Lecture 8 – Open-source Software Security

[COSE451] Software Security

Instructor: Seunghoon Woo

Spring 2024

Overview

- **Open-source software & Licenses**
- **Vulnerabilities in OSS**

Common Vulnerabilities and Exposures

- **Common Vulnerabilities and Exposures (CVE)**
  - A standardized identifier system for tracking and identifying vulnerabilities
    - Discovered in computer systems, networks, software, hardware, and other technologies
  - The CVE program is overseen by the MITRE Corporation
    - https://cve.mitre.org/
  - Currently, approximately 200,000 CVE vulnerabilities are being managed
  - CVE-YEAR-NUMBER
    - E.g., CVE-2016-5195

Common Vulnerabilities and Exposures

- **Common Vulnerabilities and Exposures (CVE)**

Field	Explanation
CVE ID	A vulnerability unique identifier assigned by the MITRE corporation
Descriptions	A summary of the overall introduction of each vulnerability, including affected products and attack vectors
Severity	An indicator that represents the severity of a vulnerability
Types	A value indicating the type of vulnerability, such as buffer overflow or out-of-bounds read/write
Affected software configurations	The name and version information of the software that are affected by the vulnerability
References	A set of reference links related to the vulnerability

Common Vulnerabilities and Exposures

CVE-2016-5195 Detail

Description

Race condition in mm/gup.c in the Linux kernel 2.x through 4.x before 4.8.3 allows local users to gain privileges by leveraging incorrect handling of a copy-on-write (COW) feature to write to a read-only memory mapping, as exploited in the wild in October 2016, aka "Dirty COW."

Severity

CVSS Version 3.x | CVSS Version 2.0

CVSS 3.x Severity and Metrics:

NIST: NVD **Base Score:** 7.8 HIGH **Vector:** CVSS:3.1/AV:L/AC:L/PR:L/UI:N/S:U/C:H/I:H/A:H

Weakness Enumeration

CWE-ID	CWE Name	Source
CWE-362	Concurrent Execution using Shared Resource with Improper Synchronization ('Race Condition')	NIST

Known Affected Software Configurations Switch to CPE 2.2

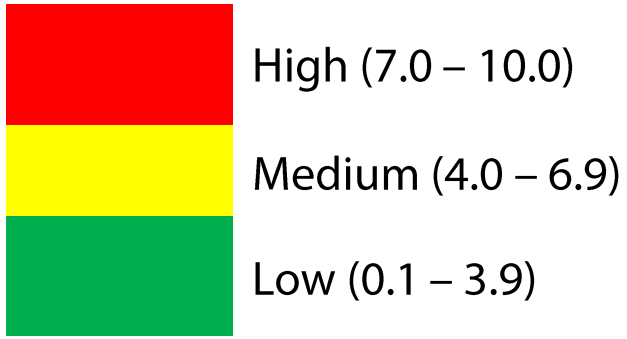
cpe:2.3:o:linux:linux_kernel:*:*:*:*:*:*:*:*	From (including)	Up to (excluding)
Show Matching CPE(s)	2.6.22	3.2.83
cpe:2.3:o:linux:linux_kernel:*:*:*:*:*:*:*:*	From (including)	Up to (excluding)
Show Matching CPE(s)	3.3	3.4.113

References to Advisories, Solutions, and Tools

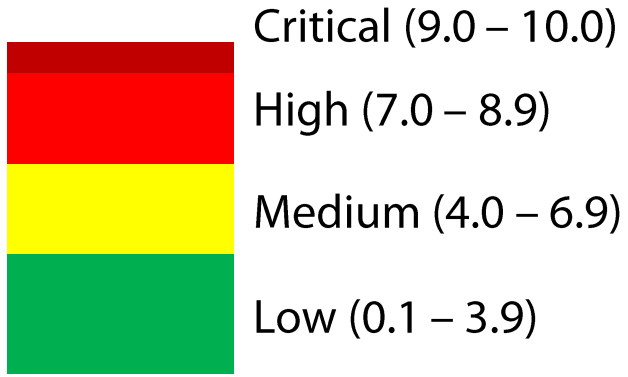
Hyperlink	Resource
http://git.kernel.org/cgit/linux/kernel/git/torvalds/linux.git/commit/?id=19be0eaffa3ac7d8eb6784ad9bdbc7d67ed8e619	Issue Tracking, Patch, Vendor Advisory

Common Vulnerabilities and Exposures

- **Common Vulnerability Scoring System (CVSS)**
  - An open standard for assigning scores to rate the severity of vulnerabilities
  - Scores are given from 0.0 to 10.0
    - Higher numbers indicating greater severity of the vulnerability



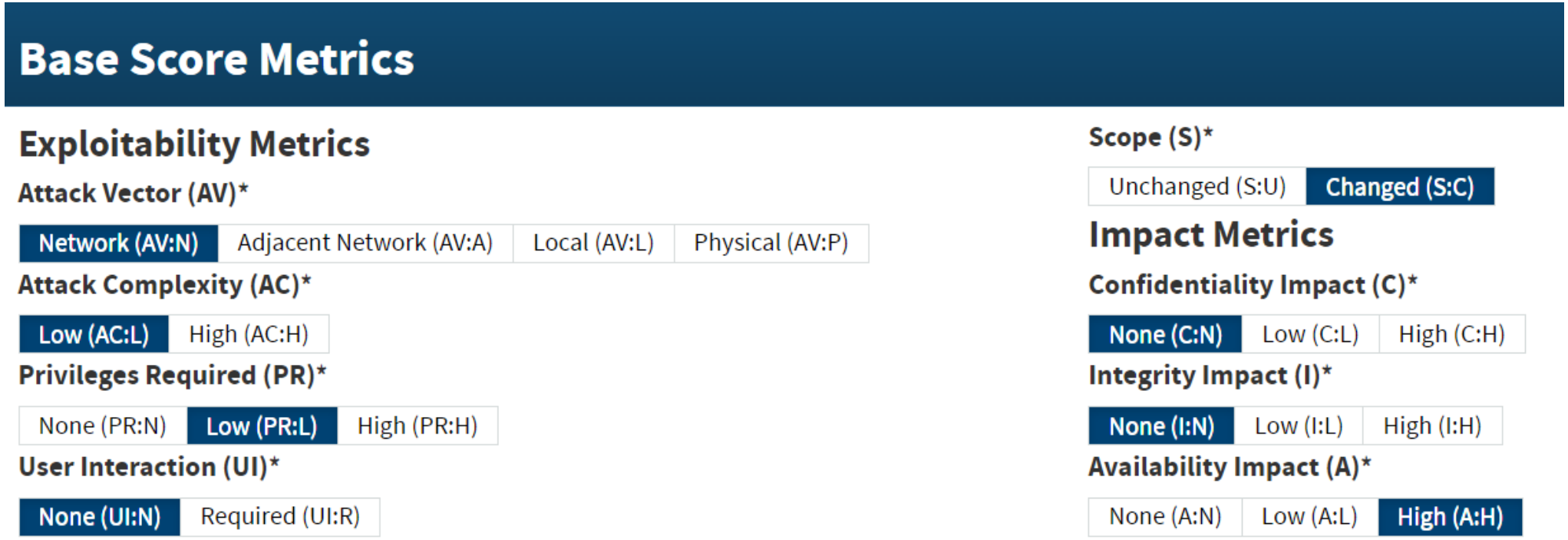
CVSS v2



CVSS v3

Common Vulnerabilities and Exposures

- **Common Vulnerability Scoring System (CVSS)**
  - Measurement standard



Common Vulnerabilities and Exposures

- **Common Weakness Enumeration (CWE)**
  - A list of common software and hardware weakness types
  - Approximately 1,000 CWEs are defined
    - Stack buffer overflow, integer overflow, race condition, out-of-bounds read, etc.
    - https://cwe.mitre.org/data/published/cwe_latest.pdf

Common Vulnerabilities and Exposures

- **Common Platform Enumeration (CPE)**
  - A structured naming scheme for information technology systems and software
    - More than 10,000 software/hardware are defined as CPE
  - Used to specify the affected software (including OSS) by vulnerabilities

Common Vulnerabilities and Exposures

CVE-2016-5195 Detail

Known Affected Software Configurations Switch to CPE 2.2

cpe:2.3:o:linux:linux_kernel:*:*:*:*:*:*:*:* Show Matching CPE(s)	From (including) 2.6.22	Up to (excluding) 3.2.83
cpe:2.3:o:linux:linux_kernel:*:*:*:*:*:*:*:* Show Matching CPE(s)	From (including) 3.3	Up to (excluding) 3.4.113
cpe:2.3:o:linux:linux_kernel:*:*:*:*:*:*:*:* Show Matching CPE(s)	From (including) 3.5	Up to (excluding) 3.10.104
cpe:2.3:o:linux:linux_kernel:*:*:*:*:*:*:*:* Show Matching CPE(s)	From (including) 3.11	Up to (excluding) 3.12.66
cpe:2.3:o:linux:linux_kernel:*:*:*:*:*:*:*:* Show Matching CPE(s)	From (including) 3.13	Up to (excluding) 3.16.38
cpe:2.3:o:linux:linux_kernel:*:*:*:*:*:*:*:* Show Matching CPE(s)	From (including) 3.17	Up to (excluding) 3.18.44
cpe:2.3:o:linux:linux_kernel:*:*:*:*:*:*:*:* Show Matching CPE(s)	From (including) 3.19	Up to (excluding) 4.1.35
cpe:2.3:o:linux:linux_kernel:*:*:*:*:*:*:*:* Show Matching CPE(s)	From (including) 4.2	Up to (excluding) 4.4.26
cpe:2.3:o:linux:linux_kernel:*:*:*:*:*:*:*:* Show Matching CPE(s)	From (including) 4.5	Up to (excluding) 4.7.9
cpe:2.3:o:linux:linux_kernel:*:*:*:*:*:*:*:* Show Matching CPE(s)	From (including) 4.8	Up to (excluding) 4.8.3

Common Vulnerabilities and Exposures

- **Security patch**
  - Code patches containing information to fix vulnerable code
  - In general, security patches are provided in the form of "`diff`" of a code before and after applying the patch

```
diff --git a/tensorflow/core/ops/array_ops.cc b/tensorflow/core/ops/array_ops.cc
index 64bd4f38478542..14c9efae1ddd3b 100644
--- a/tensorflow/core/ops/array_ops.cc
+++ b/tensorflow/core/ops/array_ops.cc
@@ -168,7 +168,7 @@ Status TransposeShapeFn(InferenceContext* c) {

     for (int32_t i = 0; i < rank; ++i) {
       int64_t in_idx = data[i];
-      if (in_idx >= rank) {
+      if (in_idx >= rank || in_idx <= -rank) {
         return errors::InvalidArgument("perm dim ", in_idx,
                                        " is out of range of input rank ", rank);
       }
```

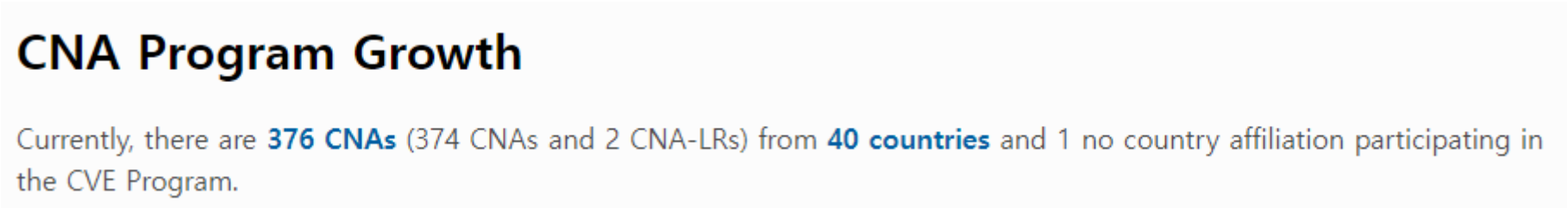
CVE reporting process

1. **Discovery**
   - Discovering vulnerabilities
     - Using static/dynamic analysis
       - These will be covered in more detail in later lectures
     - Penetration testing
   - Cause analysis and remediations must also be confirmed at this stage

CVE reporting process

2. Report submission

- Needs to be reported to a CVE Numbering Authority (CNA)
  - CNA: organizations authorized by the CVE Program to assign CVE IDs to vulnerabilities



CVE reporting process

2. **Report sul**
   - Needs to be
     - CNA: org)s to vulnerabilities

Partner	Scope	Program Role	Organization Type	Country*
Financial Security Institute (FSI)	Vulnerability assignment related to FSI's vulnerability coordination role in the South Korea financial sector that are not in another CNA's scope	CNA	CERT, Researcher, Bug Bounty Provider	South Korea
Hanwha Vision Co., Ltd.	Hanwha Vision (formerly Samsung Techwin and Hanwha Techwin) products and solutions only, including end-of-life (EOL)	CNA	Vendor	South Korea
KrCERT/CC	Vulnerability assignment related to its vulnerability coordination role	CNA	CERT	South Korea
LG Electronics	LG Electronics products only	CNA	Vendor	South Korea
Naver Corporation	Naver products only, except Line products	CNA	Vendor	South Korea
Samsung Mobile	Samsung Mobile Galaxy products, personal computers, and related services only	CNA	Vendor	South Korea
Samsung TV & Appliance	Samsung TV & Appliance products, Samsung-owned open-source projects listed on	CNA	Open Source, Researcher, Vendor	South Korea

CVE reporting process

2. Report submission

- Needs to be reported to a CVE Numbering Authority (CNA)
  - CNA: organizations authorized by the CVE Program to assign CVE IDs to vulnerabilities
- A variety of information must be reported together
  - Description, affected products, type, Proof of Concept (PoC), mitigation recommendations, security patches, etc.

CVE reporting process

3. CVE ID assignment

- The CNA reviews the vulnerability report and assigns a unique CVE ID to it
  - Only if it meets the CVE program's criteria
- Thereafter, CNA creates a CVE entry for the vulnerability
  - This entry includes detailed information about the vulnerability such as its description, affected products, severity level, and any relevant references

Response to vulnerabilities using CVE

- **Two types of vulnerability**
  - Zero-day (0-day)
  - One-day (1-day)

Response to vulnerabilities using CVE

- **Two types of vulnerability**
  - Zero-day (0-day)
    - A vulnerability has been discovered, but a patch for it has not been released yet
    - I.e., Unknown vulnerability
  - One-day (1-day)
    - A vulnerability has been discovered and a patch for it has been announced, but it has not yet been applied
    - It takes time to apply a security patch right away, so attackers target this period

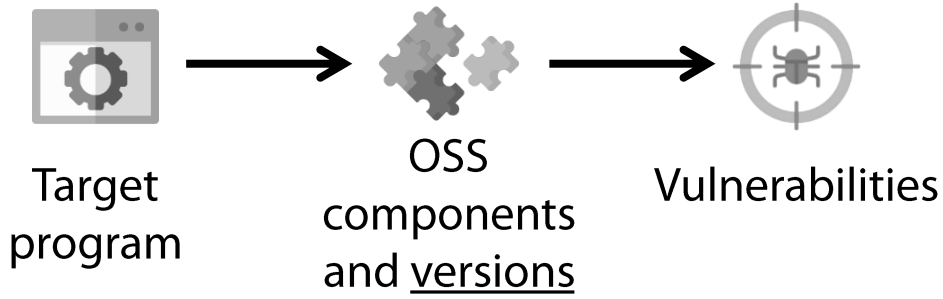
Response to vulnerabilities using CVE

- **Two main approaches to respond to 1-day vulnerabilities**
  - Version-based approach
  - Code-based approach

Response to vulnerabilities using CVE

(1) Version-based vulnerability detection approach

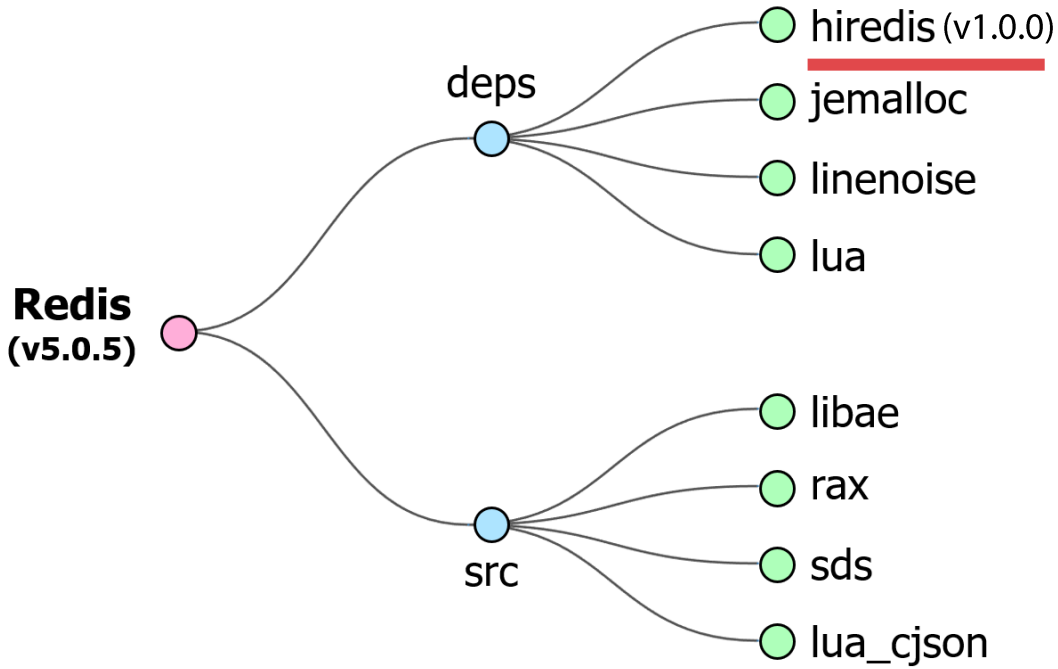
- Detect vulnerabilities based on the name and version of reused OSS



Response to vulnerabilities using CVE

(1) Version-based vulnerability detection approach

- Detect vulnerabilities based on the name and version of reused OSS



Response to vulnerabilities using CVE

(1) Version-based vulnerability detection approach

- Detect vulnerabilities based on the name and version of reused OSS

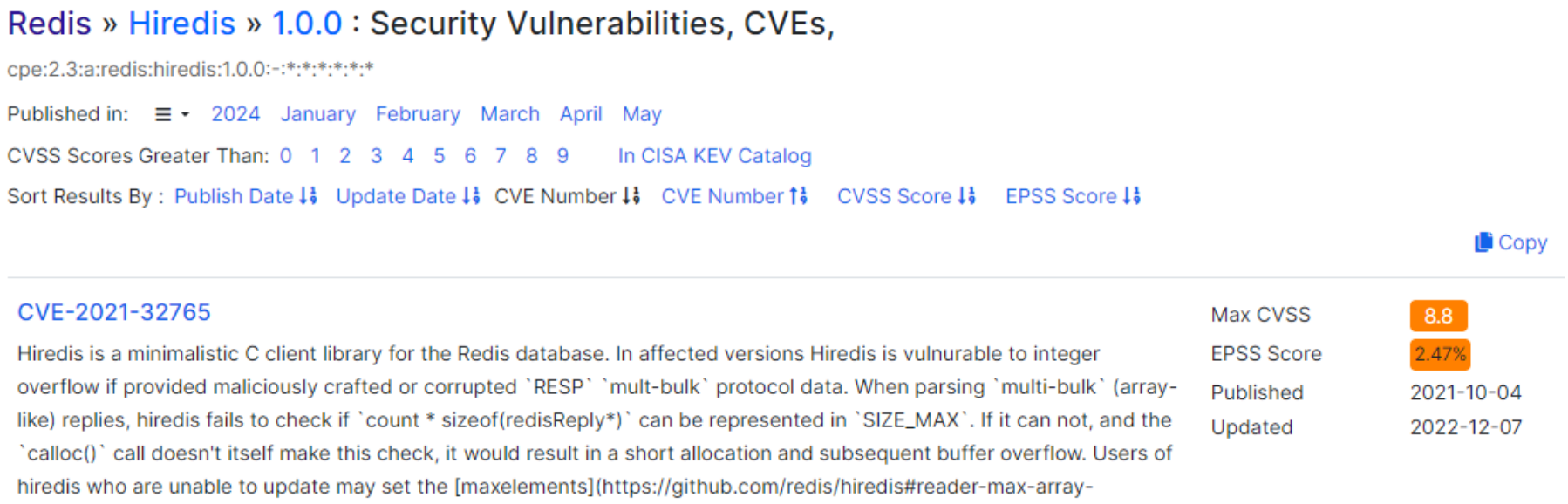
Vendor, Product, Version Search

#	Vendor	Product	Version	Language	Target Platform	Number of Vulnerabilities	
1	Redis	Hiredis	1.0.0 N/A			1	Version Details
2	Redis	Hiredis	1.0.0 rc1			1	Version Details

Response to vulnerabilities using CVE

(1) Version-based vulnerability detection approach

- Detect vulnerabilities based on the name and version of reused OSS



Response to vulnerabilities using CVE

(1) Version-based vulnerability detection approach

- Detect vulnerabilities based on the name and version of reused OSS

CVE-2021-32765 Detail

Description

Hiredis is a minimalistic C client library for the Redis database. In affected versions Hiredis is vulnurable to integer overflow if provided maliciously crafted or corrupted `RESP` `mult-bulk` protocol data. When parsing `multi-bulk` (array-like) replies, hiredis fails to check if `count * sizeof(redisReply*)` can be represented in `SIZE_MAX`. If it can not, and the `calloc()` call doesn't itself make this check, it would result in a short allocation and subsequent buffer overflow. Users of hiredis who are unable to update may set the [maxelements] (https://github.com/redis/hiredis#reader-max-array-elements) context option to a value small enough that no overflow is possible.

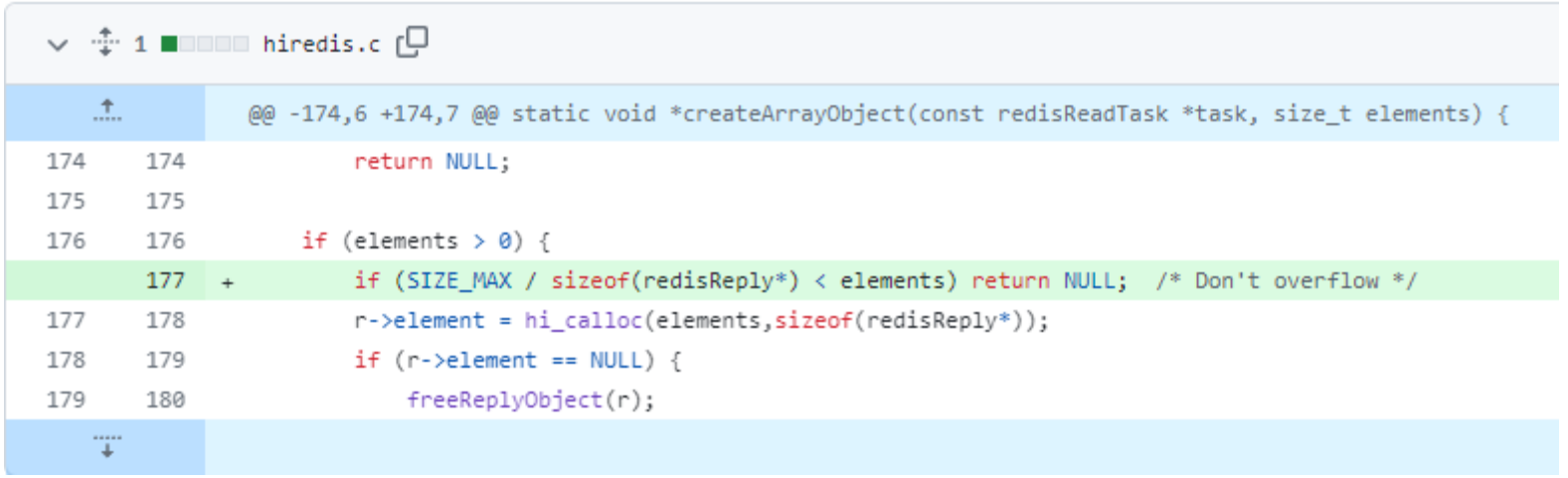
References to Advisories, Solutions, and Tools

Hyperlink	Resource
https://github.com/redis/hiredis/commit/76a7b10005c70babee357a7d0f2becf28ec7ed1e	Patch Third Party Advisory

Response to vulnerabilities using CVE

(1) Version-based vulnerability detection approach

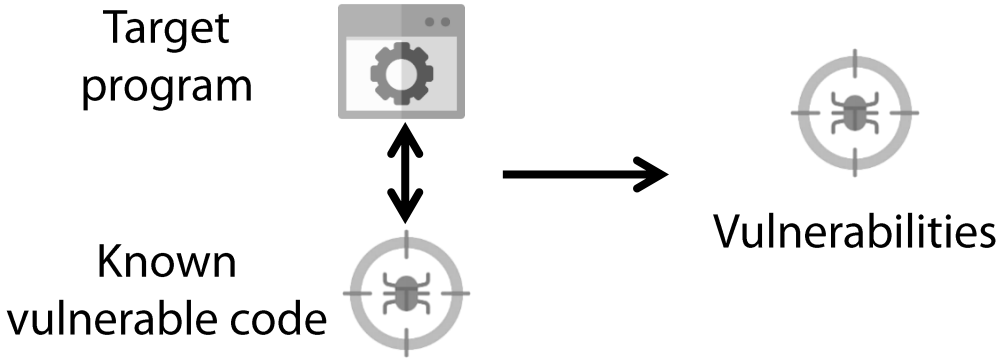
- Detect vulnerabilities based on the name and version of reused OSS



Response to vulnerabilities using CVE

(2) Code-based vulnerability detection approach

- Detect vulnerabilities based on the known vulnerable code
  - Identifying codes syntactically or semantically similar to vulnerable code



Response to vulnerabilities using CVE

(2) Code-based vulnerability detection approach

- Detect vulnerabilities based on the known vulnerable code

```
 1   ...}
 2   if ((flags & FOLL_NUMA) && pte_protnone(pte))
 3     goto no_page;
 4   if ((flags & FOLL_WRITE) && !pte_write(pte)) {
 5     pte_unmap_unlock(ptep, ptl);
 6     return NULL;
 7   }...
 8   if ((ret & VM_FAULT_WRITE) && !(vma->vm_flags & VM_WRITE))
 9     *flags &= ~FOLL_WRITE;
10     return 0;
11   }...
```

A code snippet found in the Android firmware (in 2017)

Response to vulnerabilities using CVE

(2) Code-based vulnerability detection approach

- Detect vulnerabilities based on the known vulnerable code

```
1   ...}
2   if ((flags & FOLL_NUMA) && pte_protnone(pte))
3     goto no_page;
4   if ((flags & FOLL_WRITE) && !pte_write(pte)) {
5     pte_unmap_unlock(ptep, ptl);
6     return NULL;
7   }...
8   if ((ret & VM_FAULT_WRITE) && !(vma->vm_flags & VM_WRITE))
9     *flags &= ~FOLL_WRITE;
10    return 0;
11  }...
```

A code snippet found in the Android firmware (in 2017)

- A part of security patch
  - \- Removed flag to check write permission
  - \+ Add a flag indicating that the request is a COW area

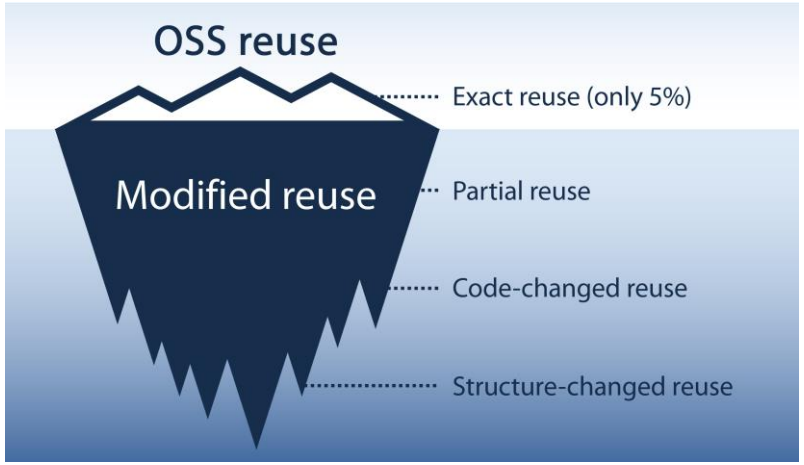

```
@@ -412,7 +422,7 @@ static int faultin_page(struct task_struct *
      * reCOWed by userspace write).
      */
     if ((ret & VM_FAULT_WRITE) && !(vma->vm_flags & VM_WRITE))
-         *flags &= ~FOLL_WRITE;
+         *flags |= FOLL_COW;
     return 0;
 }
```

Response to vulnerabilities using CVE

- **Challenge: Modified OSS reuse**
  - Languages that mainly use package managers (e.g., Java, JavaScript, Python)
    - Modified reuse does not occur often

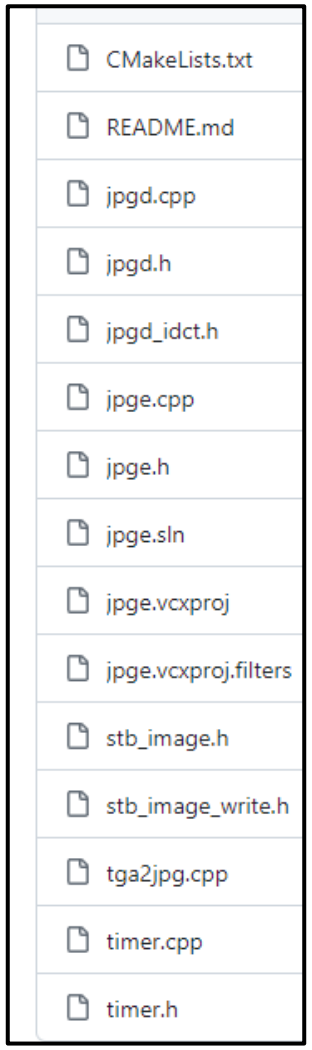
Response to vulnerabilities using CVE

- **Challenge: Modified OSS reuse**
  - Languages that mainly use package managers (e.g., Java, JavaScript, Python)
    - Modified reuse does not occur often
  - In the case of languages that mainly use code level OSS reuse (e.g., C/C++)
    - Modified OSS reuse is prevalent!
    - Reuse only the necessary code parts (partial reuse) with code/structural modifications



Response to vulnerabilities using CVE

- **Example**

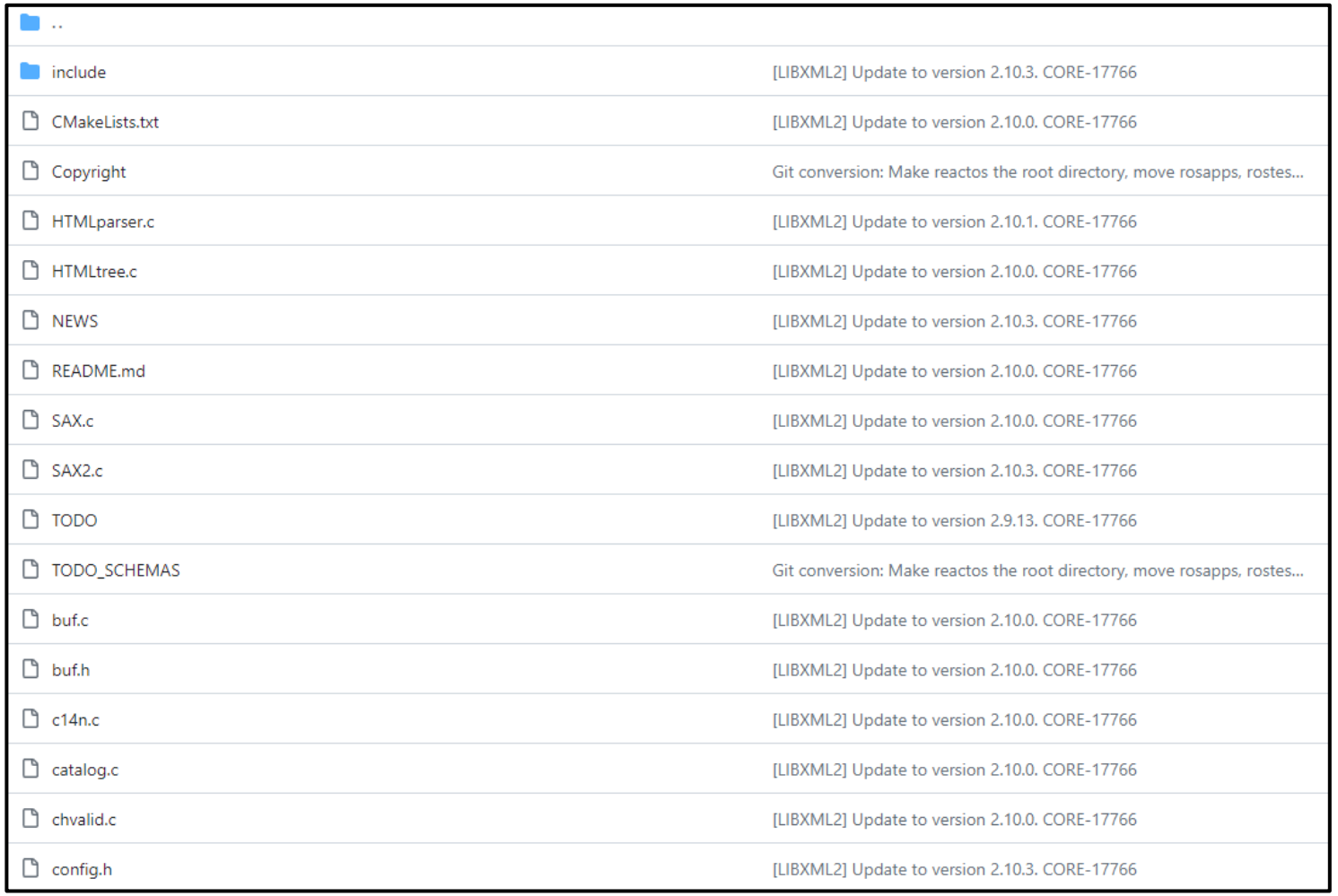


JPEG-compressor

LibGDX/JPEG-compressor

Response to vulnerabilities using CVE

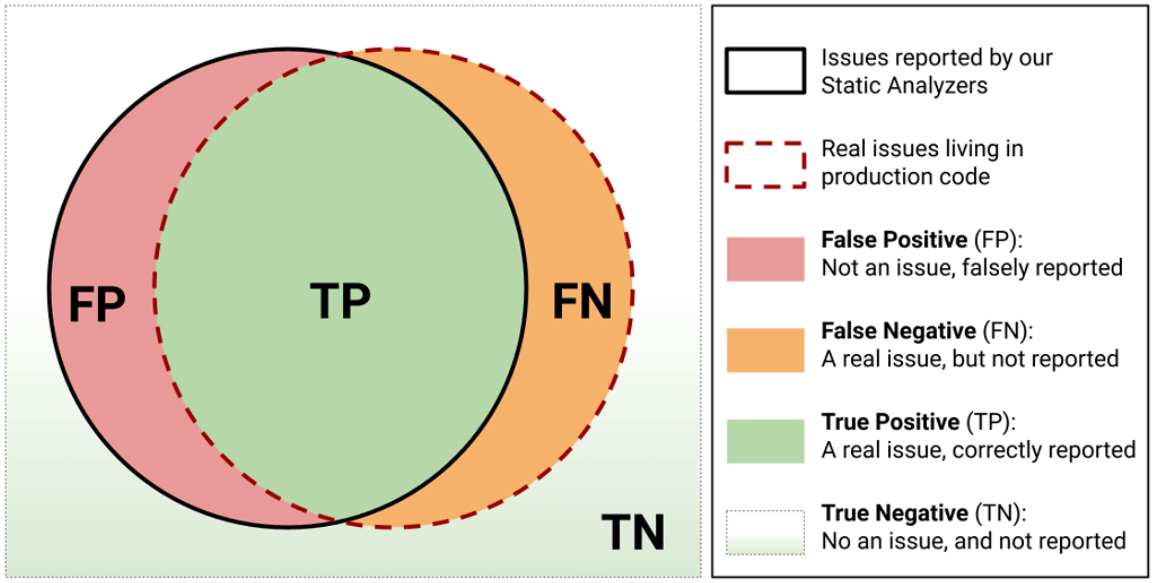
- **Example**
  - ReactOS/LibXML2



..	
include	[LIBXML2] Update to version 2.10.3. CORE-17766
CMakeLists.txt	[LIBXML2] Update to version 2.10.0. CORE-17766
Copyright	Git conversion: Make reactos the root directory, move rosapps, rostes...
HTMLparser.c	[LIBXML2] Update to version 2.10.1. CORE-17766
HTMLtree.c	[LIBXML2] Update to version 2.10.0. CORE-17766
NEWS	[LIBXML2] Update to version 2.10.3. CORE-17766
README.md	[LIBXML2] Update to version 2.10.0. CORE-17766
SAX.c	[LIBXML2] Update to version 2.10.0. CORE-17766
SAX2.c	[LIBXML2] Update to version 2.10.3. CORE-17766
TODO	[LIBXML2] Update to version 2.9.13. CORE-17766
TODO_SCHEMAS	Git conversion: Make reactos the root directory, move rosapps, rostes...
buf.c	[LIBXML2] Update to version 2.10.0. CORE-17766
buf.h	[LIBXML2] Update to version 2.10.0. CORE-17766
c14n.c	[LIBXML2] Update to version 2.10.0. CORE-17766
catalog.c	[LIBXML2] Update to version 2.10.0. CORE-17766
chvalid.c	[LIBXML2] Update to version 2.10.0. CORE-17766
config.h	[LIBXML2] Update to version 2.10.3. CORE-17766

Response to vulnerabilities using CVE

- **Why does modified reuse make vulnerability detection difficult?**
  - Version-based detection techniques produce many false positives
  - Code-based detection techniques yield many false negatives



Response to vulnerabilities using CVE

- **Why does modified reuse make vulnerability detection difficult?**
  - Version-based detection techniques produce many false positives
  - Code-based detection techniques yield many false negatives

Next Lecture

- **Supply chain security**