireless access point installálása és direkt csatlakozási kísérletek a kliensekhez
- A fentiekben szerzett információk elemzése alapján további kézi vizsgálatok lefolytatása

GPRS/3G szolgáltatás külső sérülékenység vizsgálata

- A GPRS csomagkapcsolt infrastruktúra hálózati felderítése
- A csomagkapcsolt hálózat elérhető szolgáltatásainak felderítése
- Automatikus sérülékenység vizsgálat
- Kézi sérülékenység vizsgálat
- Kézi sérülékenység vizsgálat a terminálok ellen
- Túlszámlázási lehetőségek ellenőrzése
- GPRS alapú szolgáltatások sérülékenység vizsgálata (pl.: MMS, WAP)

5.2.2.5. Social Engineering

A statisztikák lapján az érzékeny vállalati információk kiszivárgásának egyik legkomolyabb veszélyforrása az emberi tényező.

A Social Engineering vizsgálat az alábbi területekre terjed ki:

- A dolgozók általános informatikai kultúrájának, és a vállalatnál használt informatikai biztonsági előírások, szabályok betartásának vizsgálata
- Clear Desk Policy ellenőrzése
- Megtévesztő e-mail segítségével végrehajtott támadás
- Fizikai biztonsági teszt: publikus használatú irodarészek fizikai biztonságának vizsgálata, valamint eszközök hozzáférhetőségének ellenőrzése

5.2.3. White-box vizsgálat- hálózati audit

A vizsgálatot (legyen az külső, vagy belső) a white-box fázis teszi teljessé, amely egy komplett átvilágítási auditot jelent. A vizsgálat során adminisztrátori jogosultsággal, illetve

teljes hozzáféréssel vizsgáljuk át a hálózatot. Joggal merül fel a kérdés, hogy adminisztrátori jogosultsággal, bárki fel tud törni rendszereket. Fontos megjegyezni, hogy itt a cél már nem a behatolás, mint a black-box és grey-box vizsgálat során, hanem a lehető legtöbb sérülékenység feltárása, amelyek az előző fázisokban nem derültek ki. Ilyenek lehetnek a konkrét konfigurációs állományok elemzése, és egyéb biztonsági beállítások áttekintése (például jelszóbiztonsági házirend). A vizsgálat olyan hálózati gyengeségekre képes fényt deríteni, amely egy általánosított sérülékenység vizsgálat során nem feltétlenül derülnek ki. Fontos megjegyezni, hogy egy white-box vizsgálat sem feltétlenül tár fel minden sérülékenységet, így az ilyen vizsgálatok után is előfordulhat sérülékenység a rendszerekben. A célja ezen típusú vizsgálatoknak az, hogy minimalizálja a rendszerben előforduló sérülékenységeket.

5.2.3.1. Rendszer architektúra ellenőrzése

- Rendszerarchitektúra-vázlat (rendszerkapcsolati, vagy adatkapcsolati) megfelelősége
- Hálózati architektúra vázlat megfelelősége
- Topológiai vázlat és/vagy nyilvántartás megfelelősége
- A vizsgálat során az áttekintett részrendszerek, kapcsolatok szerepelnek-e a fenti dokumentumokban

A fenti részfeladatok mellett meg kell ismernünk a szervezet informatikai stratégiáját, hogy felülvizsgáljuk az ahhoz kapcsolódó szabályozási hátteret, és a különböző rendszerelemekre vonatkozó intézkedési terveket.

5.2.3.2. A fizikai biztonság ellenőrzése

A fizikai biztonság felülvizsgálatának célja annak megállapítása, hogy a hálózatot felépítő eszközök védve vannak-e az illetéktelen hozzáférésektől, hogy csak az arra hivatott személyek legyenek képesek a kialakításon, beállításokon változtatni.

Ellenőrizzük a hálózat WAN-LAN-, és kábelezési diagramját, hogy megállapítsuk az aktívelemek elhelyezkedését, a hálózat fizikai jellemzőit.

Mindezt az alábbi szempontok alapján vizsgáljuk:

1. munkaállomások kábelezése

   A munkaállomások kábelezésekor el kell különíteni a hálózati kábeleket az egyéb eszközöktől, elektromos vezetékektől az interferenciából adódó adatvesztés, sebesség veszteség elkerülése érdekében.

2. hubok, switchek elhelyezkedése

   Az aktív elemeket úgy kell elhelyezni, hogy csak az arra jogosultak férhessenek hozzájuk.

3. hálózatot üzemeltetők munkaállomásainak elhelyezkedése

   A rendszergazdák, operátorok munkaállomásait el kell különíteni az egyéb rendszerektől, beléptető-rendszerrel mindenkit azonosítani kell tudni, aki belép az ilyen jellegű helyiségekbe, különös tekintettel a szerver-szobákra.

4. hálózati management fizikai alrendszere

   A hálózat-managementet úgy kell kialakítani, hogy az aktív elemek, egyéb rendszerelemek távoli adminisztrációja csak meghatározott fizikai útvonalon legyen lehetséges.

5. a fizikai felépítés jelölései

   Az aktív elemek, hubok, switchek csatlakozásait fel kell címkézni, egyértelmű jelölésekkel meg kell határozni a kialakított kapcsolatokat.

5.2.3.3. A hálózati réteg biztonsága

A hálózaton csak olyan forgalom haladhat keresztül, amely engedélyezett tartalommal, engedélyezett forrásból, engedélyezett célpont felé halad.

Ellenőrizzük a hálózat fizikai működését a hálózati-diagramm tükrében, az alábbiak szerint:

1. IP-címek, DNS nevek, tartományok, szegmensek

   A címkiosztásnak, névfeloldásnak jól szervezettnek kell lennie, hogy ne fordulhasson elő duplikáció, IP cím lopás.

2. traceroute

   A munkállomásokról a különböző szegmensek felé indított traceroute kimeneteket elemezni kell és össze kell hasonlítani azért, hogy azok csak a meghatározott útvonalakon legyenek képesek kommunikálni.

3. traceroute maximum hops

   A traceroute vizsgálatot egy elérhetetlen címre is el kell végezni, hogy megállapítsuk, az összes munkaállomáson aktív a „maximum hops” beállítás.

4. icmp query

   A teljes hálózaton el kell végezni az ICMP alapú vizsgálatot, ennek eredményét össze kell hasonlítani a network-diagrammal, meg kell vizsgálni a nem válaszoló szegmenseket.

5. routerek
   - A routerek elhelyezkedését, konfigurációját felül kell vizsgálni. Route-táblák összehasonlítása.
   - A tűzfalakon, routereken a routing tábla update csomagokat (IGP, EGP) szűrni kell.
   - A külső kapcsolattal rendelkező routerek meghatározása, felülvizsgálata.
   - A source-routing csomagokat szűrni, tiltani kell a külső oldalról érkező forgalomban.
   - Az ICMP redirect utasításokat a routereknek el kell kerülniük, hogy a route-táblákat ne tudják a csomagok módosítani.
   - A külső routerek nem fogadhatnak el kívülről belső forráscímmel érkező kéréseket, el kell kerülni az IP address spoofing lehetőségét.
6. routerek üzemeltetése
   - A routerek üzemeltetését lehetőleg a saját szakembereknek kell végezniük.
   - Milyen rendszerességel történik route-állítás, milyen procedúra előzi meg?
   - A szoftver-upgrade milyen módon zajlik, ki végzi azt?
   - A konfigurációkról készül-e mentés, hol tárolják, ki kezeli?
   - Az összes telnet és egyéb management port védett? Nincsenek gyári passwordok?
   - A router passwordok kezelése? Hol? Ki? Secure módon?
   - A routerekhez kívülről teljesen lehetetlen hozzáférni?
   - Üzemel a rendszerben SNMP-alapú hozzáférés?
   - A routerek, switchek log-alrendszere milyen mélységű?
   - Az access-listák ellenőrzése.

5.2.3.4. Az átviteli réteg biztonsága

Meg kell győződnünk arról, hogy a TCP/IP alapú kommunikáció tervezetten, ellenőrzötten folyik-e. Ki kell zárnunk annak lehetőségét, hogy olyan kommunikációs portok is nyitva vannak a számítógépeken, melyek nem tartoznak a rendszer alkalmazásainak kommunikációjához. Ellenőriznünk kell az hálózaton áthaladó adatforgalom biztonságát, a titkosítási eljárások meglétét és erősségét. Vizsgálnunk kell a távoli management, az operációs rendszerek biztonságát, a rendszer szintű azonosítási rendszer megfelelő, biztonságos működését.

5.2.3.5. Rendszerkapcsolatok és ennek tervezettsége (kapcsolata az informatikai stratégiával)

- Adatáramlás módja az egyes rendszerek között (adatlista, egyedi jellemzőkkel)
- Interfészkapcsolatok zártsága (titkosság, integritás, hitelesség, letagadhatatlanság)

A fenti feladatokat az alábbiakra való figyelemmel végezzük:

1. A hálózaton áthaladó adatok típus-meghatározása.
2. Az áthaladó forgalom elemzése, különös tekintettel az adatok sérthetetlenségére, módosíthatatlanságára, titkosságára.
3. A küldő és fogadó oldali azonosítási procedúrák elemzése, különös tekintettel az átküldött adatok valódiságának, letagadhatatlanságának bizonyíthatóságára.

5.2.3.6. A tűzfalak, szerverek dokumentációja

- A vizsgálat elvégzéséhez szükséges dokumentumok (pl.: szegmens-diagrammok, szoftverek, hardverek, routerek, verzió-szintek leírásai, hosztnevek, IP-k, kapcsolatok leírásai), bármely speciális szabályzat, üzletmenet leírásainak összegyűjtése, amely érintheti a tűzfalak biztonsági beállításait.
- Annak megállapítása, hogy az esetlegesen a szervezetben tervezett változtatások, stratégiai célok eléréséhez szükséges módosítások mennyiben befolyásolhatják a rendszer komponenseinek elhelyezését, beállításait.

5.2.3.7. A tűzfalak, szerverek logikai hozzáférés szabályozása

A tűzfalmegoldások, szerverek különböző komponenseihez való logikai hozzáférés alatt az azokhoz való kizárólagos, egyénenként engedélyezett és azonosított (szoftveres, lokális, vagy távoli) hozzáférést értjük.

- Annak meghatározása, hogy ki, milyen hozzáférési szinttel, milyen azonosítási rendszerrel férhet hozzá a rendszerkomponensekhez. Az adminisztrátorok adatainak összegyűjtése, az autentikáció módjának, erősségének megállapítása, a konfigurációs konzolok, GUI-k elérésének feltérképezése.
- A jelszó-management átvilágítása:
    - létezik-e a jelszó-managementhez help/helpdesk rendszer?
    - Hová szükséges, és hová nem jelszóhasználat?
    - A jelszavak láthatóak-e a használat során?
    - A jelszavak a felhasználók által módosíthatóak-e?
    - A jelszavakat (min. 90 naponta) változtatják-e?
    - A jelszavakat nem használhatják-e fel újra (2 éves periódus az ajánlott)?
    - A minimum jelszóhossz meghatározott (min. 8 karakter)?
    - A jelszavak összetétele:
        - Nagybetű
        - Kisbetű
        - Számok
        - Speciális karakterek
    - A jelszavak a fent felsorolt csoportok közül 3 csoportot tartalmaznak-e?
    - A jelszavak tartalmaznak-e UID-ket?
    - Hány login próbálkozás után tiltja le a rendszer a belépést?
    - A próbálkozások mindegyike loggolt?
    - A felhasználó-nevek, jelszavak kódolva továbbítódnak a hálózaton?
    - Automatikus timeout eljárás létezik-e (a beragadás elkerülésére)?

5.2.3.8. A tűzfalak, szerverek logikai kapcsolatainak meghatározása

- A logikai kapcsolatok biztonsága:
    - Titkosított-e?
    - A távoli adminisztráció korlátozva van-e fix IP-re?
    - SSH-n TCP wrappereken kívül milyen módon csatlakoznak?
- Amennyiben TCP wrapper van használatban, meg kell bizonyosodnunk arról, hogy a „reverse look up” opció használatban van-e.
- Meg kell állapítanunk, hogy az összes bináris állomány eredetije tárolva van-e, milyen módon történik a felhasznált szoftverkomponensek patch-elése, updatje.

- A tűzfalszerverek, egyéb szerverek, kliensek betárcsázós (dial in access) kapcsolatainak felülvizsgálata
    - A modemek automatikusan bontják a kapcsolatokat egy előre beállított inaktivitás után, vagy megszakadt kapcsolatok esetén?
    - Kik tárcsázhatnak be?
    - Ki adja a betárcsázási jogokat?
    - Milyen biztonsági eljárást használnak a betárcsázós hozzáférések kontroljára?

5.2.3.9. A tűzfalak, szerverek konfigurációs alapjai

A tűzfalak, szerverek konfigurációja az azok működtetéséhez szükséges és elegendő beállítások az összes komponensen lépésről-lépésre megismételt folyamatát jelenti.

- Meg kell határoznunk, hogy a komponensek logikai/fizikai elhelyezkedése egyezik-e az intézmény tűzfal-, biztonsági-startégiájával.
- Megvizsgáljuk, hogy a komponensek a lehető legfrissebb verziójúak-e, ha nem miért. A patch-management összhangban van-e a termék support előírásaival.
- A security-, és rendszer-adminisztrátor naprakész ismeretekkel rendelkezik-e az esetleges biztonsági résekkel és hibákkal kapcsolatosan?

5.2.3.10. A tűzfal szoftver, operációs rendszerek konfigurációjának felülvizsgálata

- Alapesetben az operációs rendszerek, tűzfalak néhány portot nyitva tartanak az adminiszrátorok számára. Ezek hozzáférésének szabályozását kell felülvizsgálnunk először.
- Portscan vizsgálat
    - belső
    - külső/Internet oldali

  A porscan-nek ki kell terjednie az ICMP, UDP és TCP-re

- Az ún. stealth rule (ha létezik) beállítás ellenőrzése.
- Az ún. cleanup rule (ha létezik) beállítások ellenőrzése.
- A kapcsolati tábla beállításainak ellenőrzése a kapcsolatok számának és időlimitjeinek meghatározására.
- A szegmentált hálózatok közötti átjárás ellenőrzése, portscanek, hozzáférés-próbák egyik szegmensből a másikba. (belső hálózatról DMZ-be, és fordítva)

- Annak megállapítása, hogy a tűzfal mögötti kapcsolatok kódolt formában valósulnak-e meg, sniffing test.
- A szabályalkotás, módosítás procedúrájának átvilágítása
    - ki módosíthat
    - mikor (dátum, idő)
    - a módosítás oka

  Mindezeket naplózzák-e?

- Adatbányászati beállítások, lehetőségek ellenőrzése.
- Külön partíción van-e a tűzfalak, szerverek logállománya?
    - Létezik-e mirror a log állományokhoz?
    - Kik férhetnek mindehhez hozzá?

5.2.3.11. Üzletmenet-folytonosság vizsgálat

A cél a tűzfalak, szerverek, egyéb rendszerek komponenseinek minimális biztonsági kockázattal való folyamatos üzemben tartása.

- meg kell határoznunk a hibatűrés szintjét a tűzfalak és szerverek üzemeltetésében
- redundancia teszt elvégzése
- az ún. „egypontos hiba” elemzése, kiszűrése
- backup-alrendszer felülvizsgálata
- az off-site adattárolási procedúra átvilágítása
    - hol tárolódnak a biztonsági mentések?
    - kik, hogyan férhetnek hozzá?

5.2.3.12. Jogosultságok, jogosultságadminisztráció

- A jogosultság beállítás folyamata, naprakészsége, hitelessége
- Kiosztott jogosultságok indokoltsága
- Beállított jogosultsági profilok megfelelősége (adatbázis, operációs rendszer és alkalmazás szinten)

5.2.3.13. Policy azonosítás/meghatározás

- annak meghatározása, hogy mely policy-k vonatkoznak az információtechnológia területére, a kapcsolódó területekre

- annak meghatározása, hogy mennyire up-to-date policy-kal rendelkezik az intézmény, melyek az esetlegesen frissítést igénylő szabályzatok
- vannak-e esetlegesen folyamatban lévő, a jelenlegi audit ideje alatt zajló szabályzat-alkotó projektek

6. Sérülékenység vizsgálati projekt

6.1. Bevezetés

Adott szervezet döntéshozóiban felvetődhet egy fontos, tisztázandó kérdés, hogy miért van szükség egyáltalán sérülékenység vizsgálatra, ha jól működik az informatikai rendszerük. Az előzőekben bemutatott módszertan, illetve sérülékenységek áttanulmányozása egyértelműen rávilágít arra a tényre, hogy az informatikai rendszerek tulajdonságuknál fogva számos helyen tartalmazhatnak sérülékenységet az alkalmazás szinttől a hálózati kommunikáción keresztül az infrastruktúra szintig, ennek következtében döntő többségükben sikeresen támadhatóak, amiket a gyakorlati tapasztalatok is megerősítenek.

Sok esetben további kérdésként merül fel a döntéshozókban, hogy a sérülékenységeket a helyi üzemeltetés miért ne tudná feltárni. Egyértelműen kijelenthető, hogy nem azért nem tudja felderíteni egy szervezet a saját sérülékenységeit, mert nem lenne rá képes vagy mert nem elég jó az ott dolgozók szakmai felkészültsége (esetenként ez is probléma lehet, de nem ez a fő motívum), hanem leginkább azért, mert más a motiváció. A helyi üzemeltetés motivációja az, hogy működjön a hálózat, elérhetőek legyenek a szolgáltatások, a felhasználók el tudják végezni a munkájukat és a kiemelt felhasználók esetleges egyedi kéréseit kiszolgálják. A vizsgálatot végző szakértőknek azonban az a motivációja, hogy megtalálja azokat a pontokat a rendszerben, ahol a rendszer kompromittálható. A felsorolt motivációs és gondolkodásmódbeli különbségek okozzák azt, hogy az üzemeltetésen dolgozó szakemberek nem fogják tudni egyértelműen feltárni azokat a sérülékenységeket, amiket az erre kiképzett szakértők megtalálnak.

A vizsgálatot végző szekértők ugyanazzal az eszközkészlettel, ugyanazokkal a módszerekkel próbálják megtalálni a rendszer gyenge pontjait, mint amivel egy rosszindulatú támadó megpróbálná feltörni az adott vállalat biztonsági rendszerét, azonban van pár fontos különbség:

- Egy támadónak elég 1 gyenge pontot megtalálnia a rendszer kompromittálásához, míg, a vizsgálatot végzők feladata lehetőség szerint az összes különböző kockázati besorolású sérülékenység felderítése.
- Egy támadónak lehetősége nyílhat a támadást időben elnyújtani, illetve nem kell minden egyes rendszerelemet vizsgálnia, így a támadó mintázat belesimulhat az adott

informatikai rendszer normál üzemi működésének mintázatába, így nem lesznek feltétlenül kiugró értékek a naplóállományokban. Ezzel ellentétben a vizsgálatot végzőknek nem feltétlenül van arra ideje arra, hogy a tevékenységét elfedje, sokszor ez nem is célja a vizsgálatnak, így egy-egy sérülékenységvizsgálat általában jóval „zajosabb", mint egy egyedi rendszer kompromittáció.

- A támadó szándéka lehet károkozás, vagy a rendszer működésének megbénítása, azonban a vizsgálatot végzőnek soha nem célja (kivéve, ha a felkérő egyértelműen kitér erre, például DoS-DDoS sérülékenység tesztelésére esetén).

Amennyiben egy szervezet esetén felmerül a kérdés, hogy sérülékenység vizsgálatot folytatna le, a következő fejezetekre fontos figyelemmel lennie.

6.1. Sérülékenységvizsgálati igény felmerülése

Sokféle kiváltó oka lehet annak, hogy egy szervezetben miért merül fel a sérülékenység vizsgálat kérdésköre, alább néhány szokásos ok:

- incidens érte a szervezetet, és szeretnék azt felderíteni, illetve a jövőben megelőzni a hasonló visszaéléseket
- a vizsgálat elvégzését szervezeti szintű dokumentum írja elő
- új fejlesztés/rendszerelem integrációját tervezik a meglévő informatikai rendszerbe és fontos tudni, hogy az integráció okozhat-e biztonsági kockázatot a jelenlegi rendszerre nézve
- új vezetés pontos és független képet akar kapni a szervezet informatikai rendszeréről
- valamely szervezeti szintű feladat elvégzéséhez szükséges bemeneti adatként a vizsgálat végeredménye (pl.: fejlesztési, költségvetési kérdések, kockázatelemzés, szervezeti stratégia kidolgozása, stb)

Az igény felmerülhet akár az IT üzemeltetésen, akár a szervezet biztonsági egységében, akár döntéshozókban, egy közös jellemző van, általában mindenki tart a végeredménytől és nem feltétlenül támogatja a vizsgálat megrendelését/elvégzését.

Az információ biztonság területén dolgozókban megjelenhet az a félelem, hogy a vizsgálat végeredménye azt fogja mutatni, hogy nem végzik jól a munkájukat. Ilyenkor fontos megnyugtatni, az érintetteket, hogy a vizsgálat alapvetően nem ellenük, hanem értük van. A feltárt állapot lehet, hogy nem lesz majd nagyon pozitív, de mindenképpen egy olyan

alapállapotot rögzít, amihez képest mérhető majd az elmozdulás, amit a vizsgálat eredményeként előálló dokumentumok alapján az adott szervezet munkatársai önerőből, vagy külső segítséggel meg tudnak oldani. Ennek eredménye egyrészt egy lényegesen biztonságosabb informatikai rendszer lesz, amit könnyebb majd felügyelni, másrészt felhívja a döntéshozók figyelmét a terület fontosságára és erőforrás igényére.

A gyakorlati tapasztalatok is megerősítik, hogy egy-egy vizsgálat elvégzése és az azt követő biztonságnövelő intézkedések végrehajtása az egész szervezetre nézve pozitív hatással van mind a felhasználókat, mind az IT biztonság területén dolgozó munkatársakat, mind a döntéshozókat, mind pedig a szervezet információ biztonsági kitettségét tekintve.

6.2. Sérülékenységvizsgálat megrendelése

Amennyiben felmerült az igény adott szervezet részéről, hogy szeretnének sérülékenységvizsgálatot elvégeztetni saját rendszerükön, és beazonosításra került, hogy a vizsgálatnak pontosan mit kell érintenie, akkor a következő lépés a megrendelés, az ajánlatok bekérése. Egy sérülékenységvizsgálati projekt sikere mindig sok tényezőn múlik, fontos az alábbi szempontok figyelembevétele:

- A megrendelő legyen azzal tisztában, hogy mit vár el célként a sérülékenységvizsgálattól, az ő szempontjából mi az elérendő cél. Amennyiben ehhez esetleg már a kezdeti fázisban konzultációra van szüksége, akkor a kellő információk beszerzése érdekében egyeztessen szakértőkkel.
- Amennyiben a sérülékenységvizsgálat célja beazonosításra került, meghatározásra kerülhet, hogy pontosan mit szeretne megvizsgáltatni a rendszerén. Általános hibalehetőség a túl tág, illetve a túl szűk vizsgálati célterület („scope”) meghatározása. Egy teljes szervezet minden elemére kiterjedő vizsgálatának jelentős erőforrás igénye van, és nem is minden esetben célszerű azt egyben lefolytatni. Ugyanakkor egy nagyon szűkre szabott vizsgálati célterület meghatározása után a vizsgálat nem tudja feltárni a célterületet körülvevő rendszer sérülékenységeit, vagy hálózati kapcsolódási hiányosságait, így könnyen abba a hibába futhat a vizsgálat, hogy kimutatja, hogy az adott rendszer biztonságos, közben esetleg egy „mellette” lévő eszköz/alkalmazás sérülékenysége miatt a vizsgált elem is könnyedén kompromittálható. Összefoglalva a megrendelőnek jól meg kell tudnia határozni, hogy a vizsgálat pontosam milyen rendszerelemeket érintsen, ehhez jó, ha rendelkezik naprakész rendszertervekkel és szolgáltatás leltárral.

- Amennyiben a vizsgálandó célterület beazonosításra került, az mindenképpen jelenjen meg a megrendelői szerződében annak érdekében, hogy ez később ellenőrizhető, számon kérhető legyen.
- Már a megrendelés kezdeti fázisában legyen figyelembe véve, hogy egy-egy vizsgálat milyen határidőkkel végezhető el.

Amennyiben a vizsgálat célja meghatározásra és a célterület behatárolásra került, megkezdődhet a sérülékenységvizsgálati megrendelő dokumentum előkészítése.

A dokumentumban a következő információkat ajánlott feltüntetni:

- A feladat pontos meghatározása
    - A vizsgálati célterület pontos meghatározása (eszközök és alkalmazások típusa, számossága)
    - sérülékenységvizsgálat irányultságainak meghatározása:
        - külső kapcsolatok vizsgálata
        - webes szolgáltatások vizsgálata
        - belső hálózat vizsgálata
        - wifi hálózat vizsgálata
        - 3G/GPRS
        - social-engineering
- A projekt céljának meghatározása
- Eredménytermékek meghatározása
    - VEZETŐI ÖSSZEFOGLALÓ
        - a teljes projekt eredményének lényegre törő bemutatása
    - PREZENTÁCIÓ
        - vezetői
        - technikai
    - AKCIÓTERV ELKÉSZÍTÉSE
        - rövid táv: 1-3 hónap
        - közép táv: 1-6 hónap
        - hosszú táv: 1-12 hónap
    - RÉSZLETES HIBALISTA
        - kategória
        - felmérési állapot
        - kockázat
        - javaslat

- INTÉZKEDÉSI TERV, ELLENŐRZŐ LISTA
  - a hiba javításáért felelős személy
  - belső erőforrás igény
  - külső erőforrás igény
  - minőségbiztosító
  - időráfordítás
  - hibajavítás kezdete
  - hibajavítás vége
  - státusz (elkezdésre vár, nyitott, lezárt)
- EVIDENCE POOL
  - részletes tevékenységnapló dátum szerint rendezve, gépekre lebontva
- MELLÉKLETEK
  - felhasználói listák, adatok, adatbázisok, levelek, jelszavak, stb.
- A VIZSGÁLAT RÉSZLETES MÓDSZERTANA
  - a vizsgálat során használt módszertan részletes bemutatása

- Szakmai kompetencia bemutatása

A sérülékenységvizsgálati kiírásra beérkezett dokumentációk elbírálása után megkezdődhet a sérülékenységvizsgálati projekt előkészítése.

6.3. Szerződéskötés sérülékenységvizsgálatra

A szerződés a legtöbb esetben szervezetfüggő, egyedi kialakítású, mégis az alábbiaknak ajánlott megjelennie a szerződésben:

- szerződés tárgya
- megbízott jogai és kötelezettségei
- megbízó jogai és kötelezettségei
- megbízási díj
- fizetési feltételek
- teljesítési határidő, ütemterv
- teljesítés módja
- kapcsolattartás
- szavatosság, jótállás
- szerződésszegés
- kötbér
- a szerződés megszüntetése

- vis maior
- üzleti titok
- titoktartás
- vitarendezés
- vegyes rendelkezések

A szerződéskötést követően megkezdődhet a tényleges sérülékenységvizsgálat.

6.4. Sérülékenységvizsgálati Projektdefiníciós dokumentum

A sérülékenységvizsgálat menetét javasolt külön Projektdefiníciós dokumentumban rögzíteni, mely kitér az alábbiakra:

- Projekt megállapodás
- A projekt team felépítése
- A projekt célkitűzései és feladatai
- A cél
- A feladatok
- Adatkezelés
- A projekt ütemezése
- A projekt kritikus sikertényezői
- A siker és annak mérése
- A célkitűzések megvalósításának feltételei
- Technikai feltételek
- Személyi feltételek
- A projekt jelentési rendszere
- Kommunikációs felület
- Sérülékenység-vizsgálat jelentés
- A kommunikáció nyelve
- A dokumentumok formátuma
- A projekt tagok rendelkezésre állása
- Projektmódosítási kérések kezelésének rendje (PMK)
- Kapcsolattartók adatai
- Projekt logisztika
- Belépők
- Bemutatók és megbeszélések

- melléklet - projektmódosítási kérés
- melléklet - módszertan

A Projektdefiníciós dokumentumban rögzítettek szerit célszerű olyan projektvezetőt kinevezni mind a megbízói, mind a megrendelői oldalon, akiknek mind a lehetősége, mind a képessége megvan a hatékony intézkedésre.

Gyakran felvetődik a kérdés, hogy milyen erőforrásra van szükség a megrendelői oldalon. Alapesetben nincs jelentős erőforrás igénye a megrendelői oldalon a projekt végrehajtásának, de ezt az egyedi igények felülírhatják. Javasolt a legalább heti projektvezetői konzultáció, ahol megbeszélésre kerülnek az aktuálisan elvégzett feladatok és a tervezett vizsgálatok.

Belső vizsgálat esetén a megrendelői oldalnak biztosítania kell egy helyiséget, ahol a vizsgálatot folytató személyek dolgozni tudnak, valamint a helységben lennie kell alapértelmezett beállítású hálózati végpontnak. A vizsgálat későbbi fázisaiban szükség lehet alapértelmezett konfigurációjú munkaállomás biztosítására.

A különböző jogosultsági szintű vizsgálatokhoz (greybox, whitebox) szükséges adott jogosultságú felhasználói vagy adott esetben adminisztrátori hozzáférés létrehozása külön a vizsgálat idejére.

6.5. A sérülékenységvizsgálat menete

A sérülékenységvizsgálat megkezdése előtt a Projektdefiníciós dokumentumban minden rögzítésre került, azonban a vizsgálatot megelőzheti még egy tisztázó megbeszélés („kick-off”), ahol az operatív kérdések megbeszélésre kerülnek, majd a vizsgálat az ütemterv alapján kezdetét veszi. Amennyiben van olyan kritikus eszköz/szolgáltatás, amelynek a vizsgálata egyedi időbeosztást, különleges technikai elvárásokat igényel, azt mindenképpen fontos írásban rögzíteni, legkésőbb ezen a megbeszélésen.

Egy tipikus vizsgálat hozzávetőlegesen az alábbi határidőkkel működhet:

- külső kapcsolatok vizsgálata: 3-10 embernap
- webes szolgáltatások vizsgálata: 5-20 embernap
- belső hálózat vizsgálata: 30-60 embernap
- social-engineering: 10-20 embernap

Gyakran felvetődik a kérdés, hogy a vizsgálatról értesítsük-e az üzemeltetést vagy adott esetben a felhasználókat. Ez alapvetően mindig egy közösen eldöntendő kérdés. A megrendelő sok esetben szeretné tesztelni az üzemeltetés reagálását a vizsgálatra, ebben az

esetben az üzemeltetés jelzése után célszerű az üzemeltetést bevonni a vizsgálat további lefolytatásába.

A sérülékenységvizsgálat általában két bemutatóval zárul, egy vezetői prezentációval, ahol döntéshozói szinten és megközelítéssel kerül bemutatásra a projekt eredménye, majd a további teendők összefoglalása, illetve egy technikai prezentációval, ahol az üzemeltetés ismerheti meg a vizsgálat végeredményét, és lehetőség nyílhat személyes technikai konzultációra is.

6.6. A sérülékenységvizsgálat utáni teendők

A sérülékenységvizsgálat befejezése után a projekt eredménytermékeként előáll az intézkedési terv, mely tartalmazza a rövid-, közép és hosszútávon elvégzendő teendőket. Ez alapján célszerű egy biztonságnövelő, hibajavító („hardening") projekt definiálása, mely a megadott ütemezés alapján felszámolja a feltárt hiányosságokat. A hosszú távú intézkedések (12 hónap) elvégzése után javasolt a sérülékenységvizsgálat megismétlése annak érdekében, hogy a hibajavítások ellenőrzése megtörténhessen.

6.7. Jogi kérdések

Információs rendszer ellen elkövetett bűncselekményre a BTK 423.§ -424.§ az irányadóak:

Információs rendszer vagy adat megsértése

423. § (1) Aki

a) információs rendszerbe az információs rendszer védelmét biztosító technikai intézkedés megsértésével vagy kijátszásával jogosulatlanul belép, vagy a belépési jogosultsága kereteit túllépve vagy azt megsértve bent marad,

b) az információs rendszer működését jogosulatlanul vagy jogosultsága kereteit megsértve akadályozza, vagy

c) információs rendszerben lévő adatot jogosulatlanul vagy jogosultsága kereteit megsértve megváltoztat, töröl vagy hozzáférhetetlenné tesz,

vétség miatt két évig terjedő szabadságvesztéssel büntetendő.

(2) A büntetés bűntett miatt egy évtől öt évig terjedő szabadságvesztés, ha az (1) bekezdés b)-c) pontjában meghatározott bűncselekmény jelentős számú információs rendszert érint.

(3) A büntetés két évtől nyolc évig terjedő szabadságvesztés, ha a bűncselekményt közérdekű üzem ellen követik el.

(4) E § alkalmazásában adat: információs rendszerben tárolt, kezelt, feldolgozott vagy továbbított tények, információk vagy fogalmak minden olyan formában való megjelenése, amely információs rendszer általi feldolgozásra alkalmas, ideértve azon programot is, amely valamely funkciónak az információs rendszer által való végrehajtását biztosítja.

Információs rendszer védelmét biztosító technikai intézkedés kijátszása

424. § (1) Aki a 375. vagy a 423. §-ban meghatározott bűncselekmény elkövetése céljából az ehhez szükséges vagy ezt könnyítő

a) jelszót vagy számítástechnikai programot készít, átad, hozzáférhetővé tesz, megszerez, vagy forgalomba hoz, illetve

b) jelszó vagy számítástechnikai program készítésére vonatkozó gazdasági, műszaki, szervezési ismereteit más rendelkezésére bocsátja,

vétség miatt két évig terjedő szabadságvesztéssel büntetendő.

(2) Nem büntethető az (1) bekezdés a) pontjában meghatározott bűncselekmény elkövetője, ha - mielőtt a bűncselekmény elkövetéséhez szükséges vagy ezt megkönnyítő jelszó vagy számítástechnikai program készítése a büntető ügyekben eljáró hatóság tudomására jutott volna - tevékenységét a hatóság előtt felfedi, az elkészített dolgot a hatóságnak átadja, és lehetővé teszi a készítésben részt vevő más személy kilétének megállapítását.

(3) E § alkalmazásában jelszó: az információs rendszerbe vagy annak egy részébe való belépést lehetővé tevő, számokból, betűkből, jelekből, biometrikus adatokból vagy ezek kombinációjából álló bármely azonosító.

Mire kell figyelni egy szerződéskötéskor?

Számítógépes rendszerekbe történő behatolás akár bűncselekmény is lehet, így fontos szerződésben tisztázni, hogy a sérülékenység vizsgálat nem jogosulatlanul történik, hanem felkérésre. A szerződésben tisztázni kell a vizsgálat pontos dátumát, a vizsgálatban résztvevők személyes adatait, és egyértelműen definiálni kell a vizsgálat tárgyát. A BTK 423. § (1) pontja egyértelműen kimondja: „Aki a) információs rendszerbe az információs rendszer védelmét biztosító technikai intézkedés megsértésével vagy kijátszásával ***jogosulatlanul belép…****” Amennyiben felkérés van egy biztonsági teszt végrehajtására, úgy az információs rendszerek kijátszása nem jogosulatlanul történik, hanem felkérésére. Fontos, hogy a jóváhagyó nyilatkozatot olyan felelős vezető hagyja jóvá, akinek jogköre van a vizsgálat elrendeléséhez.*

7. Esettanulmány

Ebben a fejezetben egy valós életből vett sérülékenység vizsgálati jelentés tartalmát vizsgáljuk meg. Fontos megjegyezni, hogy az alábbi esettanulmány bemutatásának célja, hogy egy felelős vezető megfelelően tudja értelmezni a jelentésben leírtakat. Nem a technikai kivitelezésre fordítjuk a fő hangsúlyt, hanem arra, hogy ha ilyen típusú hibákkal találkozunk, megfelelően tudjunk az elhárításában közreműködni, illetve a megfelelő szakembereket bevonva tudjuk elhárítani azokat. A fejezetben alapvetően az alábbi struktúrát fogjuk követni:

- sérülékenység bemutatása rész megoldási javaslattal
- értelmező rész, amelyben a leggyakoribb kérdéseket tisztázzuk.

Az esettanulmányokon keresztül megfelelően el lehet sajátítani, hogy a manapság legelterjedtebb hibák mit is takarnak pontosan, hogyan kell azokra megfelelően, jól reagálni.

Fontos megjegyezni, hogy az alábbi sérülékenységek az életből vett példák, azonban a konkrét adatok, illetve technikai paraméterek anonimizálva lettek. A jelentésben a Kitalált Bt. Nevet fogjuk alkalmazni, továbbá minden technikai paramétert (pl: IP címek, szerver nevek, honlap nevek) megváltoztattunk, hogy a valós szervezethez semmilyen módon ne legyen köthető az adott jelentés. Egy sérülékenységvizsgálati jelentés esetenként több száz oldal is lehet, így csak néhány ízelítő példát emeltünk ki egy külső illetve belső sérülékenységvizsgálati jelentésből, segítve megérteni, hogy hogyan is néz ki egy ilyen jelentés a gyakorlatban.

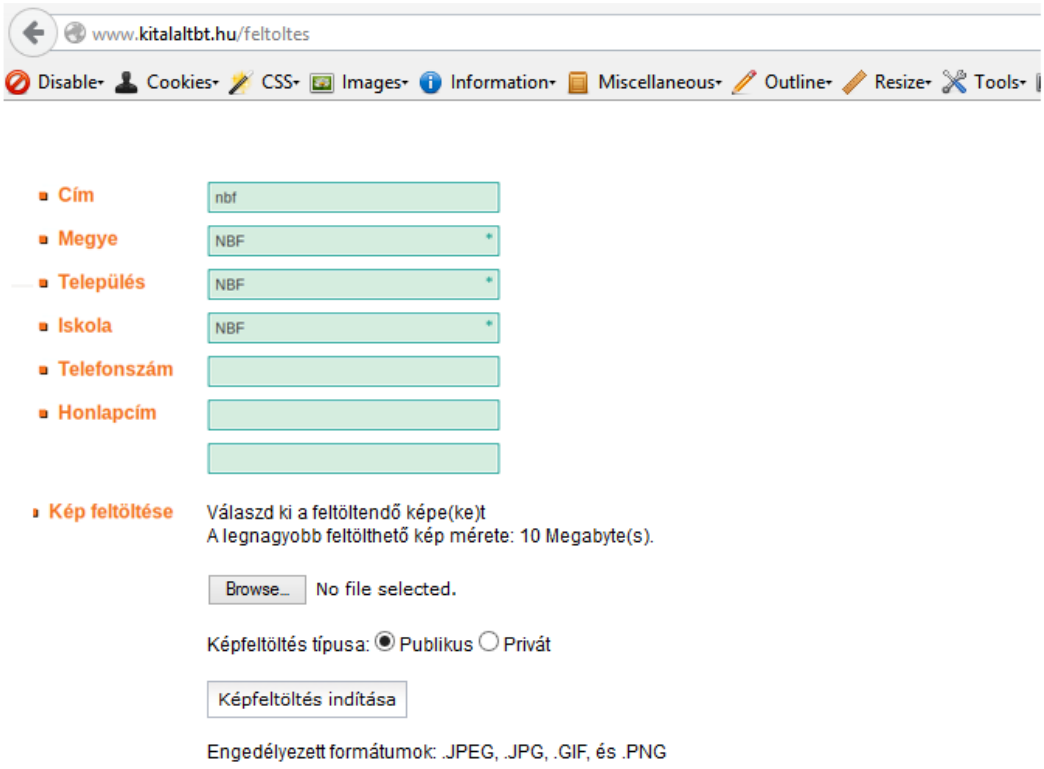
7.1. Külső sérülékenység vizsgálat jelentés - MINTA

_____ KÜLSŐ MINTAJELENTÉS _____

7.1.1. Fájl feltöltési lehetőség a Kitalált Bt. oldalán

Kategória: Konfigurációs hiba, alkalmazás logikai hiba
Kockázat mértéke: Kritikus

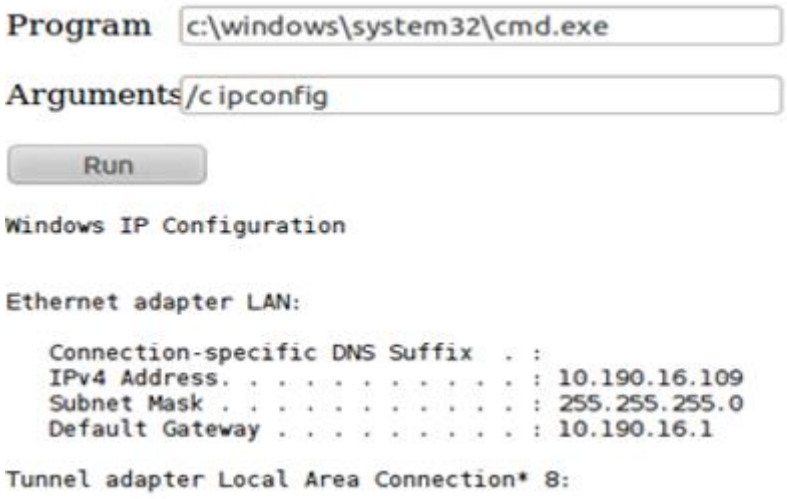
Felmérési állapot: A Kitalált Bt. oldalára egy rosszindulatú felhasználó képes fájlokat feltölteni a cég egy publikus fórum szolgáltatásán keresztül, a regisztráció menüpont alatt. A fájl feltöltésnél a kép kiterjesztését a szerver javascript-ből ellenőrzi kliens oldalon, amely egyszerűen megkerülhető, így tetszőleges kiterjesztésű állományt lehet feltölteni:



A regisztráció után az új felhasználóval belépve az oldal forrásából jól látszik, hogy a feltöltött kép elérési útvonala könnyen meghatározható, így a kitalaltbt.hu/images-upload/ után írva az elérési útvonalat a fájl futtathatóvá válik. Jelen esetben egy proba.aspx szerver oldali futtatható fájl lett feltöltve.

```
  <div class="inputs">
      <input type="hidden" name="upload-uploaded" value="proba.aspx" />
      <input type="hidden" name="upload" value="/kepek/uploadreg/109f0b77-9888-437a-9763-
a8441cd7688b/proba.aspx" />
      <span id="Upload">proba.aspx</span></div>
    </div>
```

Kockázat: A sérülékenységet kihasználva egy támadó képes tetszőleges kódot futtatni a szerveren. Az alábbi kép egy futtatható cmd.aspx feltöltését szemlélteti, amely tetszőleges parancs futtatását teszi lehetővé.



További parancsok adhatóak ki a szerveren, amely segítségével további érzékeny információkhoz juthat egy támadó:

Kiadott parancs: **dir C:**

```
Volume in drive C has no label.
Volume Serial Number is C80E-E8AC

Directory of c:\

10/17/2011  11:40 AM    <DIR>          !backup
10/05/2011  11:20 AM    <DIR>          inetpub
09/30/2011  02:11 PM    <DIR>          Install
01/19/2008  11:11 AM    <DIR>          PerfLogs
04/13/2010  03:11 PM    <DIR>          Program Fájls
02/24/2009  05:53 PM    <DIR>          Program Fájls (x86)
02/25/2012  04:57 PM                 0 rad2C0DA.tmp
02/27/2012  08:02 PM             5,485 rad3957D.tmp
02/25/2012  04:57 PM                 0 radC238E.tmp
02/27/2012  08:01 PM             5,485 radC6205.tmp
09/10/2009  11:52 AM    <DIR>          temp
01/18/2012  05:47 PM    <DIR>          update
03/19/2011  01:22 PM    <DIR>          Users
02/29/2012  04:53 AM    <DIR>          Windows
               4 Fájl(s)         10,970 bytes
              10 Dir(s)  30,445,568,000 bytes free
```

Kiadott parancs: **type c:\inetpub\kitalaltbt.hu\web.config**

Output: részlet a web.config fájl tartalmából:

```
      <!--SmartUrl-->
            <add key="SmartUrlSiteID" value="3756e6a3-49d2-4c5f-11f4-6da6c8c12347" />

<!--
            <add key="SmartUrlConnectionString" value="Data Source=SNSQL03\SQL2008;Initial Catalog=smarturl;User
ID=KitalaltBT;password=ugysetudod" />
-->

            <add key="SmartUrlConnectionString" value="Data Source=sql02\sql02;Initial Catalog=smarturl;Integrated
Security=SSPI;" />
```

```
<add key="SmartUrl_Trace" value="true" />
<!--SmartUrl END-->
<add key="EShopServiceUrl" value="http://shop.kitalaltbt.hu/Service.asmx" />
<add key="EShopHost" value="www.kitalaltbt.hu" />
<add key="EShop_Trace" value="true" />
<add key="EShop_AllowAnonymousCart" value="true" />
```

Egy támadó jelszavakat, illetve egyéb érzékeny információkhoz juthat hozzá egy egyszerű fájlfeltöltési probléma kihasználásával, és teljes mértékben képes átvenni a szerver felett az irányítást.

Javaslat: A képfeltöltés kontrollálására elengedhetetlen a szerver oldali ellenőrzés bevezetése, mert kliens oldalon ez könnyen kikapcsolható. Nem elegendő kizárólag a file kiterjesztésének ellenőrzése, hanem szükségszerű további védelmi szinteket is bevezetni. Futtatható állományok feltöltésének megakadályozása, a feltöltendő filek elérési útvonalának elrejtése , stb.

7.1.2. Belső szerverek és felhasználók kompromittálása

Kategória: Konfigurációs hiba, alkalmazás logikai hiba

Kockázat mértéke: Kritikus

Felmérési állapot: A Kitalált Bt. oldalára egy rosszindulatú felhasználó képes fájlokat feltölteni a cég egy publikus fórum szolgáltatásán, ahogy az az előző pontban bemutatásra került. Egy támadó olyan hálózati parancsokat képes kiadni, amelynek segítségével további belső gépek és felhasználók is kompromittálhatóvá válnak

Kiadott parancs: **net config workstation**

Output:

```
Computer name                \\KITALALTDMZ02
Full Computer name           kitalaltdmz02.extranet.dmz
User name                    _kitalaltextdmz_portal

Workstation active on
        NetbiosSmb (000000000000)
        NetBT_Tcpip_{B9D3C53A-148A-4B27-1FA9-FBB9251890C8} (00155D6026A2)

Software version             Windows Server (R) 2008 Enterpr

Workstation domain           EXTRANET
Workstation Domain DNS Name  extranet.dmz
```

```
Logon domain                 EXTRANET

COM Open Timeout (sec)       0
COM Send Count (byte)        16
COM Send Timeout (msec)      250
The command completed successfully.
```

Az outputból jól kivehetőek további hasznos információk:

- A számítógép neve: \\KITALALTDMZ02
- A bejelentkezett felhasználó: _kitalaltextdmz_portal
- A DOMAIN amelyben a számítógép található: EXTRANET

A net use /DOMAIN paranccsal lekérdezhetjük a DOMAIN CONTROLERTŐL a felhasználókat, akik a belső hálózaton regisztrálva vannak:

Kiadott parancs: **net use /DOMAIN**

Output:

```
The request will be processed at a domain controller for domain extranet.dmz.


User accounts for \\KITALALTDC01.extranet.dmz

-------------------------------------------------------------------------------
_sql08db01           _edsd        _extpsql08db02
_adminkitalalt2
extpsql08db02as      _extranetadservice      _extsql2svc
_extsql3svc          _ExtSqlSvc              _fcsservice
_fiokberlo_svc       _ftpreader              _gwssvcuser
abda                 AbrokV                  accord
acs                  acsa                    AcsJ
acstesz              aczelb                  AdamJ
adasztevel           admba                   admiCs
adminB               adminB                  AdmBSz
Admindi              adminEB                 adminep
adminka              AdminKitalalt           AdminK
```

A listán több ezer felhasználónév szerepel, amelyekkel egy támadó hozzájuthat a belső hálózaton található felhasználónevek listájához.

Az outputból egy támadó megtudhatja a DOMAIN CONTROLER pontos nevét, amely a felhasználók névsorát adta ki: \\KITALALTDC01

Egy támadó képes lekérdezni az egyes számítógépeket NETBIOS-on keresztül, hogy vannak-e olyan megosztások, amelyekhez felhasználónév/jelszó nélkül lehet csatlakozni.

Végignézve a szervereket több ilyen megosztást lehet találni, azonban az egyik legnagyobb kockázatot a \\kitalaltrgw megosztás jelenti.

Kiadott parancs: **net view \\kitalaltrgw**

Output:

```
Shared resources at \\kitalaltrgw

Share name  Type  Used as  Comment

-------------------------------------------------------------------------------
FPasswd Disk  R:
The command completed successfully.
```

Az FPasswd könyvtárat fel lehet csatolni felhasználónév/jelszó nélkül:

Kiadott parancs: net use g: \\ kitalaltrgw\Fpasswd

Így a G:\-t kilistázva hozzáférhetünk a \\ kitalaltrgw\Fpasswd megosztáshoz.

Kiadott parancs: **dir** g:\

Output:

```
 Volume in drive G has no label.
 Volume Serial Number is 241A-9FE3

 Directory of G:\

01/19/2009  03:05 PM    <DIR>          .
01/19/2009  03:05 PM    <DIR>          ..
01/19/2009  02:59 PM             3,391 EXTRANET.CMD
01/19/2009  03:08 PM            11,817 extranet.txt
               6 Fájl(s)        313,621 bytes
               2 Dir(s)  10,081,577,472 bytes free
```

Az extranet.txt fájl-ban a következő sor volt megtalálható:

Kiadott parancs: type g:\extranet.txt

Output: (részlet)

```
\\server01.extranet.dmz -u EXTRANET\ADMIN_kitalalt  -p dasd!%!--TRdd -c run.cmd
```

A fájl-ban egy felhasználót lehetett találni jelszóval együtt:

Felhasználó: **EXTRANET\ADMIN_kitalalt**

Jelszó: **dasd!%!--TRdd**

A megszerzett felhasználónév és jelszó DOMAIN Adminisztrátori jogosultsággal bír, így a hálózaton hozzáférhetővé vált a többi szerver is.

Mit tegyünk egy ilyen hiba láttán?

A fenti hibát kihasználva egy külső támadó képes volt a belső szervert kompromittálni és a hálózaton további szervereket, felhasználókat kompromittálni. Ez egy jó példa ara, hogy egy külső vizsgálat hogyan tud átalakulni belső vizsgálattá, illetve, hogyan lehet belső információkhoz jutni egy rosszul konfigurált web szerver hibáit kihasználva. A leggyakoribb probléma, hogy a vállalatok szerverei nem elszeparálva helyezkedik el a belső hálózaton, így a szervert kompomittálva a legtöbb esetben a támadó hozzáfér a belső hálózat egyéb adataihoz is. Fontos látni, hogy a webszerverek védelmére különös hangsúlyt kell fordítani, mert hibák tömkelegei teszik lehetővé a belső hálózat kompromittálását.

7.2. Belső Sérülékenység vizsgálat jelentés – MINTA

Az alábbi minta jelentés egy belső vizsgálatból származatik. Jól látható, hogy a belső vizsgálat illetve egy külső vizsgálat teljesen más jellegű hibákat tárhat fel.

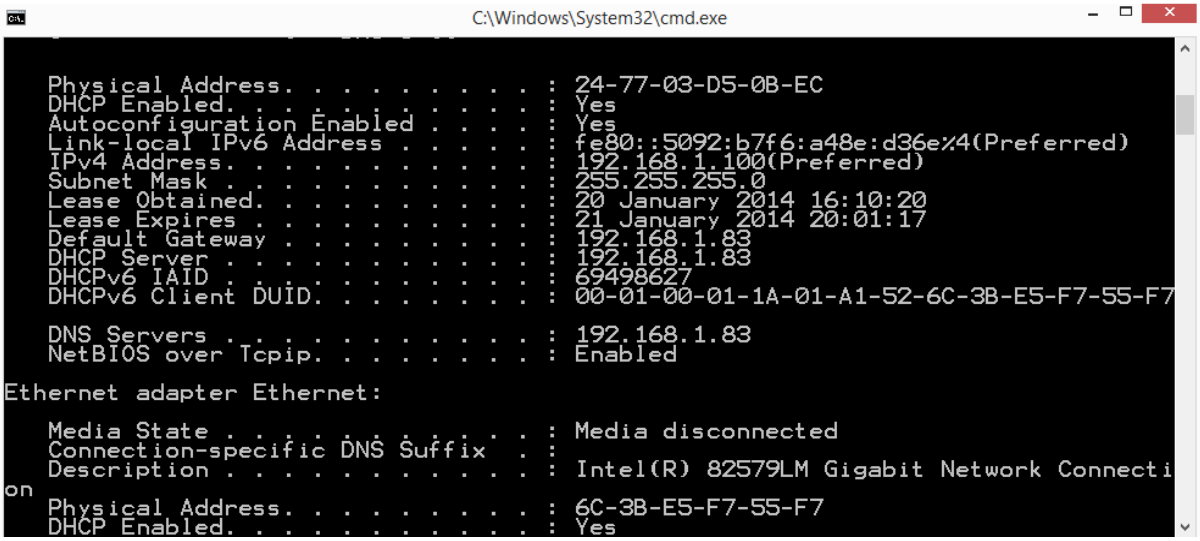
_____ MINTA _____

1.1 Hálózati végponti védelem hiánya (Port Security)

Kategória: Hozzáférés kontroll probléma, konfigurációs hiba

Kockázat mértéke: Közepes

Felmérési állapot: A végpontokra csatlakoztatott számítógépeink hálózati kommunikációját a hálózat végponti védelem nem akadályozza. A LAN hálózatba illesztett eszköz az IP címet a DHCP kiszolgáltatótól megkapta az egyéb szükséges információkkal együtt (DNS, Default Gateway, stb.).



Kockázat: Megfelelő hálózati végponti védelem (Port Security) nélkül jogosulatlan felhasználók idegen eszközöket csatlakoztathatnak a Kitalált Kft. informatikai rendszerébe.

Javaslat: A tapasztalt probléma megszűntethető a hálózati hozzáférés-szabályzás (NAC, Network Access Control) nevű védelmi technológia bevezetésével, amely védelmet nyújt a MAC cím hamisítással szemben is. A NAC megoldásoknak három fő feladata van:

- Hálózati azonosítás

  Az idegen, nem a vállalat által menedzselt informatikai eszközök elkülönítése emberi beavatkozás nélkül (a hitelesítéshez használt felhasználónév/jelszó páros vagy tanúsítvány nem, vagy sokkal nehezebben hamisíthatóak, mint pl. egy MAC cím).

- Új, ismeretlen (pl. vírus) támadások csillapítása, megállítása

  A NAC megoldások bevezetésének kulcsfontosságú támogatást az ad, hogy képes megakadályozni a nem megfelelő antivírus védelemmel, nem naprakész szoftverfrissítésekkel vagy helyi behatolás-detektáló rendszerrel nem ellátott kliens számítógépeket, hogy elérjék a céges hálózatot.

- Vállalati biztonsági szabályok kikényszerítése

  A biztonsági szabályokat megsértő számítógépek azonosításra kerülnek, a rendszer karanténba zárja azokat. Itt lehetőségük van a javításokat (többnyire automatizáltan) elvégezni, mely a vállalati informatikai erőforrások elérésének előfeltétele.

Mit tegyünk egy ilyen hiba láttán?

A fenti hiba egy nagyon gyakori probléma, és rengeteg helyen lehet vele találkozni. A hiba röviden annyit jelent, hogy bármilyen idegen eszköz csatlakoztatható a hálózatunkhoz. Megfelelő hálózati hozzáférés szabályozás bevezetésével a probléma orvosolható. Végezzünk költségelemzést, hogy szükséges-e új tűzfal, NAC, és egyéb eszköz beszerzése, ugyanis sok esetben a vállalati infrastruktúra már támogatja a megoldást, csak emberi beavatkozás és a hálózati eszközök átkonfigurálása szükséges.

1.2 Titkosítatlan (clear text, plain text) protokollok használata

Kategória: Konfigurációs hiba

Kockázat mértéke: Közepes

Felmérési állapot: A vizsgálat során titkosítás nélküli protokollokkal, illetve adatforgalommal találkoztunk (pl. FTP, SNMP, http). Számos szolgáltatás közülük nem feltétlenül szükséges (pl. alapvető TCP/IP szolgáltatások, FTP, stb.).

Kockázat: Titkosításra nem képes protokollokban található információ a hálózat lehallgatása által illetéktelen kezekbe kerülhet, valamint hozzáférés nyerhető az adott eszközhöz, szolgáltatáshoz.

Javaslat: A titkosított adattovábbításra nem képes programokat, protokollokat le kell cserélni a biztonságos megfelelőjükre (pl.: az SSH-ra, vagy SSL alapú változatukra), vagy pedig a titkosítást a teljes adatforgalmon be kell vezetni (pl. IPsec felhasználásával). Amennyiben ez nem lehetséges, akkor az érintett rendszereket a lehetőségekhez mérten, hálózati szinten kell elkülöníteni, tűzfallal védeni. A nélkülözhető szolgáltatásokat a támadási felület minimalizálása érdekében le kell állítani. Ezután az érintett jelszavak cseréje, az Informatikai Biztonsági Szabályzat minőségi kritériumai (hossz, komplexitás, egyediség, csere gyakorisága) alapján indokolt.

Mit tegyünk egy ilyen hiba láttán?

Gyakori probléma, hogy sok olyan alkalmazás és szolgáltatás fut a hálózaton, amelyet egy rosszindulatú felhasználó lehallgathat. Ezen hibák általbán viszonylag gyorsan orvosolhatóak, így a belső szervereink átkonfigurálása nagyban segítheti a hálózatunk biztonságát.

1.3 Alapértelmezett jelszavú hálózati nyomtatók, nyomtatószerverek

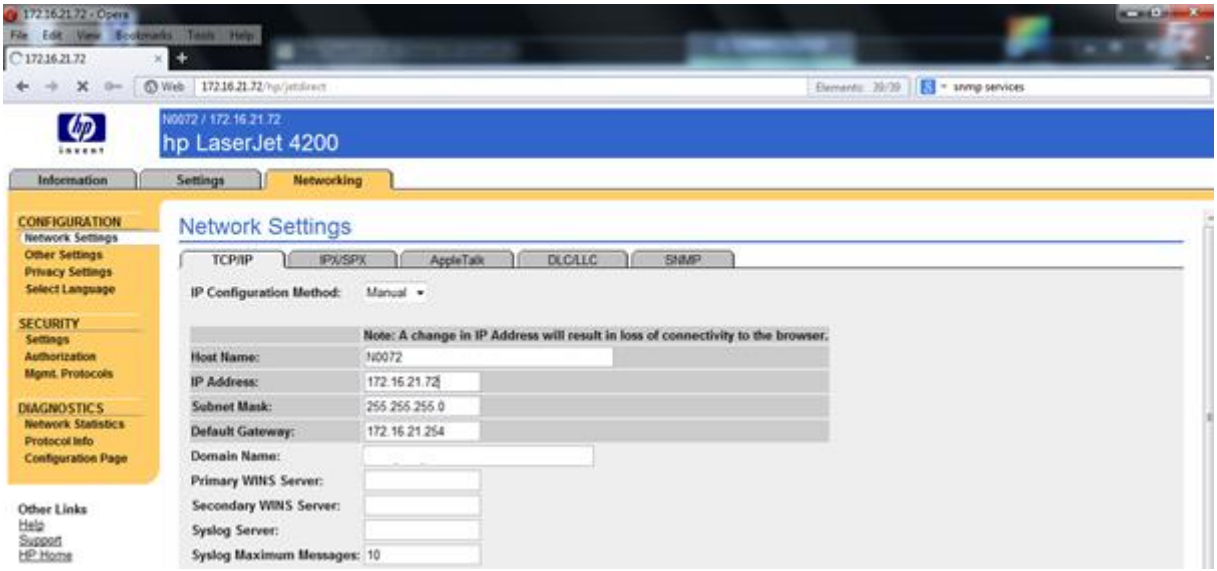
Kategória: Jelszókezelési probléma

Kockázat mértéke: Közepes

Felmérési állapot: A vizsgálat során több hálózati nyomtatóhoz, nyomtató kiszolgálóhoz alapértelmezett, jelszó nélkül adminisztrátori jogosultsággal hozzá lehetett férni.

Kockázat: A hálózati nyomtatók beállításait bárki megváltoztathatja, hozzáférhet a nyomtatott dokumentumokhoz, sőt a hozzáféréshez új jelszót is megadhat, ezzel a munkavégzés és az eszközök szervizelése is nehézségekbe ütközhet.

Javaslat: A hálózati nyomtatók fiókjait az Informatikai Biztonsági Szabályzat minőségi kritériumainak megfelelő jelszóval (hossz, komplexitás, egyediség, csere gyakorisága) kell védeni, és lehetőség szerint rendszerbeállításokkal kell kikényszeríteni, időközönként cserélni.



Mit tegyünk egy ilyen hiba láttán?

Talán az egyik leggyakoribb hiba, azonban a legkönnyebben javítható is. A nyomtatók memóriájában, illetve adminisztrációs felületén olyan érzékeny információk kompromittálódhatnak, mint például scannelt bizalmas anyagok, magán céges levelezések, illetve további érzékeny információk. Minden esetben követeljük meg, hogy az üzemeltetés állítson be megfelelő jelszavakat ezen eszközökre. A beállítások néhány óra alatt elvégezhetőek a teljes hálózaton, így semmilyen többletköltséget nem ró a cégre.

1.4 NULL session LDAP autentikáció

Kategória: Jelszókezelési probléma, konfigurációs hiba

Kockázat mértéke: Közepes

Felmérési állapot: A hálózaton több DOMAIN kontroller hibás konfigurálása miatt (Pl: 172.16.1.103) lehetőség nyílik LDAP-on keresztül az AD-ből lekérdezni a DOMAIN felhasználókat.

Kockázat: Egy rosszindulatú felhasználó képes lekérdezni az Active Directoryban található összes felhasználó bejelentkezési nevét, amellyel további visszaéléseket képes elkövetni.

Javaslat: Az Active Directory megfelelő védelme, és kizárólag autentikált felhasználó legyen képes lekérdezni az LDAP-on keresztül a userek listáját.

Mit tegyünk egy ilyen hiba láttán?

Ez a hiba jelenség annyit takar, hogy megfelelő felhasználónév/jelszó nélkül tudunk érzékeny adatokat lekérdezni a központi felhasználó adatbázisért felelős szervertől. Pl: felhasználónevek, jelszóbiztonsági házirend, utolsó bejelentkezés időpontja stb. Ezen információkat felhasználva további támadások indíthatóak a hálózat többi gépe ellen. Például egy lekérdezéssel meg lehet szerezni az összes regisztrált felhasználót. Nulla költségből az átkonfigurálás megoldható, nagyban növelve hálózatunk biztonságát.