

# **Lecture 10 – Supply Chain Security**

[COSE451] Software Security

Instructor: Seunghoon Woo

Spring 2024

# Overview

- **Supply chain security**
- **Software Bill of Materials (SBOM)**

# Supply chain security

- **Supply chain security**



Secure coding



Vulnerability detection



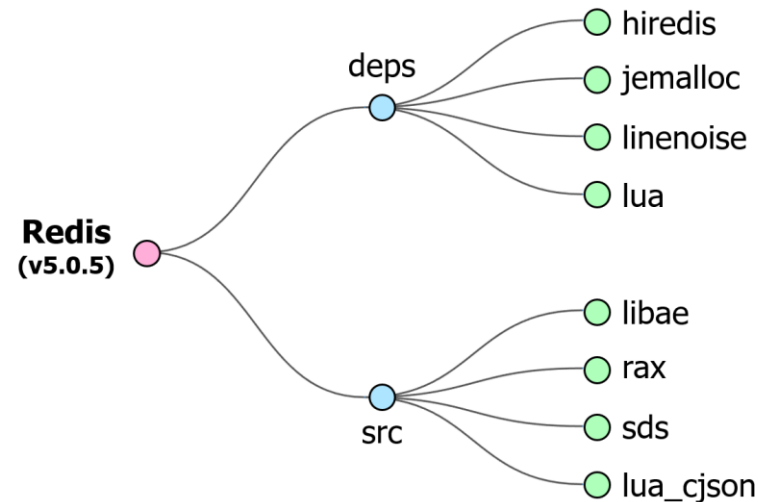
DevSecOps

...

# Software Bill of Materials (SBOM)

- **Software Bill of Materials (SBOM)**

- A type of metadata that describes and lists all SW components
- Various standards: SPDX, CycloneDX, SWID



# Softw

# (BOM)

```
{
  "bomFormat": "CycloneDX",
  "specVersion": "1.5",
  "serialNumber": "...",
  "version": 1,
  "metadata": {
    "timestamp": "2024-05-23T06:44:11Z",
    "tools": "...",
    "authors": "...",
    "component": {
      "name": "redis",
      "type": "application",
      "bom-ref": "root"
    }
  },
  "components": [
    {
      "name": "hiredis",
      "version": "v1.1.0",
      "licenses": [
        {
          "license": {
            "id": "BSD-3-Clause",
            "url": "https://spdx.org/licenses/BSD-3-Clause.html"
          }
        }
      ],
      "purl": "pkg:github/redis/hiredis@v1.1.0",
      "type": "library",
      "bom-ref": "4947|659035"
    },
    {
      "name": "lua",
      "version": "v5.1.1",
```

# Software Bill of Materials (SBOM)

- **Software Bill of Materials (SBOM)**

- SBOM can be used when developing, purchasing, or operating a system, etc.

- Benefits

- [Developers]

- They often create products using components of public SW (OSS) and third-party SW

- The SW development company can identify whether the corresponding component is the **latest version** through SBOM and **quickly respond to new security vulnerabilities**

- [SW Buyers & Users]

- They can use SBOM to **transparently perform security vulnerability or license analysis**, and these two analyzes can be used to evaluate the overall risk level of the product

# Software Bill of Materials (SBOM)

- **Minimum Requirements**

- The minimum standard for global supply is presented by the U.S. NTIA

Element	Description
Supplier Name	The name of the entity that created, defined, and identified the component.
Timestamp	Record of date and time of conversion to SBOM data
Author Name	Name of the entity generating SBOM data for the component.
Component Name	Name assigned to the SW unit as defined by the original supplier.
Version String	Identifier used to specify changes in SW from a previous version.
Unique Identifier	Identifier used to identify components or serve as look-up keys in related DBs.
Relationship	Specifies the relationship that upstream component.

# Software Bill of Materials (SBOM)

- **Vulnerability examination**

- Both version-based and code-based vulnerability detection techniques can be linked to SBOM!



# Software Bill of Materials (SBOM)

- **Vulnerability examination: Version-based approach**

```
"components":  
[  
  {  
    "name": "hiredis",  
    "version": "v1.1.0",  
    "licenses":  
    [  
      {  
        "license":  
        {  
          "id": "BSD-3-Clause",  
          "url": "https://spdx.org/licenses/BSD-3-Clause.html"  
        }  
      }  
    ],  
    "purl": "pkg:github/redis/hiredis@v1.1.0",  
    "type": "library",  
    "bom-ref": "4947|659035"  
  },  
  {  
    "name": "lua",  
    "version": "v5.1.1",
```

# Software Bill of Materials (SBOM)

- **Vulnerability examination: Code-based approach**
  - During the SBOM creation process, code-based vulnerability information is often recorded together

Soft

PM)

• Vulnerability

- During the
- often rec

```

"vulnerabilities":
[
  {
    "bom-ref": "054F26F0-3B28-7002-0000-000000000000|BCA423CCCC4949785C804",
    "id": "CVE-2020-14147",
    "source":
    {
      "name": "NVD",
      "url": "https://nvd.nist.gov/vuln/detail/CVE-2020-14147"
    },
    "references":
    [
      {
        "id": "CVE-2020-14147",
        "source":
        {
          "name": "NVD",
          "url": "https://nvd.nist.gov/vuln/detail/CVE-2020-14147"
        }
      }
    ],
    "ratings":
    [
      {
        "source":
        {
          "name": "NVD",
          "url": "https://nvd.nist.gov/vuln-metrics/cvss/v2-calculat
        },
        "score": 4,
        "method": "CVSSv2",
        "vector": "AV:N/AC:L/Au:S/C:N/I:N/A:P"
      }
    ]
  }
]

```

Information is

# Software Bill of Materials (SBOM)

- **License comparability**

- Because SBOM also contains component license information, it can also respond to license comparability/violation issues

```
"components":  
[  
  {  
    "name": "hiredis",  
    "version": "v1.1.0",  
    "licenses":  
    [  
      {  
        "license":  
        {  
          "id": "BSD-3-Clause",  
          "url": "https://spdx.org/licenses/BSD-3-Clause.html"  
        }  
      }  
    ]  
  }  
]
```

# Software Bill of Materials (SBOM)

```
"SPDXID" : "SPDXRef-Package-161423",
"copyrightText" : "NOASSERTION",
"downloadLocation" : "https://www.kernel.org/pub/linux/kernel/v2.6",
"filesAnalyzed" : false,
"homepage" : "https://www.kernel.org",
"licenseConcluded" : "GPL-2.0",
"licenseDeclared" : "GPL-2.0",
"licenseInfoFromFiles" : [ "GPL-2.0" ],
"name" : "Linux Kernel",
"originator" : "Organization: \"\"",
"supplier" : "Person: \"\"",
"versionInfo" : "2.6"
}, {
  "SPDXID" : "SPDXRef-Package-161422",
  "copyrightText" : "Bite Chalk is free for personal use only. Please, talk with the author for",
  "downloadLocation" : "https://www.maisfontes.com/bite-chalk-normal.font",
  "filesAnalyzed" : false,
  "homepage" : "https://www.maisfontes.com/bite-chalk.font",
  "licenseConcluded" : "LicenseRef-Non-Commercial-Use-Only-Font-License",
  "licenseDeclared" : "LicenseRef-Non-Commercial-Use-Only-Font-License",
  "licenseInfoFromFiles" : [ "LicenseRef-Non-Commercial-Use-Only-Font-License" ],
  "name" : "BiteChalk Font",
  "originator" : "Organization: \"\"",
  "supplier" : "Person: \"\"",
  "versionInfo" : ""
} ],
```

# Software Bill of Materials (SBOM)

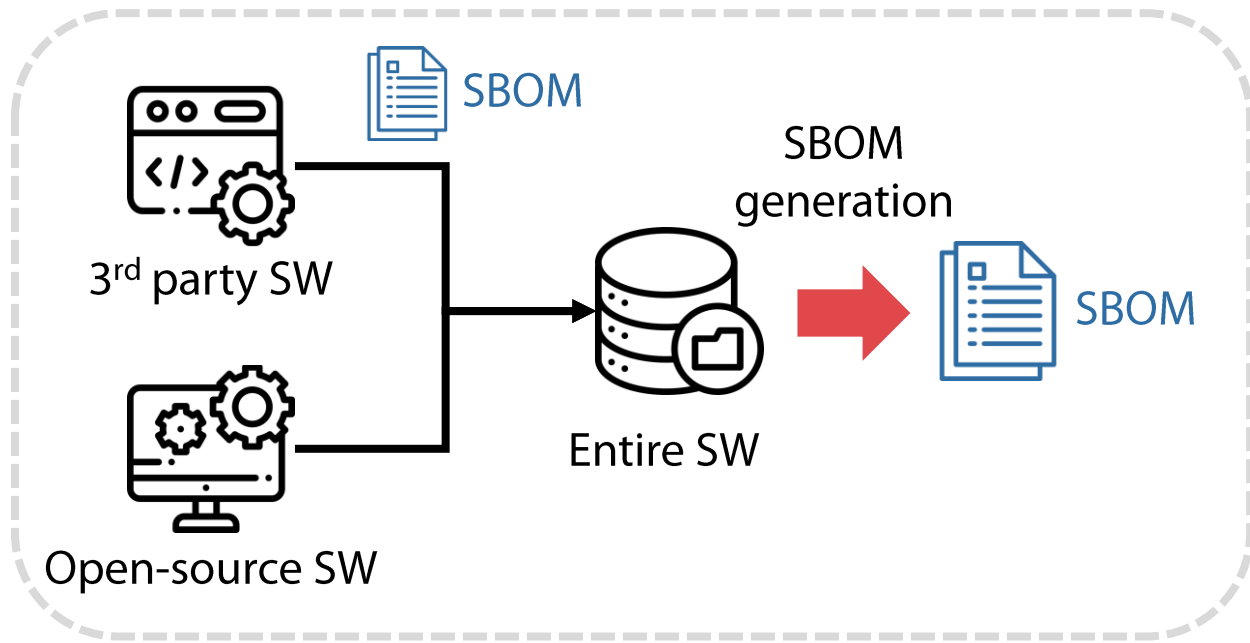
- **How to generate SBOM?**
  1. Take advantage of automated tools
  2. Manual examination..

# Software Bill of Materials (SBOM)

- **SBOM distribution**

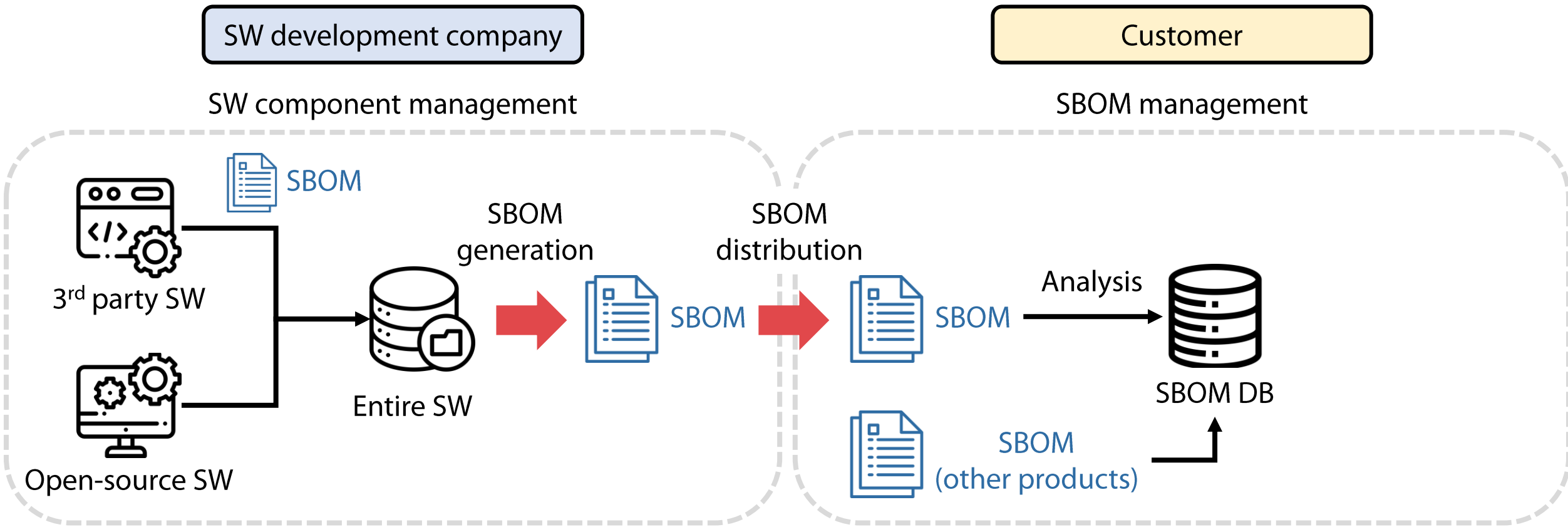
SW development company

SW component management



# Software Bill of Materials (SBOM)

## • SBOM distribution





# Software Bill of Materials (SBOM)

## • References

- <https://www.federalregister.gov/documents/2021/05/17/2021-10460/improving-the-nations-cybersecurity>
  - (EN) Executive Order 14028: Improving the Nation's Cybersecurity
- <https://www.kisa.or.kr/2060204/form?postSeq=15&page=1>
  - (KR) 소프트웨어 공급망 보안 가이드라인 1.0

# Next Lecture

- **Defense strategies**

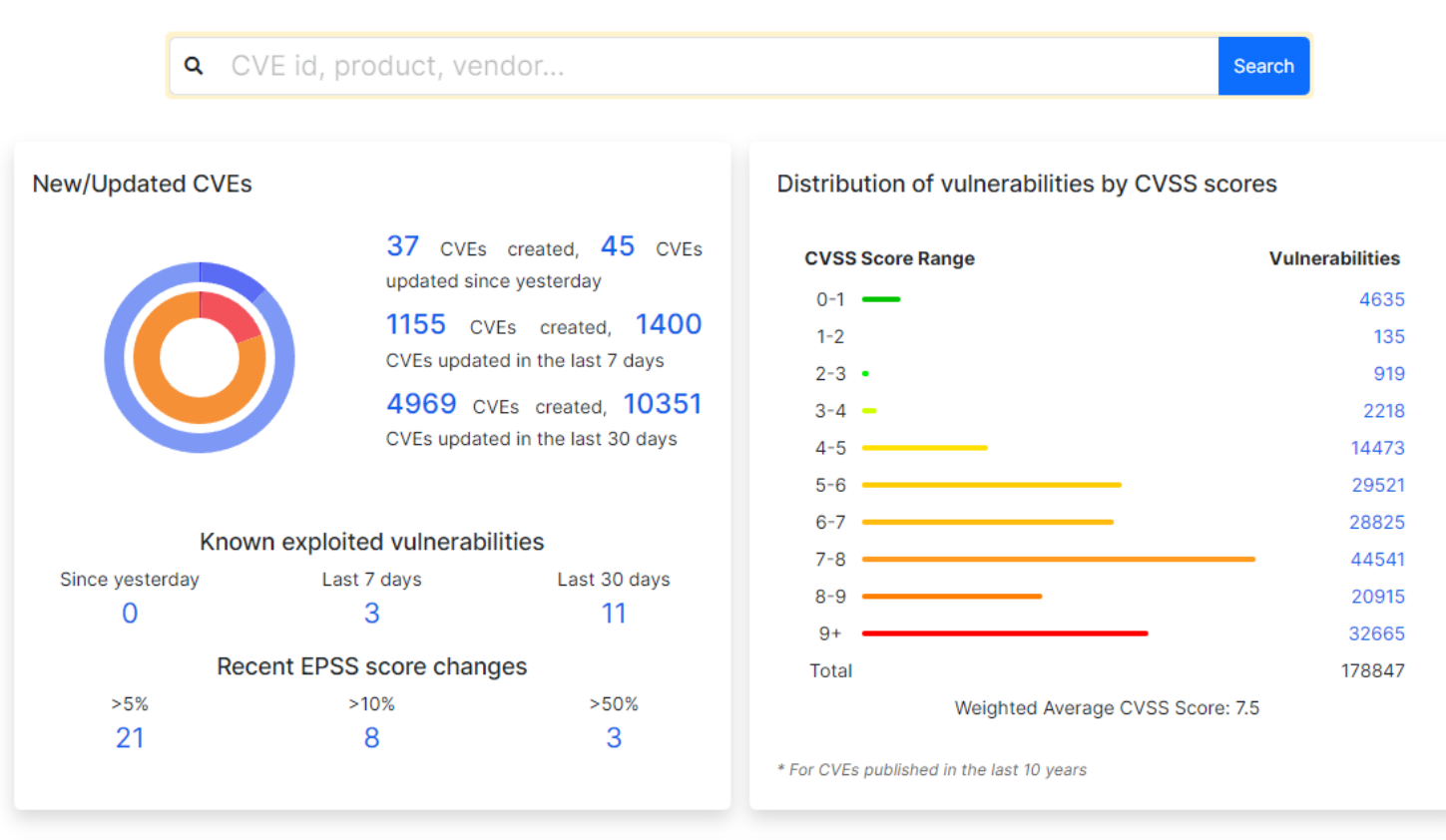
# Assignment II(Edited)

- **Due date:** June ~~14<sup>th</sup>~~ 19<sup>th</sup> 11:59 PM
- **To be submitted:**
  - Please compress the following three items into a single file (.zip) and submit it
    1. Source code with vulnerabilities
      - For real-world OSS, only submit the files containing vulnerabilities
    2. Source code with patches applied
    3. Report
      - As shown in this material, you must include the following information (e.g., using screenshots)
        - where the vulnerability was located
        - how the vulnerability was triggered
        - how it was patched
        - confirmation that the vulnerability is no longer triggered

# Assignment II(Appendix)

## • CVEdetails

- CVEdetails.com  
powered by SecurityScorecard
- ▼ Vulnerabilities
  - By Date
  - By Type
  - Known Exploited
  - Assigners
  - CVSS Scores
  - EPSS Scores
  - Search
- ▼ Vulnerable Software
  - Vendors
  - Products
  - Version Search
- ▼ Vulnerability Intel.
  - Newsfeed
  - Open Source Vulns
  - Emerging CVEs
  - Feeds
  - Exploits
  - Advisories
  - Code Repositories
  - Code Changes



<https://www.cvedetails.com/>

# Assignment II(Appendix)

## • CVEdetails

**CVEdetails.com**  
powered by SecurityScorecard

- ▼ Vulnerabilities
  - By Date
  - By Type
  - Known Exploited
  - Assigners
  - CVSS Scores
  - EPSS Scores
  - Search
- ▼ **Vulnerable Software**
  - Vendors
  - Products
  - Version Search
- ▼ Vulnerability Intel.
  - Newsfeed
  - Open Source Vulns
  - Emerging CVEs
  - Feeds
  - Exploits
  - Advisories
  - Code Repositories
  - Code Changes

Search CVE id, product, vendor... Search

### New/Updated CVEs

**37** CVEs created, **45** CVEs updated since yesterday  
**1155** CVEs created, **1400** CVEs updated in the last 7 days  
**4969** CVEs created, **10351** CVEs updated in the last 30 days

### Known exploited vulnerabilities

Since yesterday	Last 7 days	Last 30 days
0	3	11

### Recent EPSS score changes

>5%	>10%	>50%
21	8	3

### Distribution of vulnerabilities by CVSS scores

CVSS Score Range	Vulnerabilities
0-1	4635
1-2	135
2-3	919
3-4	2218
4-5	14473
5-6	29521
6-7	28825
7-8	44541
8-9	20915
9+	32665
Total	178847

Weighted Average CVSS Score: 7.5

\* For CVEs published in the last 10 years

# Assignment II(Appendix)

## • CVEdetails

The screenshot displays the CVEdetails.com website interface. On the left, a navigation menu is visible, with the 'Vulnerable Software' section highlighted in a red box. This section includes links for 'Vendors', 'Products', and 'Version Search'. The main content area features a search bar at the top with the placeholder text 'CVE id, product, vendor...'. Below the search bar, a modal window titled 'Vendor, Product, Version Search' is open, containing three input fields: 'Vendor Name' (with the example 'e.g Apache or apac\*'), 'Product Name' (with the value 'Redis'), and 'Version' (with the value '5.0.5'). A blue 'Search' button is located at the bottom of this modal. In the background, a chart titled 'Recent EPSS score changes' shows data for 'Since yesterday' (0), '>5%' (21), '>10%' (8), and '>50%' (3). To the right, a table titled 'Vulnerabilities by CVSS scores' lists various scores and their corresponding counts, with a total of 178847. The weighted average CVSS score is noted as 7.5. A footer note states '\* For CVEs published in the last 10 years'.

**Vulnerable Software**

- Vendors
- Products
- Version Search

**Vendor, Product, Version Search**

Vendor Name:

Product Name:

Version:

**Recent EPSS score changes**

Category	Count
Since yesterday	0
>5%	21
>10%	8
>50%	3

**Vulnerabilities by CVSS scores**

Score	Count
4635	4635
135	135
919	919
2218	2218
14473	14473
29521	29521
28825	28825
44541	44541
20915	20915
32665	32665
<b>Total</b>	<b>178847</b>

Weighted Average CVSS Score: 7.5

\* For CVEs published in the last 10 years

<https://www.cvedetails.com/>

# Assignment II(Appendix)

## • CVEdetails

<b>CVE-2021-3470</b> A heap overflow issue was found in Redis in versions before 5.0.10, before 6.0.9 and before 6.2.0 when using a heap allocator other than jemalloc or glibc's malloc, leading to potential out of bound write or process crash. Effectively this flaw does not affect the vast majority of users, who use jemalloc or glibc malloc. Source: Red Hat, Inc.	Max CVSS EPSS Score Published Updated	5.3 0.09% 2021-03-31 2021-04-05
<b>CVE-2020-14147</b> An integer overflow in the getnum function in lua_struct.c in Redis before 6.0.3 allows context-dependent attackers with permission to run Lua code in a Redis session to cause a denial of service (memory corruption and application crash) or possibly bypass intended sandbox restrictions via a large number, which triggers a stack-based buffer overflow. NOTE: this issue exists because of a CVE-2015-8080 regression. Source: MITRE	Max CVSS EPSS Score Published Updated	7.7 0.31% 2020-06-15 2021-07-30
<b>CVE-2015-8080</b> Integer overflow in the getnum function in lua_struct.c in Redis 2.8.x before 2.8.24 and 3.0.x before 3.0.6 allows context-dependent attackers with permission to run Lua code in a Redis session to cause a denial of service (memory corruption and application crash) or possibly bypass intended sandbox restrictions via a large number, which triggers a stack-based buffer overflow. Source: MITRE	Max CVSS EPSS Score Published Updated	7.5 82.72% 2016-04-13 2022-08-05