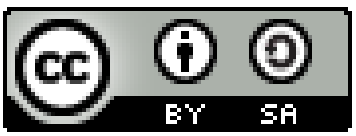
Biztonsági rések és kihasználhatóságuk

Dravecz Tibor, INTEGRITY Kft., 2019.

Tartalom



'Fix security issues correctly!'

Biztonsági rések és frissítések

Kétségkívül egyik legfontosabb feladat a **szoftverek rendszeres frissítése** (update), különösen a biztonsági javítások (patch, bugfix, illetve hotfix) gyors végrehajtása, illetve ismert rés, de szoftver fix híján a megfelelő óvintézkedések vagy kerülőmegoldások (workaround) indokolt időn belüli - adott esetben azonnali - megtétele.

Ugyanakkor sokszor nem célszerű a legfrissebb javítás alkalmazása, mert az maga is hibás lehet, vagy alkalmazása kompatibilitási problémákkal járhat. Sőt, sok esetben egy szoftveren talált biztonsági rés nem is jelent adott környezetben veszélyt, vagy a rés jelentette kockázat elhanyagolható.

Vannak rendszerek, melyeknél szinte sosem alkalmaznak biztonsági javításokat, pl. igen gyakran a folyamatosan számításokat végző szuperszámítógépeken, és vannak, ahol a legújabb update-ek alkalmazása azonnal kötelező, mint jellemzően malware védelemi (antivírus) szoftverek esetén. Másként kell alkalmazni fixeket például egy internal storage-nál, és egy felhasználói munkaállomásnál. Egy cluster esetén tipikusan egy teszt clusteren először is tesztelni kell a fixet, munkaállomásoknál ez egyrészt jellemzően indokolatlan késedelmet okozhat, másrészt jellemzően gazdaságtalan, de nem mindig és minden esetben ez a helyzet.

Se megijedni, még kevésbe kell pánikolni egy rés miatt. **A legfontosabb az, hogyha tudomást szerezzünk résről, akkor az ismertté vált rés kezelésére legyen tervünk, legyen már előzetes terv potenciális rések kezelésére.**

A legfontosabb, hogy általában a szoftverfrissítésekre előzetes tervek kellenek, és arra is, hogy mit tegyünk, ha éppen egy frissítés vezet problémához. Sőt nem csak terv, de felkészülés is szükség.

Eleve baj, ha egy biztonsági rés önmagában kihasználható, ebben az esetben szigorúan sérül 'defense in depth' elve, mely szerint **több szintű/rétegű védelem**mel kell ellátni a rendszerek (layered defense). Biztosítsuk, hogy ne legyen egy pontbeli hiba kihasználható (**Ensure no single point of vulerability**).

És persze rések lehetnek akkor is, ha azokat nem ismerjük, vagy még senki sem ismeri, és az is lehet, hogy először mindenki más előtt egy támadó fedez fel rést. Ha rendszerünk egyetlen hiba folytán támadható, akkor az nem jó.

A frissítésekről azt mondjuk, hogy korrektül kell alkalmazni azokat a biztonsági problémákra (security issue). Így szokás fogalmazni: "**Fix security issues correctly**".

A szorosan kapcsolódó fő biztonsági elvek

Fontos kérdés, hogy milyen támadási felületet nyújtunk. Ha kicsit, akkor lehet, hogy bár egy rés kihasználható, de mégsem várható, hogy azt kihasználják. A támadási felületet mindig minimalizálni kellene (**támadási felület minimalizálásának elve /principles of minimize attack surface area/**).

A támadási felület minimalizálásának elvéhez szorosan kapcsolódik a **legkisebb privilégium elve (Principle of least privilege)**, lényegben ez az elv is a támadási felület minimalizálást írja elő egyrészt, de ennél többet is, azt, hogy csak indokolt hozzáférés valósulhasson meg. Valójában nem is

privilégiumról van szó, hanem legkisebb hozzáférésről, leginkább minimalizált hozzáférési jogosultságokról.

Ezen elvekben a minimalizálás egy irányt jelent, szó sincs arról, hogy valóban a minimum elérése a tényleges cél, a legkisebb privilégium vagy legkisebb felület túlzás, a cél az ésszerűen kicsiny hozzáférés és támadási felület kialakítása.

Ha az említett elvek közt fontossági sorrendet vagy hierarchiát állítunk fel, akkor a szerző így rendezné a hierarchiát, lentről felfelé fontossági sorrendben:

Principle of Least privile privilege - Principle of Defense in depth
Principle of Minimize attack surface area
Fix security issues correctly

Azonban minden elv nagyon fontos. Inkább csak azt akartuk szemléltetni, hogy már tervezés szintjén elsőként a felső sorban szereplő elvet tartsuk mindenekelőtt szem előtt, majd állandóan törekedjünk ezeket betartani, és egyúttal minimalizálni a támadási felületet. És mindegyik elvet már tervezési fázistól szem előtt kell tartani és alkalmazni kell.

Több további fontos biztonsági elv van és a fő elvek mindig követendők, nincs kivétel, pontosabban nem szabadna kivételnek lennie.

Az alapelveket minden IT rendszerben alkalmazni kell, minden IT szakembernek ismerni kell, tovább vezetőknek is, de alapjaiban a nem IT szakember felhasználóknak is.

Biztonsági káresemények gyökere

A biztonsági problémák gyökere nem a biztonsági hibákban van, nem is azok nem kijavításában, hanem ennél mélyebbről ered.

Először az ún. Network Time Protocol (NTP), pontosabban az ún. NTP szoftver implementáció sérülékenységről szólunk, melynél bár szoftver sérülékenység kellett, hogy a probléma jelentkezzen, önmagában a sérülékenység igen kevés szolgáltatónál szabadott volna, hogy gondot okozzon[1].

A többrétegű védelem vagy a támadó felület minimalizálásának elvének követése mellett a rendszerek döntő részében a szoftverhiba semmilyen kárhoz nem vezetethetett volna. Még inkább nem, ha minden elv be lett volna tartva.

Szarvashibák vezettek problémához, nem csupán egy szoftver hibája. A szoftver hibáját a rendszerek döntő részében nem szabadott volna tudni támadóknak kihasználni. Tudatlanság, gondatlanság, nemtörődömség vezetett valójában gondokhoz.

[1] nagy publikus NTP szolgáltatók számára nyilván igen, és rajtuk keresztül másoknak is, de valójában a gondok minimális része kötődött a nagy publikus NTP szolgáltatókhoz

"Semmi sem rettenetesebb, mint a tudatlanság megnyilvánulása."[2]
- Johann Wolfgang von Goethe

Szarvashibák - az NTP sérülékenység példája

A Network Time Protocol (NTP) első RFC szabványa[3] 1985-ben jelenet meg. 2014-ig bár számos kisebb hibát felfedeztek implementációiban, illetve magában a protokollban, az első komoly problémákat 2014-ben jelentették.

2014-ben nem csak NTP sérülékenység (vulnerability) vált ismerté, de sérülékenységet kihasználva erősítéses DDoS támadásra lehet kihasználni NTP szervereket. 2014-ben a második legjelentősebb DDoS támadási módszerré vált az NTP sérülékenység kihasználása.

A felismert sérülékenységre hamar megjelent szoftver update, és az egész világon azonnal tanácsolták a frissítést.

Noha a frissítés indokolt, a tanács alapvetően nem jó!

Mindenekelőtt azt kellett volna és kell ma is tanácsolni, hogy NTP szolgáltatást korlátozzuk le, ne nyissuk meg a szolgáltatást a nagyvilág felé.

Legtöbbek teljesen feleslegesen hagyták nyitva az NTP portot a nagyvilág felé, mert senkinek nem akartak NTP szolgáltatást nyújtani.

NTP szolgáltatást egy maroknyi szolgáltatón kívül senkinek nem kellene publikusan szolgáltatni. Ha egy szervezet magának vagy partnereinek ilyet szolgáltatat, akkor a szolgáltatást le kell korlátozni, hogy csak azok a hosztok érjék el a szolgáltatást, akiknek valóban erre szükségük van.

A korlátozás egyszerű, vagy külső hálózat felé egyáltalán nem nyitjuk meg a szolgáltatást, vagy lekorlátozzuk azon hosztokra a szolgáltatást, melyeket ki akarunk szolgálni.

Biztonsági rések jelentős részében nem az a fő baj, hogy nincs frissítve a rendszer, hanem az, hogy a hiba kihasználására alkalmat sem szabadott volna adni. Azaz, frissítéstől függetlenül a rendszernek nem szabadna sebezhetőnek lennie.

Biztonsági hibás szoftver frissítése az elmondottaktól függetlenül még nagyon fontos.

A NTP szolgáltatás nem korlátozása vagy felesleges nyújtása két alapvető biztonsági alapelv megsértését jelenti, az alábbiakét:

- **többrétegű védelem alkalmazásásnak elve** (**Defense in Depth, Layered security**).
- **támadási felület minimalizálásának elve** (P**rinciple of Minimize attack surface area**).

Tegyük azt, amit kell

- egyrészt **minimalizáljuk a támadási felületet**,
- másrészt **több szintű védelmet alkalmazzunk**, egyetlen hiba ne vezethessen még bajhoz!

[2] "Es ist nichts schrecklicher als eine tätige Unwissenheit." - Johann Wolfgang von Goethe
[3] legfrissebb szabványa: Network Time Protocol Version 4: Protocol and Algorithms Specification, RFC 5905

"Tudatlanság - minden gonosz gyökere és szára." - Platón

További példák szarvashibákra

SQL injection

Ha egy weboldalunkra egy keresőt teszünk ki, és hiba folytán SQL injection támadást válik végrehajthatóvá, a támadást nem hajtja végre senki, ha az oldalhoz a hozzáférést csak arra jogosított felhasználókra szűkítjük el.

Szintén szűkítjük a támadási felületet, de egyben egy védelemi falat is húzunk, ha webalkalmazás tűzfalat is alkalmazunk. A védelemnek sok más szintje és megoldása van, így az ellenőrzés is, mind kódszintű audit, mind behatolási tesztelés.

Zsarolóvírusok (Ransomware)

Zsarolóvírus sokkal nagyobb kárt okozhat, ha felhasználóink hozzáférhetnek írásjoggal olyan file-okhoz is, melyet csak olvasniuk elég lenne, vagy hozzáférhetnek olyan file-okhoz, melyet egyáltalán nem szükséges elérniük.

A védelem egy mélyebb szintje, hogy megfelelő, többszintű, megfelelően védett, megfelelően megőrzött mentést alkalmazunk. A védelem sok más-más faktora, szintje lehet, sok más védelmi megoldást alkalmazhatunk párhuzamosan. Ez igaz az SQL injection támadásra, a zsarolóvírusok elleni védelemre, és minden más biztonsági veszélyforrásra.

SQL injectionra csak hozzá nem értés esetén adódik lehetőség. Aki tud programozni, olyan technikát használ, mely mellett ilyen hibát véteni sem tud. Ha persze ha nem hozzáértő, hanem gyányoló csinálta a programot, akkor adódhat ilyen.

Sajnos gányolók programjait lépten-nyomon használjuk, már csak ezért is szükség van arra, hogy ahol ne bízzunk meg programjainkban. Biztonsági hibát azonban még a legjobb szoftveres fejlesztőcsapatok is követnek el.

- Egyrészt **szűkítsük a támadási felületet**,
- másrészt **több védelmi megoldást alkalmazzunk, több szinten párhuzamosan.**

- Trehányság nem szűkíteni a támadási felületet,
- trehányság nem alkalmazni többszintű védelmet.

- Falazzuk be a felesleges hátsó ajtót,
- és szereljünk fel a bejárati ajtónkra plusz biztonsági zárát.

- Trehányság nem befalazni a felesleges bejáratot,
- trehányság nem alkalmazni második zárat. Ugyanakkor mindkettő plusz költség, valamint kényelmetlenség is.

Butaság nem tudni azt, hogy van biztonságosabb zár, vagy nem tudni arról, hogy egy ajtóra két zár is szerelhető.

"Nem létezik tökéletes védelem, csak különböző szintű védetlenség."[4]
- Salman Rushdie

Nincs tökéletes védekezés, de hathatós védelem általában lehet

- Hathatós védelemre kell törekedni'
- Szarvashibákat megfelelő ismeret, gondosság, ellenőrzés mellett el lehet kerülni.
- Tervezetlen vagy dokumentálatlan, illetve nem gondosan tervezett vagy gyengén dokumentált rendszerek a biztonsági káresemények forrása.
- A káresetek döntő részében a fő biztonsági alapelvek megsértése történt.
- Ismerjük és kövessük az alapvető biztonsági elveket! Követeljük meg ezen elvek ismeretét és betartását! - Tartsuk be és tartassuk be!
- Egyszintű védelem szinte sosem elégséges. Egyszintű védelmet legfejlebb akkor alkalmazunk, ha nincs igazán mit és kit védeni.
    - Sokszor azonban sajnos csak azt gondoljuk, hogy nincs kit vagy mit védeni!
- Egy pontbeli hibaforrásokat nem szabad megengedni!
- Nem csak magunkat, ügyfeleinket kell védeni, hanem másnak sem szabad kárt okoznunk, nem szabad engedni, hogy rajtunk keresztül másnak kárt okozhassanak! Felelősek azok, aki gondatlanul lehetővé teszik, hogy rendszereink, erőforrásaink felhasználásával másnak kárt okozzanak. A felelősség nem csak erkölcsi, de jogi is.

[4] "There is no such thing as perfect security, only varying levels of insecurity." - Salman Rushdie

Ajánlott irodalom

Az alábbiakban olyan alapvető ismeretekről adunk meg irodalmat, mely minden informatikus számára alapvető:

Ezen anyag a fő biztonsági elveket ismerteti, jól összefoglalja:
Engineering Principles for Information Technology Security (A Baseline for Achieving Security), Revision A
NIST Special Publication 800-27 Rev A, 2017
https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-27ra.pdf

Az anyag informatikai mércével régi, de az alapelvek nemigen változnak:
Information Security Handbook: A Guide for Managers Recommendations of the National Institute of Standards and Technology
NIST Special Publication 800-100, 2006
https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-100.pdf

Fő biztonsági elvek rövid összefoglalása:
OWASP Foundation: Security by Desing Principles
https://www.owasp.org/index.php/Security_by_Design_Principles

Wikipedia: Vulnerability (computing)
https://en.wikipedia.org/wiki/Vulnerability_(computing)

Wikipedia: Patch (computing)
https://en.wikipedia.org/wiki/Patch_(computing)

Wikipedia: Hotfix
https://en.wikipedia.org/wiki/Hotfix

Wikipedia: Information assurance
https://en.wikipedia.org/wiki/Information_assurance

Wikipedia: Defense in depth (computing)
https://en.wikipedia.org/wiki/Defense_in_depth_(computing)

Wikipedia: Layered security
https://en.wikipedia.org/wiki/Layered_security

Wikipedia: Single point of failure
https://en.wikipedia.org/wiki/Single_point_of_failure

Az NTP sérülékenységről:

Serious NTP security holes have appeared and are being exploited
Steven J. Vaughan-Nichols,2014
https://www.zdnet.com/article/major-ntp-security-holes-appears-and-are-being-exploited/

SC-CERT ICSA-14-353-01B
Advisory (ICSA-14-353-01C)
Network Time Protocol Vulnerabilities (Update C)
Original release date: February 05, 2015 | Last revised: August 29, 2018
https://ics-cert.us-cert.gov/advisories/ICSA-14-353-01C

ASERT Threat Intelligence Brief 2014-1
Mitigating NTP Reflection/Amplification DDoS Attacks
ASERTThreat Intelligence, January 201
http://pages.arbornetworks.com/rs/arbor/images/ASERT%20Threat%20Intelligence%20Brief%202014-01%20NTP%20Amplification%20Attacks.pdf

SQL injectionról:

Wikipedia: SQL injection
https://en.wikipedia.org/wiki/SQL_injection

Kulcsszavak: biztonsági elvek, legkisebb privilégium elve, támadási felület minimalizálása, többszintű védelem, biztonsági rések, biztonsági frissítések, Network Time Protocol (NTP);

Kewords: security principles, Principle of Least privile privilege (PoLP), Minimize attack surface area, Defense in Depth, Layered security, Single Point of Failure (SPOF), fix, patch, hotfix, update, Network Time Protocol (NTP).