vul_files_50 Scan Report

Project Name	vul_files_50
Scan Start	Wednesday, January 8, 2025 11:24:42 AM
Preset	Checkmarx Default
Scan Time	03h:08m:26s
Lines Of Code Scanned	288969
Files Scanned	84
Report Creation Time	Wednesday, January 8, 2025 2:58:13 PM
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052
Team	CxServer
Checkmarx Version	8.7.0
Scan Type	Full
Source Origin	LocalPath
Density	3/1000 (Vulnerabilities/LOC)
Visibility	Public

Filter Settings

Severity

Included: High, Medium, Low, Information

Excluded: None

Result State

Included: Confirmed, Not Exploitable, To Verify, Urgent, Proposed Not Exploitable

Excluded: None

Assigned to

Included: All

Categories

Included:

Uncategorized	All
Custom	All
PCI DSS v3.2	All
OWASP Top 10 2013	All
FISMA 2014	All
NIST SP 800-53	All
OWASP Top 10 2017	All
OWASP Mobile Top 10 2016	All

Excluded:

Uncategorized	None
Custom	None
PCI DSS v3.2	None
OWASP Top 10 2013	None
FISMA 2014	None

NIST SP 800-53	None
OWASP Top 10 2017	None
OWASP Mobile Top 10 2016	None

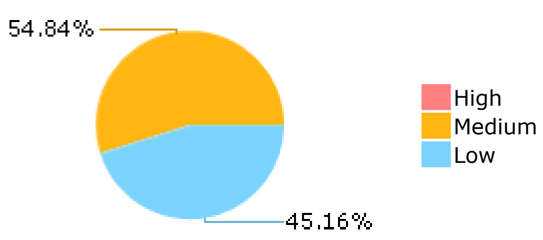
Results Limit

Results limit per query was set to 50

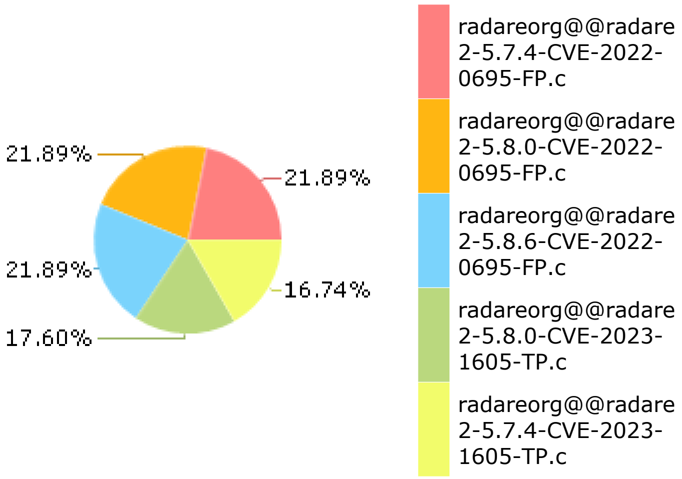
Selected Queries

Selected queries are listed in Result Summary

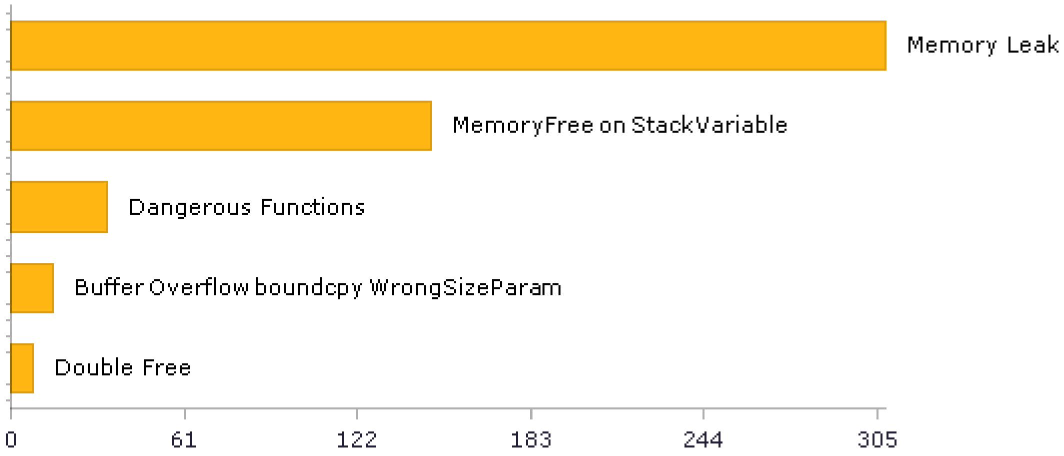
Result Summary



Most Vulnerable Files



Top 5 Vulnerabilities



Scan Summary - OWASP Top 10 2017

Further details and elaboration about vulnerabilities and risks can be found at: OWASP Top 10 2017

Category	Threat Agent	Exploitability	Weakness Prevalence	Weakness Detectability	Technical Impact	Business Impact	Issues Found	Best Fix Locations
A1-Injection	App. Specific	EASY	COMMON	EASY	SEVERE	App. Specific	17	17
A2-Broken Authentication	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	4	4
A3-Sensitive Data Exposure	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	App. Specific	0	0
A4-XML External Entities (XXE)	App. Specific	AVERAGE	COMMON	EASY	SEVERE	App. Specific	0	0
A5-Broken Access Control*	App. Specific	AVERAGE	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A6-Security Misconfiguration	App. Specific	EASY	WIDESPREAD	EASY	MODERATE	App. Specific	0	0
A7-Cross-Site Scripting (XSS)	App. Specific	EASY	WIDESPREAD	EASY	MODERATE	App. Specific	0	0
A8-Insecure Deserialization	App. Specific	DIFFICULT	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A9-Using Components with Known Vulnerabilities*	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	MODERATE	App. Specific	34	34
A10-Insufficient Logging & Monitoring	App. Specific	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	App. Specific	0	0

***** Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - OWASP Top 10 2013

Further details and elaboration about vulnerabilities and risks can be found at: OWASP Top 10 2013

Category	Threat Agent	Attack Vectors	Weakness Prevalence	Weakness Detectability	Technical Impact	Business Impact	Issues Found	Best Fix Locations
A1-Injection	EXTERNAL, INTERNAL, ADMIN USERS	EASY	COMMON	AVERAGE	SEVERE	ALL DATA	0	0
A2-Broken Authentication and Session Management	EXTERNAL, INTERNAL USERS	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	AFFECTED DATA AND FUNCTIONS	0	0
A3-Cross-Site Scripting (XSS)	EXTERNAL, INTERNAL, ADMIN USERS	AVERAGE	VERY WIDESPREAD	EASY	MODERATE	AFFECTED DATA AND SYSTEM	0	0
A4-Insecure Direct Object References	SYSTEM USERS	EASY	COMMON	EASY	MODERATE	EXPOSED DATA	0	0
A5-Security Misconfiguration	EXTERNAL, INTERNAL, ADMIN USERS	EASY	COMMON	EASY	MODERATE	ALL DATA AND SYSTEM	0	0
A6-Sensitive Data Exposure	EXTERNAL, INTERNAL, ADMIN USERS, USERS BROWSERS	DIFFICULT	UNCOMMON	AVERAGE	SEVERE	EXPOSED DATA	0	0
A7-Missing Function Level Access Control*	EXTERNAL, INTERNAL USERS	EASY	COMMON	AVERAGE	MODERATE	EXPOSED DATA AND FUNCTIONS	0	0
A8-Cross-Site Request Forgery (CSRF)	USERS BROWSERS	AVERAGE	COMMON	EASY	MODERATE	AFFECTED DATA AND FUNCTIONS	0	0
A9-Using Components with Known Vulnerabilities*	EXTERNAL USERS, AUTOMATED TOOLS	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	AFFECTED DATA AND FUNCTIONS	34	34
A10-Unvalidated Redirects and Forwards	USERS BROWSERS	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	AFFECTED DATA AND FUNCTIONS	0	0

***** Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - PCI DSS v3.2

Category	Issues Found	Best Fix Locations
PCI DSS (3.2) - 6.5.1 - Injection flaws - particularly SQL injection	0	0
PCI DSS (3.2) - 6.5.2 - Buffer overflows	15	15
PCI DSS (3.2) - 6.5.3 - Insecure cryptographic storage	0	0
PCI DSS (3.2) - 6.5.4 - Insecure communications	0	0
PCI DSS (3.2) - 6.5.5 - Improper error handling*****	0	0
PCI DSS (3.2) - 6.5.7 - Cross-site scripting (XSS)	0	0
PCI DSS (3.2) - 6.5.8 - Improper access control	0	0
PCI DSS (3.2) - 6.5.9 - Cross-site request forgery	0	0
PCI DSS (3.2) - 6.5.10 - Broken authentication and session management	0	0

***** Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - FISMA 2014

Category	Description	Issues Found	Best Fix Locations
Access Control	Organizations must limit information system access to authorized users, processes acting on behalf of authorized users, or devices (including other information systems) and to the types of transactions and functions that authorized users are permitted to exercise.	2	2
Audit And Accountability*	Organizations must: (i) create, protect, and retain information system audit records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful, unauthorized, or inappropriate information system activity; and (ii) ensure that the actions of individual information system users can be uniquely traced to those users so they can be held accountable for their actions.	0	0
Configuration Management	Organizations must: (i) establish and maintain baseline configurations and inventories of organizational information systems (including hardware, software, firmware, and documentation) throughout the respective system development life cycles; and (ii) establish and enforce security configuration settings for information technology products employed in organizational information systems.	0	0
Identification And Authentication*	Organizations must identify information system users, processes acting on behalf of users, or devices and authenticate (or verify) the identities of those users, processes, or devices, as a prerequisite to allowing access to organizational information systems.	2	2
Media Protection	Organizations must: (i) protect information system media, both paper and digital; (ii) limit access to information on information system media to authorized users; and (iii) sanitize or destroy information system media before disposal or release for reuse.	0	0
System And Communications Protection	Organizations must: (i) monitor, control, and protect organizational communications (i.e., information transmitted or received by organizational information systems) at the external boundaries and key internal boundaries of the information systems; and (ii) employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational information systems.	0	0
System And Information Integrity	Organizations must: (i) identify, report, and correct information and information system flaws in a timely manner; (ii) provide protection from malicious code at appropriate locations within organizational information systems; and (iii) monitor information system security alerts and advisories and take appropriate actions in response.	0	0

***** Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - NIST SP 800-53

Category	Issues Found	Best Fix Locations
AC-12 Session Termination (P2)	0	0
AC-3 Access Enforcement (P1)	4	4
AC-4 Information Flow Enforcement (P1)	0	0
AC-6 Least Privilege (P1)	0	0
AU-9 Protection of Audit Information (P1)	0	0
CM-6 Configuration Settings (P2)	0	0
IA-5 Authenticator Management (P1)	0	0
IA-6 Authenticator Feedback (P2)	0	0
IA-8 Identification and Authentication (Non-Organizational Users) (P1)	0	0
SC-12 Cryptographic Key Establishment and Management (P1)	0	0
SC-13 Cryptographic Protection (P1)	0	0
SC-17 Public Key Infrastructure Certificates (P1)	0	0
SC-18 Mobile Code (P2)	0	0
SC-23 Session Authenticity (P1)*****	0	0
SC-28 Protection of Information at Rest (P1)	0	0
SC-4 Information in Shared Resources (P1)	0	0
SC-5 Denial of Service Protection (P1)*****	318	318
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*****	3	3
SI-11 Error Handling (P2)*****	422	422
SI-15 Information Output Filtering (P0)	0	0
SI-16 Memory Protection (P1)	8	8

***** Project scan results do not include all relevant queries. Presets and\or Filters should be changed to include all relevant standard queries.

Scan Summary - OWASP Mobile Top 10 2016

Category	Description	Issues Found	Best Fix Locations
M1-Improper Platform Usage	This category covers misuse of a platform feature or failure to use platform security controls. It might include Android intents, platform permissions, misuse of TouchID, the Keychain, or some other security control that is part of the mobile operating system. There are several ways that mobile apps can experience this risk.	0	0
M2-Insecure Data Storage	This category covers insecure data storage and unintended data leakage.	0	0
M3-Insecure Communication	This category covers poor handshaking, incorrect SSL versions, weak negotiation, cleartext communication of sensitive assets, etc.	0	0
M4-Insecure Authentication	This category captures notions of authenticating the end user or bad session management. This can include: -Failing to identify the user at all when that should be required -Failure to maintain the user's identity when it is required -Weaknesses in session management	0	0
M5-Insufficient Cryptography	The code applies cryptography to a sensitive information asset. However, the cryptography is insufficient in some way. Note that anything and everything related to TLS or SSL goes in M3. Also, if the app fails to use cryptography at all when it should, that probably belongs in M2. This category is for issues where cryptography was attempted, but it wasnt done correctly.	0	0
M6-Insecure Authorization	This is a category to capture any failures in authorization (e.g., authorization decisions in the client side, forced browsing, etc.). It is distinct from authentication issues (e.g., device enrolment, user identification, etc.). If the app does not authenticate users at all in a situation where it should (e.g., granting anonymous access to some resource or service when authenticated and authorized access is required), then that is an authentication failure not an authorization failure.	0	0
M7-Client Code Quality	This category is the catch-all for code-level implementation problems in the mobile client. That's distinct from server-side coding mistakes. This would capture things like buffer overflows, format string vulnerabilities, and various other code-level mistakes where the solution is to rewrite some code that's running on the mobile device.	0	0
M8-Code Tampering	This category covers binary patching, local resource modification, method hooking, method swizzling, and dynamic memory modification. Once the application is delivered to the mobile device, the code and data resources are resident there. An attacker can either directly modify the code, change the contents of memory dynamically, change or replace the system APIs that the application uses, or	0	0

	modify the application's data and resources. This can provide the attacker a direct method of subverting the intended use of the software for personal or monetary gain.		
M9-Reverse Engineering	This category includes analysis of the final core binary to determine its source code, libraries, algorithms, and other assets. Software such as IDA Pro, Hopper, otool, and other binary inspection tools give the attacker insight into the inner workings of the application. This may be used to exploit other nascent vulnerabilities in the application, as well as revealing information about back end servers, cryptographic constants and ciphers, and intellectual property.	0	0
M10-Extraneous Functionality	Often, developers include hidden backdoor functionality or other internal development security controls that are not intended to be released into a production environment. For example, a developer may accidentally include a password as a comment in a hybrid app. Another example includes disabling of 2-factor authentication during testing.	0	0

Scan Summary - Custom

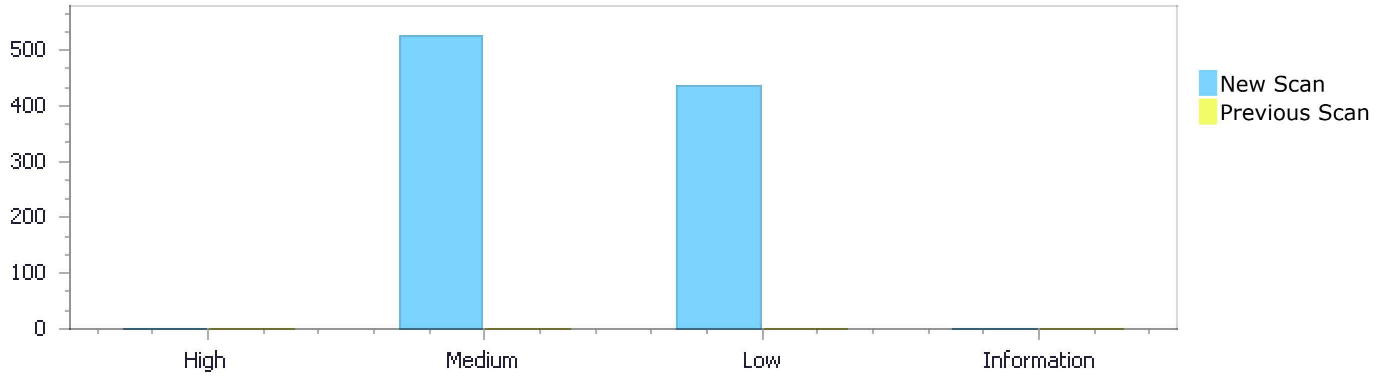
Category	Issues Found	Best Fix Locations
Must audit	0	0
Check	0	0
Optional	0	0

Results Distribution By Status

First scan of the project

	High	Medium	Low	Information	Total
New Issues	0	527	434	0	961
Recurrent Issues	0	0	0	0	0
Total	0	527	434	0	961

Fixed Issues	0	0	0	0	0



Results Distribution By State

	High	Medium	Low	Information	Total
Confirmed	0	0	0	0	0
Not Exploitable	0	0	0	0	0
To Verify	0	527	434	0	961
Urgent	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0
Total	0	527	434	0	961

Result Summary

Vulnerability Type	Occurrences	Severity
Memory Leak	308	Medium
MemoryFree on StackVariable	148	Medium
Dangerous Functions	34	Medium
Buffer Overflow boundcpy WrongSizeParam	15	Medium
Double Free	8	Medium

Use of Zero Initialized Pointer	8	Medium
Wrong Memory Allocation	3	Medium
Wrong Size t Allocation	3	Medium
Unchecked Return Value	422	Low
Sizeof Pointer Argument	4	Low
Improper Resource Access Authorization	2	Low
Incorrect Permission Assignment For Critical Resources	2	Low
NULL Pointer Dereference	2	Low
TOCTOU	2	Low

10 Most Vulnerable Files

High and Medium Vulnerabilities

File Name	Issues Found
radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c	56
radareorg@@radare2-5.8.0-CVE-2022-0695-FP.c	56
radareorg@@radare2-5.8.6-CVE-2022-0695-FP.c	56
radareorg@@radare2-5.8.0-CVE-2023-1605-TP.c	44
radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c	42
radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	24
radareorg@@radare2-5.8.0-CVE-2022-0523-FP.c	24
radareorg@@radare2-5.8.6-CVE-2022-0523-FP.c	24
radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	22
radareorg@@radare2-5.8.0-CVE-2022-0520-FP.c	22

Scan Results Details

Memory Leak

Query Path:
CPP\Cx\CPP Medium Threat\Memory Leak Version:1

Categories

NIST SP 800-53: SC-5 Denial of Service Protection (P1)

Description

Memory Leak\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=637
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Line	290	290
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Method static bool __ne_get_resources(r_bin_ne_obj_t *bin) {

```
....
290.                    res->name = __resource_type_str (ti.rtTypeID &
~0x8000);
```

Memory Leak\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=638
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Line	290	290
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c

Method static bool __ne_get_resources(r_bin_ne_obj_t *bin) {

```
....
290.                    res->name = __resource_type_str (ti.rtTypeID &
~0x8000);
```

Memory Leak\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=639
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1237-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1237-FP.c
Line	290	290
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-1237-FP.c
Method static bool __ne_get_resources(r_bin_ne_obj_t *bin) {

```
....
290.                    res->name = __resource_type_str (ti.rtTypeID &
~0x8000);
```

Memory Leak\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=640
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1238-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1238-FP.c
Line	290	290
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-1238-FP.c
Method static bool __ne_get_resources(r_bin_ne_obj_t *bin) {

```
....
290.                        res->name = __resource_type_str (ti.rtTypeID &
~0x8000);
```

Memory Leak\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=641
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-1237-FP.c	radareorg@@radare2-5.8.6-CVE-2022-1237-FP.c
Line	290	290
Object	name	name

Code Snippet

File Name	radareorg@@radare2-5.8.6-CVE-2022-1237-FP.c
Method	static bool __ne_get_resources(r_bin_ne_obj_t *bin) {

```
....
290.                        res->name = __resource_type_str (ti.rtTypeID &
~0x8000);
```

Memory Leak\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=642
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-1238-FP.c	radareorg@@radare2-5.8.6-CVE-2022-1238-FP.c
Line	290	290
Object	name	name

Code Snippet

File Name	radareorg@@radare2-5.8.6-CVE-2022-1238-FP.c
Method	static bool __ne_get_resources(r_bin_ne_obj_t *bin) {

```
....
290.                        res->name = __resource_type_str (ti.rtTypeID &
~0x8000);
```

Memory Leak\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=643
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	283	283
Object	s	s

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method static pyc_object *get_float_object(RBuffer *buffer) {

```
....
283.         ut8 *s = malloc (n + 1);
```

Memory Leak\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=644
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	343	343
Object	s1	s1

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method static pyc_object *get_complex_object(RBuffer *buffer) {

```
....
343.         ut8 *s1 = malloc (n1 + 1);
```

Memory Leak\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=645

Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	364	364
Object	s2	s2

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method static pyc_object *get_complex_object(RBuffer *buffer) {

```
....
364.        ut8 *s2 = malloc (n2 + 1);
```

Memory Leak\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=646
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Line	42	42
Object	str	str

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Method static char *__read_nonnull_str_at(RBuffer *buf, ut64 offset) {

```
....
42.   char *str = malloc ((ut64)sz + 1);
```

Memory Leak\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=647
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c

Line	132	132
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Method RList *r_bin_ne_get_symbols(r_bin_ne_obj_t *bin) {

```
....
132.            char *name = malloc ((ut64)sz + 1);
```

Memory Leak\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=648
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Line	338	338
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Method RList *r_bin_ne_get_imports(r_bin_ne_obj_t *bin) {

```
....
338.            char *name = malloc ((ut64)sz + 1);
```

Memory Leak\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=649
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Line	42	42
Object	str	str

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c

Method	static char *__read_nonnull_str_at(RBuffer *buf, ut64 offset) {

```
....
42.    char *str = malloc ((ut64)sz + 1);
```

Memory Leak\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=650
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Line	132	132
Object	name	name

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Method	RList *r_bin_ne_get_symbols(r_bin_ne_obj_t *bin) {

```
....
132.            char *name = malloc ((ut64)sz + 1);
```

Memory Leak\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=651
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Line	338	338
Object	name	name

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Method	RList *r_bin_ne_get_imports(r_bin_ne_obj_t *bin) {

```
....
338.            char *name = malloc ((ut64)sz + 1);
```

Memory Leak\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=652
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-0523-FP.c	radareorg@@radare2-5.8.0-CVE-2022-0523-FP.c
Line	283	283
Object	s	s

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-0523-FP.c
Method	static pyc_object *get_float_object(RBuffer *buffer) {

```
....
283.         ut8 *s = malloc (n + 1);
```

Memory Leak\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=653
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-0523-FP.c	radareorg@@radare2-5.8.0-CVE-2022-0523-FP.c
Line	343	343
Object	s1	s1

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-0523-FP.c
Method	static pyc_object *get_complex_object(RBuffer *buffer) {

```
....
343.         ut8 *s1 = malloc (n1 + 1);
```

Memory Leak\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=654
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-0523-FP.c	radareorg@@radare2-5.8.0-CVE-2022-0523-FP.c
Line	365	365
Object	s2	s2

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-0523-FP.c
Method	static pyc_object *get_complex_object(RBuffer *buffer) {

```
....
365.          ut8 *s2 = malloc (n2 + 1);
```

Memory Leak\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=655
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1237-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1237-FP.c
Line	42	42
Object	str	str

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-1237-FP.c
Method	static char *__read_nonnull_str_at(RBuffer *buf, ut64 offset) {

```
....
42.    char *str = malloc ((ut64)sz + 1);
```

Memory Leak\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=656
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1237-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1237-FP.c
Line	132	132

Object	name	name

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-1237-FP.c
Method RList *r_bin_ne_get_symbols(r_bin_ne_obj_t *bin) {

```
....
132.                  char *name = malloc ((ut64)sz + 1);
```

Memory Leak\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=657
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1237-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1237-FP.c
Line	338	338
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-1237-FP.c
Method RList *r_bin_ne_get_imports(r_bin_ne_obj_t *bin) {

```
....
338.                  char *name = malloc ((ut64)sz + 1);
```

Memory Leak\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=658
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1238-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1238-FP.c
Line	42	42
Object	str	str

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-1238-FP.c
Method static char *__read_nonnull_str_at(RBuffer *buf, ut64 offset) {

```
....
42.    char *str = malloc ((ut64)sz + 1);
```

Memory Leak\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=659
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1238-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1238-FP.c
Line	132	132
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-1238-FP.c
Method RList *r_bin_ne_get_symbols(r_bin_ne_obj_t *bin) {

```
....
132.            char *name = malloc ((ut64)sz + 1);
```

Memory Leak\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=660
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1238-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1238-FP.c
Line	338	338
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-1238-FP.c
Method RList *r_bin_ne_get_imports(r_bin_ne_obj_t *bin) {

```
....
338.            char *name = malloc ((ut64)sz + 1);
```

Memory Leak\Path 25:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=661
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-0523-FP.c	radareorg@@radare2-5.8.6-CVE-2022-0523-FP.c
Line	283	283
Object	s	s

Code Snippet

File Name	radareorg@@radare2-5.8.6-CVE-2022-0523-FP.c
Method	static pyc_object *get_float_object(RBuffer *buffer) {

```
....
283.        ut8 *s = malloc (n + 1);
```

Memory Leak\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=662
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-0523-FP.c	radareorg@@radare2-5.8.6-CVE-2022-0523-FP.c
Line	343	343
Object	s1	s1

Code Snippet

File Name	radareorg@@radare2-5.8.6-CVE-2022-0523-FP.c
Method	static pyc_object *get_complex_object(RBuffer *buffer) {

```
....
343.        ut8 *s1 = malloc (n1 + 1);
```

Memory Leak\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=663
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-0523-FP.c	radareorg@@radare2-5.8.6-CVE-2022-0523-FP.c
Line	365	365
Object	s2	s2

Code Snippet
File Name radareorg@@radare2-5.8.6-CVE-2022-0523-FP.c
Method static pyc_object *get_complex_object(RBuffer *buffer) {

```
....
365.        ut8 *s2 = malloc (n2 + 1);
```

Memory Leak\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=664
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-1237-FP.c	radareorg@@radare2-5.8.6-CVE-2022-1237-FP.c
Line	42	42
Object	str	str

Code Snippet
File Name radareorg@@radare2-5.8.6-CVE-2022-1237-FP.c
Method static char *__read_nonnull_str_at(RBuffer *buf, ut64 offset) {

```
....
42.    char *str = malloc ((ut64)sz + 1);
```

Memory Leak\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=665
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-1237-FP.c	radareorg@@radare2-5.8.6-CVE-2022-1237-FP.c
Line	132	132

Object	name	name

Code Snippet
File Name radareorg@@radare2-5.8.6-CVE-2022-1237-FP.c
Method RList *r_bin_ne_get_symbols(r_bin_ne_obj_t *bin) {

```
....
132.            char *name = malloc ((ut64)sz + 1);
```

Memory Leak\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=666
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-1237-FP.c	radareorg@@radare2-5.8.6-CVE-2022-1237-FP.c
Line	338	338
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.8.6-CVE-2022-1237-FP.c
Method RList *r_bin_ne_get_imports(r_bin_ne_obj_t *bin) {

```
....
338.            char *name = malloc ((ut64)sz + 1);
```

Memory Leak\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=667
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-1238-FP.c	radareorg@@radare2-5.8.6-CVE-2022-1238-FP.c
Line	42	42
Object	str	str

Code Snippet
File Name radareorg@@radare2-5.8.6-CVE-2022-1238-FP.c
Method static char *__read_nonnull_str_at(RBuffer *buf, ut64 offset) {

```
....
42.    char *str = malloc ((ut64)sz + 1);
```

Memory Leak\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=668
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-1238-FP.c	radareorg@@radare2-5.8.6-CVE-2022-1238-FP.c
Line	132	132
Object	name	name

Code Snippet

File Name	radareorg@@radare2-5.8.6-CVE-2022-1238-FP.c
Method	RList *r_bin_ne_get_symbols(r_bin_ne_obj_t *bin) {

```
....
132.            char *name = malloc ((ut64)sz + 1);
```

Memory Leak\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=669
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-1238-FP.c	radareorg@@radare2-5.8.6-CVE-2022-1238-FP.c
Line	338	338
Object	name	name

Code Snippet

File Name	radareorg@@radare2-5.8.6-CVE-2022-1238-FP.c
Method	RList *r_bin_ne_get_imports(r_bin_ne_obj_t *bin) {

```
....
338.            char *name = malloc ((ut64)sz + 1);
```

Memory Leak\Path 34:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=670
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	141	141
Object	dbg_file	dbg_file

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method	void init_pdb_downloader(SPDBDownloaderOpt *opt, SPDBDownloader *pd) {

```
....
141.        pd->opt->dbg_file = strdup (opt->dbg_file);
```

Memory Leak\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=671
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	142	142
Object	guid	guid

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method	void init_pdb_downloader(SPDBDownloaderOpt *opt, SPDBDownloader *pd) {

```
....
142.        pd->opt->guid = strdup (opt->guid);
```

Memory Leak\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=672
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	143	143
Object	symbol_server	symbol_server

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method	void init_pdb_downloader(SPDBDownloaderOpt *opt, SPDBDownloader *pd) {

```
....
143.         pd->opt->symbol_server = strdup (opt->symbol_server);
```

Memory Leak\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=673
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	144	144
Object	user_agent	user_agent

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method	void init_pdb_downloader(SPDBDownloaderOpt *opt, SPDBDownloader *pd) {

```
....
144.         pd->opt->user_agent = strdup (opt->user_agent);
```

Memory Leak\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=674
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	145	145

Object	symbol_store_path	symbol_store_path

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method	void init_pdb_downloader(SPDBDownloaderOpt *opt, SPDBDownloader *pd) {

```
....
145.         pd->opt->symbol_store_path = strdup (opt->symbol_store_path);
```

Memory Leak\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=675
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	97	97
Object	data	data

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method	static pyc_object *get_none_object(void) {

```
....
97.          ret->data = strdup ("None");
```

Memory Leak\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=676
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	111	111
Object	data	data

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method	static pyc_object *get_false_object(void) {

```
....
111.         ret->data = strdup ("False");
```

Memory Leak\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=677
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	124	124
Object	data	data

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method static pyc_object *get_true_object(void) {

```
....
124.         ret->data = strdup ("True");
```

Memory Leak\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=678
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	193	193
Object	data	data

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method static pyc_object *get_long_object(RBuffer *buffer) {

```
....
193.                 ret->data = strdup ("0x0");
```

Memory Leak\Path 43:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=679
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	210	210
Object	hexstr	hexstr

Code Snippet

File Name radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c

Method static pyc_object *get_long_object(RBuffer *buffer) {

```
....
210.            hexstr = calloc (size, sizeof (char));
```

Memory Leak\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=680
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	777	777
Object	data	data

Code Snippet

File Name radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c

Method static pyc_object *copy_object(pyc_object *object) {

```
....
777.            copy->data = strdup (object->data);
```

Memory Leak\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=681
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	1144	1144
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method static bool extract_sections_symbols(pyc_object *obj, RList *sections, RList *symbols, RList *cobjs, char *prefix) {

```
....
1144.        section->name = strdup (prefix);
```

Memory Leak\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=682
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	1156	1156
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method static bool extract_sections_symbols(pyc_object *obj, RList *sections, RList *symbols, RList *cobjs, char *prefix) {

```
....
1156.        symbol->name = strdup (prefix);
```

Memory Leak\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=683
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c

Line	18	18
Object	header	header

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c
Method static int r_bin_te_init_hdr(struct r_bin_te_obj_t *bin) {

```
....
18.    if (!(bin->header = malloc (sizeof (TE_image_file_header)))) {
```

Memory Leak\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=684
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c
Line	105	105
Object	section_header	section_header

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c
Method static bool r_bin_te_init_sections(struct r_bin_te_obj_t* bin) {

```
....
105.        if (!(bin->section_header = malloc (sections_size))) {
```

Memory Leak\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=685
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c
Line	166	166
Object	entry	entry

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c

Method	RBinAddr* r_bin_te_get_entrypoint(struct r_bin_te_obj_t* bin) {

```
....
166.        if (!(entry = malloc (sizeof (RBinAddr)))) {
```

Memory Leak\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=686
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c
Line	192	192
Object	machine	machine

Code Snippet	
File Name	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c
Method	char* r_bin_te_get_machine(struct r_bin_te_obj_t* bin) {

```
....
192.                machine = strdup ("Alpha");
```

MemoryFree on StackVariable

Query Path:
CPP\Cx\CPP Medium Threat\MemoryFree on StackVariable Version:0

Description

MemoryFree on StackVariable\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=438
Status	New

Calling free() (line 16) on a variable that was not dynamically allocated (line 16) in file radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	22	22
Object	dir	dir

Code Snippet	
File Name	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c

Method	static bool download_and_write(SPDBDownloaderOpt *opt, const char *file) {

```
....
22.         free (dir);
```

MemoryFree on StackVariable\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=439
Status	New

Calling free() (line 16) on a variable that was not dynamically allocated (line 16) in file radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	32	32
Object	dir	dir

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method	static bool download_and_write(SPDBDownloaderOpt *opt, const char *file) {

```
....
32.         free (dir);
```

MemoryFree on StackVariable\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=440
Status	New

Calling free() (line 16) on a variable that was not dynamically allocated (line 16) in file radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	33	33
Object	path	path

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method	static bool download_and_write(SPDBDownloaderOpt *opt, const char *file) {

```
....
33.         free (path);
```

MemoryFree on StackVariable\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=441
Status	New

Calling free() (line 16) on a variable that was not dynamically allocated (line 16) in file radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	41	41
Object	url	url

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method	static bool download_and_write(SPDBDownloaderOpt *opt, const char *file) {

```
....
41.    free (url);
```

MemoryFree on StackVariable\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=442
Status	New

Calling free() (line 16) on a variable that was not dynamically allocated (line 16) in file radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	43	43
Object	dir	dir

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method	static bool download_and_write(SPDBDownloaderOpt *opt, const char *file) {

```
....
43.         free (dir);
```

MemoryFree on StackVariable\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=443
Status	New

Calling free() (line 16) on a variable that was not dynamically allocated (line 16) in file radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	44	44
Object	file_buf	file_buf

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method static bool download_and_write(SPDBDownloaderOpt *opt, const char *file) {

```
....
44.         free (file_buf);
```

MemoryFree on StackVariable\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=444
Status	New

Calling free() (line 16) on a variable that was not dynamically allocated (line 16) in file radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	45	45
Object	path	path

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method static bool download_and_write(SPDBDownloaderOpt *opt, const char *file) {

```
....
45.         free (path);
```

MemoryFree on StackVariable\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=445
Status	New

Calling free() (line 16) on a variable that was not dynamically allocated (line 16) in file radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	53	53
Object	dir	dir

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method	static bool download_and_write(SPDBDownloaderOpt *opt, const char *file) {

```
....
53.    free (dir);
```

MemoryFree on StackVariable\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=446
Status	New

Calling free() (line 16) on a variable that was not dynamically allocated (line 16) in file radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	54	54
Object	path	path

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method	static bool download_and_write(SPDBDownloaderOpt *opt, const char *file) {

```
....
54.    free (path);
```

MemoryFree on StackVariable\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=447
Status	New

Calling free() (line 16) on a variable that was not dynamically allocated (line 16) in file radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	55	55
Object	file_buf	file_buf

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method	static bool download_and_write(SPDBDownloaderOpt *opt, const char *file) {

```
....
55.    free (file_buf);
```

MemoryFree on StackVariable\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=448
Status	New

Calling free() (line 59) on a variable that was not dynamically allocated (line 59) in file radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	77	77
Object	abspath_to_file	abspath_to_file

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method	static int download(struct SPDBDownloader *pd) {

```
....
77.         free (abspath_to_file);
```

MemoryFree on StackVariable\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=449
Status	New

Calling free() (line 59) on a variable that was not dynamically allocated (line 59) in file radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	99	99
Object	abs_file_esc	abs_file_esc

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method static int download(struct SPDBDownloader *pd) {

```
....
99.         free (abs_file_esc);
```

MemoryFree on StackVariable\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=450
Status	New

Calling free() (line 59) on a variable that was not dynamically allocated (line 59) in file radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	110	110
Object	abs_arch_esc	abs_arch_esc

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method static int download(struct SPDBDownloader *pd) {

```
....
110.                free (abs_arch_esc);
```

MemoryFree on StackVariable\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=451
Status	New

Calling free() (line 59) on a variable that was not dynamically allocated (line 59) in file radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	122	122
Object	abspath_to_archive	abspath_to_archive

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method	static int download(struct SPDBDownloader *pd) {

```
....
122.                free (abspath_to_archive);
```

MemoryFree on StackVariable\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=452
Status	New

Calling free() (line 59) on a variable that was not dynamically allocated (line 59) in file radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	123	123
Object	extractor_cmd	extractor_cmd

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method	static int download(struct SPDBDownloader *pd) {

```
....
123.              free (extractor_cmd);
```

MemoryFree on StackVariable\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=453
Status	New

Calling free() (line 59) on a variable that was not dynamically allocated (line 59) in file radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	130	130
Object	abspath_to_file	abspath_to_file

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method	static int download(struct SPDBDownloader *pd) {

```
....
130.        free (abspath_to_file);
```

MemoryFree on StackVariable\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=454
Status	New

Calling free() (line 84) on a variable that was not dynamically allocated (line 84) in file radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	87	87
Object	ret	ret

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method	static ut8 *get_bytes(RBuffer *buffer, ut32 size) {

```
....
87.          free (ret);
```

MemoryFree on StackVariable\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=455
Status	New

Calling free() (line 270) on a variable that was not dynamically allocated (line 270) in file radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	285	285
Object	ret	ret

Code Snippet	
File Name	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method	static pyc_object *get_float_object(RBuffer *buffer) {

```
....
285.              free (ret);
```

MemoryFree on StackVariable\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=456
Status	New

Calling free() (line 323) on a variable that was not dynamically allocated (line 323) in file radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	340	340
Object	ret	ret

Code Snippet	
File Name	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method	static pyc_object *get_complex_object(RBuffer *buffer) {

```
....
340.                free (ret);
```

MemoryFree on StackVariable\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=457
Status	New

Calling free() (line 323) on a variable that was not dynamically allocated (line 323) in file radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	345	345
Object	ret	ret

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method static pyc_object *get_complex_object(RBuffer *buffer) {

```
....
345.                free (ret);
```

MemoryFree on StackVariable\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=458
Status	New

Calling free() (line 487) on a variable that was not dynamically allocated (line 487) in file radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	498	498
Object	ret	ret

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method static pyc_object *get_array_object_generic(RBuffer *buffer, ut32 size) {

```
....
498.                free (ret);
```

MemoryFree on StackVariable\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=459
Status	New

Calling free() (line 487) on a variable that was not dynamically allocated (line 487) in file radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	507	507
Object	ret	ret

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method	static pyc_object *get_array_object_generic(RBuffer *buffer, ut32 size) {

```
....
507.                       free (ret);
```

MemoryFree on StackVariable\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=460
Status	New

Calling free() (line 829) on a variable that was not dynamically allocated (line 829) in file radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	835	835
Object	ret	ret

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method	static pyc_object *get_code_object(RBuffer *buffer) {

```
....
835.                free (ret);
```

MemoryFree on StackVariable\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=461
Status	New

Calling free() (line 829) on a variable that was not dynamically allocated (line 829) in file radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	836	836
Object	cobj	cobj

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method	static pyc_object *get_code_object(RBuffer *buffer) {

```
....
836.                free (cobj);
```

MemoryFree on StackVariable\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=462
Status	New

Calling free() (line 829) on a variable that was not dynamically allocated (line 829) in file radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	852	852
Object	ret	ret

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method	static pyc_object *get_code_object(RBuffer *buffer) {

```
....
852.                free (ret);
```

MemoryFree on StackVariable\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=463
Status	New

Calling free() (line 829) on a variable that was not dynamically allocated (line 829) in file radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	853	853
Object	cobj	cobj

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method	static pyc_object *get_code_object(RBuffer *buffer) {

```
....
853.                free (cobj);
```

MemoryFree on StackVariable\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=464
Status	New

Calling free() (line 829) on a variable that was not dynamically allocated (line 829) in file radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	953	953
Object	cobj	cobj

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method	static pyc_object *get_code_object(RBuffer *buffer) {

```
....
953.                free (cobj);
```

MemoryFree on StackVariable\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=465
Status	New

Calling free() (line 398) on a variable that was not dynamically allocated (line 398) in file radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c
Line	412	412
Object	buf	buf

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c
Method struct r_bin_te_obj_t* r_bin_te_new(const char* file) {

```
....
412.                free (buf);
```

MemoryFree on StackVariable\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=466
Status	New

Calling free() (line 398) on a variable that was not dynamically allocated (line 398) in file radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c
Line	415	415
Object	buf	buf

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c
Method struct r_bin_te_obj_t* r_bin_te_new(const char* file) {

```
....
415.        free (buf);
```

MemoryFree on StackVariable\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=467
Status	New

Calling free() (line 51) on a variable that was not dynamically allocated (line 51) in file radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Line	67	67
Object	ord	ord

Code Snippet	
File Name	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Method	static char *__func_name_from_ord(const char *module, ut16 ordinal) {

```
....
67.             free (ord);
```

MemoryFree on StackVariable\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=468
Status	New

Calling free() (line 256) on a variable that was not dynamically allocated (line 256) in file radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Line	259	259
Object	en	en

Code Snippet	
File Name	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Method	static void __free_resource_entry(void *entry) {

```
....
259.        free (en);
```

MemoryFree on StackVariable\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=469
Status	New

Calling free() (line 262) on a variable that was not dynamically allocated (line 262) in file radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Line	266	266
Object	res	res

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Method	static void __free_resource(void *resource) {

```
....
266.        free (res);
```

MemoryFree on StackVariable\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=470
Status	New

Calling free() (line 353) on a variable that was not dynamically allocated (line 353) in file radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Line	405	405
Object	entry	entry

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Method	RList *r_bin_ne_get_entrypoints(r_bin_ne_obj_t *bin) {

```
....
405.                          free (entry);
```

MemoryFree on StackVariable\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=471
Status	New

Calling free() (line 353) on a variable that was not dynamically allocated (line 353) in file radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Line	412	412
Object	entry	entry

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Method	RList *r_bin_ne_get_entrypoints(r_bin_ne_obj_t *bin) {

```
....
412.                               free (entry);
```

MemoryFree on StackVariable\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=472
Status	New

Calling free() (line 353) on a variable that was not dynamically allocated (line 353) in file radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Line	421	421
Object	entry	entry

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Method	RList *r_bin_ne_get_entrypoints(r_bin_ne_obj_t *bin) {

```
....
421.                                    free (entry);
```

MemoryFree on StackVariable\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=473
Status	New

Calling free() (line 439) on a variable that was not dynamically allocated (line 439) in file radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Line	532	532
Object	func	func

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Method	RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
532.                                    free (func);
```

MemoryFree on StackVariable\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=474
Status	New

Calling free() (line 439) on a variable that was not dynamically allocated (line 439) in file radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Line	534	534
Object	name	name

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Method	RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
534.                              free (name);
```

MemoryFree on StackVariable\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=475
Status	New

Calling free() (line 51) on a variable that was not dynamically allocated (line 51) in file radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Line	67	67
Object	ord	ord

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Method	static char *__func_name_from_ord(const char *module, ut16 ordinal) {

```
....
67.                 free (ord);
```

MemoryFree on StackVariable\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=476
Status	New

Calling free() (line 256) on a variable that was not dynamically allocated (line 256) in file radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Line	259	259
Object	en	en

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Method	static void __free_resource_entry(void *entry) {

```
....
259.        free (en);
```

MemoryFree on StackVariable\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=477
Status	New

Calling free() (line 262) on a variable that was not dynamically allocated (line 262) in file radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Line	266	266
Object	res	res

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Method	static void __free_resource(void *resource) {

```
....
266.        free (res);
```

MemoryFree on StackVariable\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=478
Status	New

Calling free() (line 353) on a variable that was not dynamically allocated (line 353) in file radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Line	405	405
Object	entry	entry

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Method	RList *r_bin_ne_get_entrypoints(r_bin_ne_obj_t *bin) {

```
....
405.                            free (entry);
```

MemoryFree on StackVariable\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=479
Status	New

Calling free() (line 353) on a variable that was not dynamically allocated (line 353) in file radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Line	412	412
Object	entry	entry

Code Snippet

File Name radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c

Method RList *r_bin_ne_get_entrypoints(r_bin_ne_obj_t *bin) {

```
....
412.                                  free (entry);
```

MemoryFree on StackVariable\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=480
Status	New

Calling free() (line 353) on a variable that was not dynamically allocated (line 353) in file radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Line	421	421
Object	entry	entry

Code Snippet

File Name radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c

Method RList *r_bin_ne_get_entrypoints(r_bin_ne_obj_t *bin) {

```
....
421.                                        free (entry);
```

MemoryFree on StackVariable\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=481
Status	New

Calling free() (line 439) on a variable that was not dynamically allocated (line 439) in file radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Line	532	532
Object	func	func

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Method	RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
532.                                        free (func);
```

MemoryFree on StackVariable\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=482
Status	New

Calling free() (line 439) on a variable that was not dynamically allocated (line 439) in file radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Line	534	534
Object	name	name

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Method	RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
534.                    free (name);
```

MemoryFree on StackVariable\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=483
Status	New

Calling free() (line 131) on a variable that was not dynamically allocated (line 131) in file radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c	radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c
Line	134	134
Object	ptr	ptr

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c
Method	static RBinImport *_fill_bin_import(struct r_bin_coff_obj *bin, int idx) {

```
....
134.            free (ptr);
```

MemoryFree on StackVariable\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=484
Status	New

Calling free() (line 131) on a variable that was not dynamically allocated (line 131) in file radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c	radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c
Line	139	139
Object	ptr	ptr

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c
Method	static RBinImport *_fill_bin_import(struct r_bin_coff_obj *bin, int idx) {

```
....
139.                 free (ptr);
```

MemoryFree on StackVariable\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=485
Status	New

Calling free() (line 131) on a variable that was not dynamically allocated (line 131) in file radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c	radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c
Line	144	144
Object	ptr	ptr

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c
Method	static RBinImport *_fill_bin_import(struct r_bin_coff_obj *bin, int idx) {

```
....
144.                 free (ptr);
```

MemoryFree on StackVariable\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=486
Status	New

Calling free() (line 168) on a variable that was not dynamically allocated (line 168) in file radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c	radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c
Line	189	189
Object	tmp	tmp

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c
Method	static RList *sections(RBinFile *bf) {

```
....
189.                              free (tmp);
```

MemoryFree on StackVariable\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=487
Status	New

Calling free() (line 168) on a variable that was not dynamically allocated (line 168) in file radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c	radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c
Line	193	193
Object	tmp	tmp

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c
Method	static RList *sections(RBinFile *bf) {

```
....
193.                        free (tmp);
```

Dangerous Functions

Query Path:

CPP\Cx\CPP Medium Threat\Dangerous Functions Version:1

Categories

OWASP Top 10 2013: A9-Using Components with Known Vulnerabilities
OWASP Top 10 2017: A9-Using Components with Known Vulnerabilities

Description

Dangerous Functions\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=595
Status	New

The dangerous function, memcpy, was found in use at line 750 in radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination

File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	786	786
Object	memcpy	memcpy

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method	static pyc_object *copy_object(pyc_object *object) {

```
....
786.                memcpy (dst, src, sizeof (*dst));
```

Dangerous Functions\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=596
Status	New

The dangerous function, memcpy, was found in use at line 315 in radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c
Line	330	330
Object	memcpy	memcpy

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c
Method	struct r_bin_te_section_t* r_bin_te_get_sections(struct r_bin_te_obj_t* bin) {

```
....
330.                memcpy (sections[i].name, shdr[i].Name,
TE_IMAGE_SIZEOF_NAME);
```

Dangerous Functions\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=597
Status	New

The dangerous function, memcpy, was found in use at line 67 in radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c
Line	103	103
Object	memcpy	memcpy

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c
Method	static int r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
103.        memcpy ((void *)(volatile void *) buf, desc->recv.data,
copy_size);
```

Dangerous Functions\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=598
Status	New

The dangerous function, memcpy, was found in use at line 67 in radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c
Line	105	105
Object	memcpy	memcpy

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c
Method	static int r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
105.        memcpy ((void *)(volatile void *) reg_buf, desc->recv.data,
copy_size);
```

Dangerous Functions\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=599
Status	New

The dangerous function, memcpy, was found in use at line 93 in radareorg@@radare2-5.7.4-CVE-2023-27590-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-27590-TP.c	radareorg@@radare2-5.7.4-CVE-2023-27590-TP.c
Line	112	112
Object	memcpy	memcpy

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2023-27590-TP.c
Method static int __reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
112.            memcpy (buf, bregs, R_MIN (size, sz));
```

Dangerous Functions\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=600
Status	New

The dangerous function, memcpy, was found in use at line 751 in radareorg@@radare2-5.8.0-CVE-2022-0523-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-0523-FP.c	radareorg@@radare2-5.8.0-CVE-2022-0523-FP.c
Line	787	787
Object	memcpy	memcpy

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-0523-FP.c
Method static pyc_object *copy_object(pyc_object *object) {

```
....
787.                memcpy (dst, src, sizeof (*dst));
```

Dangerous Functions\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=601
Status	New

The dangerous function, memcpy, was found in use at line 315 in radareorg@@radare2-5.8.0-CVE-2022-0695-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-0695-FP.c	radareorg@@radare2-5.8.0-CVE-2022-0695-FP.c
Line	330	330
Object	memcpy	memcpy

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-0695-FP.c
Method struct r_bin_te_section_t* r_bin_te_get_sections(struct r_bin_te_obj_t* bin) {

```
....
330.                memcpy (sections[i].name, shdr[i].Name,
TE_IMAGE_SIZEOF_NAME);
```

Dangerous Functions\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=602
Status	New

The dangerous function, memcpy, was found in use at line 66 in radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Line	101	101
Object	memcpy	memcpy

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Method static bool r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
101.          memcpy ((void *)(volatile void *) buf, desc->recv.data,
copy_size);
```

Dangerous Functions\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=603
Status	New

The dangerous function, memcpy, was found in use at line 66 in radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Line	103	103
Object	memcpy	memcpy

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Method static bool r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
103.        memcpy ((void *)(volatile void *) reg_buf, desc->recv.data,
copy_size);
```

Dangerous Functions\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=604
Status	New

The dangerous function, memcpy, was found in use at line 92 in radareorg@@radare2-5.8.0-CVE-2023-27590-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2023-27590-TP.c	radareorg@@radare2-5.8.0-CVE-2023-27590-TP.c
Line	111	111
Object	memcpy	memcpy

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2023-27590-TP.c
Method static bool __reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
111.                memcpy (buf, bregs, R_MIN (size, sz));
```

Dangerous Functions\Path 11:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=605
Status	New

The dangerous function, memcpy, was found in use at line 751 in radareorg@@radare2-5.8.6-CVE-2022-0523-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-0523-FP.c	radareorg@@radare2-5.8.6-CVE-2022-0523-FP.c
Line	787	787
Object	memcpy	memcpy

Code Snippet

File Name	radareorg@@radare2-5.8.6-CVE-2022-0523-FP.c
Method	static pyc_object *copy_object(pyc_object *object) {

```
....
787.                    memcpy (dst, src, sizeof (*dst));
```

Dangerous Functions\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=606
Status	New

The dangerous function, memcpy, was found in use at line 315 in radareorg@@radare2-5.8.6-CVE-2022-0695-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-0695-FP.c	radareorg@@radare2-5.8.6-CVE-2022-0695-FP.c
Line	330	330
Object	memcpy	memcpy

Code Snippet

File Name	radareorg@@radare2-5.8.6-CVE-2022-0695-FP.c
Method	struct r_bin_te_section_t* r_bin_te_get_sections(struct r_bin_te_obj_t* bin) {

```
....
330.               memcpy (sections[i].name, shdr[i].Name,
TE_IMAGE_SIZEOF_NAME);
```

Dangerous Functions\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=607
Status	New

The dangerous function, memcpy, was found in use at line 66 in radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c	radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c
Line	101	101
Object	memcpy	memcpy

Code Snippet
File Name radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c
Method static bool r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
101.        memcpy ((void *)(volatile void *) buf, desc->recv.data,
copy_size);
```

Dangerous Functions\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=608
Status	New

The dangerous function, memcpy, was found in use at line 66 in radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c	radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c
Line	103	103
Object	memcpy	memcpy

Code Snippet
File Name radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c
Method static bool r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
103.        memcpy ((void *)(volatile void *) reg_buf, desc->recv.data,
copy_size);
```

Dangerous Functions\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=609
Status	New

The dangerous function, memcpy, was found in use at line 92 in radareorg@@radare2-5.8.6-CVE-2023-27590-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2023-27590-TP.c	radareorg@@radare2-5.8.6-CVE-2023-27590-TP.c
Line	111	111
Object	memcpy	memcpy

Code Snippet
File Name radareorg@@radare2-5.8.6-CVE-2023-27590-TP.c
Method static bool __reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
111.                memcpy (buf, bregs, R_MIN (size, sz));
```

Dangerous Functions\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=610
Status	New

The dangerous function, strlen, was found in use at line 59 in radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	84	84
Object	strlen	strlen

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method	static int download(struct SPDBDownloader *pd) {

```
....
84.         archive_name[strlen (archive_name) - 1] = '_';
```

Dangerous Functions\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=611
Status	New

The dangerous function, strlen, was found in use at line 17 in radareorg@@radare2-5.7.4-CVE-2023-27590-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-27590-TP.c	radareorg@@radare2-5.7.4-CVE-2023-27590-TP.c
Line	45	45
Object	strlen	strlen

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2023-27590-TP.c
Method	static RList *__io_maps(RDebug *dbg) {

```
....
45.                      memmove (_s_, _s_ + 2, strlen (_s_));
```

Dangerous Functions\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=612
Status	New

The dangerous function, strlen, was found in use at line 17 in radareorg@@radare2-5.7.4-CVE-2023-27590-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-27590-TP.c	radareorg@@radare2-5.7.4-CVE-2023-27590-TP.c
Line	49	49
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2023-27590-TP.c
Method static RList *__io_maps(RDebug *dbg) {

```
....
49.                         memmove (_s_, _s_ + 2, strlen (_s_));
```

Dangerous Functions\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=613
Status	New

The dangerous function, strlen, was found in use at line 93 in radareorg@@radare2-5.7.4-CVE-2023-27590-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-27590-TP.c	radareorg@@radare2-5.7.4-CVE-2023-27590-TP.c
Line	104	104
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2023-27590-TP.c
Method static int __reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
104.          ut8 *bregs = calloc (1, strlen (dr8));
```

Dangerous Functions\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=614
Status	New

The dangerous function, strlen, was found in use at line 59 in radareorg@@radare2-5.8.0-CVE-2022-0520-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-0520-FP.c	radareorg@@radare2-5.8.0-CVE-2022-0520-FP.c
Line	84	84
Object	strlen	strlen

Code Snippet	
File Name	radareorg@@radare2-5.8.0-CVE-2022-0520-FP.c
Method	static int download(struct SPDBDownloader *pd) {

```
....
84.         archive_name[strlen (archive_name) - 1] = '_';
```

Dangerous Functions\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=615
Status	New

The dangerous function, strlen, was found in use at line 16 in radareorg@@radare2-5.8.0-CVE-2023-27590-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2023-27590-TP.c	radareorg@@radare2-5.8.0-CVE-2023-27590-TP.c
Line	44	44
Object	strlen	strlen

Code Snippet	
File Name	radareorg@@radare2-5.8.0-CVE-2023-27590-TP.c
Method	static RList *__io_maps(RDebug *dbg) {

```
....
44.                     memmove (_s_, _s_ + 2, strlen (_s_));
```

Dangerous Functions\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=616
Status	New

The dangerous function, strlen, was found in use at line 16 in radareorg@@radare2-5.8.0-CVE-2023-27590-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2023-27590-TP.c	radareorg@@radare2-5.8.0-CVE-2023-27590-TP.c
Line	48	48

Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2023-27590-TP.c
Method static RList *__io_maps(RDebug *dbg) {

```
....
48.                        memmove (_s_, _s_ + 2, strlen (_s_));
```

Dangerous Functions\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=617
Status	New

The dangerous function, strlen, was found in use at line 92 in radareorg@@radare2-5.8.0-CVE-2023-27590-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2023-27590-TP.c	radareorg@@radare2-5.8.0-CVE-2023-27590-TP.c
Line	103	103
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2023-27590-TP.c
Method static bool __reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
103.        ut8 *bregs = calloc (1, strlen (dr8));
```

Dangerous Functions\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=618
Status	New

The dangerous function, strlen, was found in use at line 16 in radareorg@@radare2-5.8.6-CVE-2023-27590-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2023-27590-TP.c	radareorg@@radare2-5.8.6-CVE-2023-27590-TP.c

Line	44	44
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.8.6-CVE-2023-27590-TP.c
Method static RList *__io_maps(RDebug *dbg) {

```
....
44.                         memmove (_s_, _s_ + 2, strlen (_s_));
```

Dangerous Functions\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=619
Status	New

The dangerous function, strlen, was found in use at line 16 in radareorg@@radare2-5.8.6-CVE-2023-27590-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2023-27590-TP.c	radareorg@@radare2-5.8.6-CVE-2023-27590-TP.c
Line	48	48
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.8.6-CVE-2023-27590-TP.c
Method static RList *__io_maps(RDebug *dbg) {

```
....
48.                         memmove (_s_, _s_ + 2, strlen (_s_));
```

Dangerous Functions\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=620
Status	New

The dangerous function, strlen, was found in use at line 92 in radareorg@@radare2-5.8.6-CVE-2023-27590-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2023-	radareorg@@radare2-5.8.6-CVE-2023-

	27590-TP.c	27590-TP.c
Line	103	103
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.8.6-CVE-2023-27590-TP.c
Method static bool __reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
103.        ut8 *bregs = calloc (1, strlen (dr8));
```

Dangerous Functions\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=621
Status	New

The dangerous function, strncpy, was found in use at line 301 in radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	306	306
Object	strncpy	strncpy

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Method R_API int r_java_assemble(ut64 addr, ut8 *bytes, const char *string) {

```
....
306.        strncpy (name, string, sizeof (name) - 1);
```

Dangerous Functions\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=622
Status	New

The dangerous function, strncpy, was found in use at line 301 in radareorg@@radare2-5.8.0-CVE-2023-5686-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination

File	radareorg@@radare2-5.8.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.8.0-CVE-2023-5686-TP.c
Line	306	306
Object	strncpy	strncpy

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2023-5686-TP.c
Method R_API int r_java_assemble(ut64 addr, ut8 *bytes, const char *string) {

```
....
306.        strncpy (name, string, sizeof (name) - 1);
```

Dangerous Functions\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=623
Status	New

The dangerous function, realloc, was found in use at line 67 in radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c
Line	87	87
Object	realloc	realloc

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c
Method static int r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
87.             ut8 *new_buf = realloc (reg_buf, copy_size);
```

Dangerous Functions\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=624
Status	New

The dangerous function, realloc, was found in use at line 114 in radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c
Line	136	136
Object	realloc	realloc

Code Snippet

File Name radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c

Method static int r_debug_qnx_reg_write(RDebug *dbg, int type, const ut8 *buf, int size) {

```
....
136.                ut8 *new_buf = realloc (reg_buf, buflen * sizeof
(ut8));
```

Dangerous Functions\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=625
Status	New

The dangerous function, realloc, was found in use at line 66 in radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Line	85	85
Object	realloc	realloc

Code Snippet

File Name radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c

Method static bool r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
85.                 ut8 *new_buf = realloc (reg_buf, copy_size);
```

Dangerous Functions\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=626
Status	New

The dangerous function, realloc, was found in use at line 111 in radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Line	133	133
Object	realloc	realloc

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Method static bool r_debug_qnx_reg_write(RDebug *dbg, int type, const ut8 *buf, int size) {

```
....
133.                    ut8 *new_buf = realloc (reg_buf, buflen * sizeof
(ut8));
```

Dangerous Functions\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=627
Status	New

The dangerous function, realloc, was found in use at line 66 in radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c	radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c
Line	85	85
Object	realloc	realloc

Code Snippet
File Name radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c
Method static bool r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
85.                     ut8 *new_buf = realloc (reg_buf, copy_size);
```

Dangerous Functions\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20

	052&pathid=628
Status	New

The dangerous function, realloc, was found in use at line 111 in radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c	radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c
Line	133	133
Object	realloc	realloc

Code Snippet

File Name	radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c
Method	static bool r_debug_qnx_reg_write(RDebug *dbg, int type, const ut8 *buf, int size) {

```
....
133.                ut8 *new_buf = realloc (reg_buf, buflen * sizeof
(ut8));
```

Buffer Overflow boundcpy WrongSizeParam

Query Path:
CPP\Cx\CPP Buffer Overflow\Buffer Overflow boundcpy WrongSizeParam Version:1

Categories

PCI DSS v3.2: PCI DSS (3.2) - 6.5.2 - Buffer overflows
OWASP Top 10 2017: A1-Injection

Description

Buffer Overflow boundcpy WrongSizeParam\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=423
Status	New

The size of the buffer used by r_bin_te_get_sections in TE_IMAGE_SIZEOF_NAME, at line 315 of radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that r_bin_te_get_sections passes to TE_IMAGE_SIZEOF_NAME, at line 315 of radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c, to overwrite the target buffer.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c
Line	330	330
Object	TE_IMAGE_SIZEOF_NAME	TE_IMAGE_SIZEOF_NAME

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c
Method	struct r_bin_te_section_t* r_bin_te_get_sections(struct r_bin_te_obj_t* bin) {

```
....
330.                memcpy (sections[i].name, shdr[i].Name,
TE_IMAGE_SIZEOF_NAME);
```

Buffer Overflow boundcpy WrongSizeParam\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=424
Status	New

The size of the buffer used by r_debug_qnx_reg_read in copy_size, at line 67 of radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that r_debug_qnx_reg_read passes to copy_size, at line 67 of radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c, to overwrite the target buffer.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c
Line	103	103
Object	copy_size	copy_size

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c
Method	static int r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
103.         memcpy ((void *)(volatile void *) buf, desc->recv.data,
copy_size);
```

Buffer Overflow boundcpy WrongSizeParam\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=425
Status	New

The size of the buffer used by r_debug_qnx_reg_read in copy_size, at line 67 of radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that r_debug_qnx_reg_read passes to copy_size, at line 67 of radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c, to overwrite the target buffer.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c
Line	105	105

Object	copy_size	copy_size

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c
Method static int r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
105.        memcpy ((void *)(volatile void *) reg_buf, desc->recv.data,
copy_size);
```

Buffer Overflow boundcpy WrongSizeParam\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=426
Status	New

The size of the buffer used by r_bin_te_get_sections in TE_IMAGE_SIZEOF_NAME, at line 315 of radareorg@@radare2-5.8.0-CVE-2022-0695-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that r_bin_te_get_sections passes to TE_IMAGE_SIZEOF_NAME, at line 315 of radareorg@@radare2-5.8.0-CVE-2022-0695-FP.c, to overwrite the target buffer.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-0695-FP.c	radareorg@@radare2-5.8.0-CVE-2022-0695-FP.c
Line	330	330
Object	TE_IMAGE_SIZEOF_NAME	TE_IMAGE_SIZEOF_NAME

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-0695-FP.c
Method struct r_bin_te_section_t* r_bin_te_get_sections(struct r_bin_te_obj_t* bin) {

```
....
330.            memcpy (sections[i].name, shdr[i].Name,
TE_IMAGE_SIZEOF_NAME);
```

Buffer Overflow boundcpy WrongSizeParam\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=427
Status	New

The size of the buffer used by r_debug_qnx_reg_read in copy_size, at line 66 of radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that r_debug_qnx_reg_read passes to copy_size, at line 66 of radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c, to overwrite the target buffer.

	Source	Destination

File	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Line	101	101
Object	copy_size	copy_size

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Method static bool r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
101.        memcpy ((void *)(volatile void *) buf, desc->recv.data,
copy_size);
```

Buffer Overflow boundcpy WrongSizeParam\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=428
Status	New

The size of the buffer used by r_debug_qnx_reg_read in copy_size, at line 66 of radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that r_debug_qnx_reg_read passes to copy_size, at line 66 of radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c, to overwrite the target buffer.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Line	103	103
Object	copy_size	copy_size

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Method static bool r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
103.        memcpy ((void *)(volatile void *) reg_buf, desc->recv.data,
copy_size);
```

Buffer Overflow boundcpy WrongSizeParam\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=429
Status	New

The size of the buffer used by r_bin_te_get_sections in TE_IMAGE_SIZEOF_NAME, at line 315 of radareorg@@radare2-5.8.6-CVE-2022-0695-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that r_bin_te_get_sections passes to

TE_IMAGE_SIZEOF_NAME, at line 315 of radareorg@@radare2-5.8.6-CVE-2022-0695-FP.c, to overwrite the target buffer.

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-0695-FP.c	radareorg@@radare2-5.8.6-CVE-2022-0695-FP.c
Line	330	330
Object	TE_IMAGE_SIZEOF_NAME	TE_IMAGE_SIZEOF_NAME

Code Snippet
File Name radareorg@@radare2-5.8.6-CVE-2022-0695-FP.c
Method struct r_bin_te_section_t* r_bin_te_get_sections(struct r_bin_te_obj_t* bin) {

```
....
330.                memcpy (sections[i].name, shdr[i].Name, 
TE_IMAGE_SIZEOF_NAME);
```

Buffer Overflow boundcpy WrongSizeParam\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=430
Status	New

The size of the buffer used by r_debug_qnx_reg_read in copy_size, at line 66 of radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that r_debug_qnx_reg_read passes to copy_size, at line 66 of radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c, to overwrite the target buffer.

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c	radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c
Line	101	101
Object	copy_size	copy_size

Code Snippet
File Name radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c
Method static bool r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
101.        memcpy ((void *)(volatile void *) buf, desc->recv.data, 
copy_size);
```

Buffer Overflow boundcpy WrongSizeParam\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=431
Status	New

The size of the buffer used by r_debug_qnx_reg_read in copy_size, at line 66 of radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that r_debug_qnx_reg_read passes to copy_size, at line 66 of radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c, to overwrite the target buffer.

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c	radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c
Line	103	103
Object	copy_size	copy_size

Code Snippet
File Name radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c
Method static bool r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
103.         memcpy ((void *)(volatile void *) reg_buf, desc->recv.data, 
copy_size);
```

Buffer Overflow boundcpy WrongSizeParam\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=432
Status	New

The size of the buffer used by r_debug_qnx_reg_read in size, at line 67 of radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that r_debug_qnx_reg_read passes to size, at line 67 of radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c, to overwrite the target buffer.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c
Line	102	102
Object	size	size

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c
Method static int r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
102.         memset ((void *)(volatile void *) buf, 0, size);
```

Buffer Overflow boundcpy WrongSizeParam\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20

	052&pathid=433
Status	New

The size of the buffer used by r_debug_qnx_reg_read in buflen, at line 67 of radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that r_debug_qnx_reg_read passes to buflen, at line 67 of radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c, to overwrite the target buffer.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c
Line	104	104
Object	buflen	buflen

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c
Method	static int r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
104.        memset ((void *)(volatile void *) reg_buf, 0, buflen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=434
Status	New

The size of the buffer used by r_debug_qnx_reg_read in size, at line 66 of radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that r_debug_qnx_reg_read passes to size, at line 66 of radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c, to overwrite the target buffer.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Line	100	100
Object	size	size

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Method	static bool r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
100.        memset ((void *)(volatile void *) buf, 0, size);
```

Buffer Overflow boundcpy WrongSizeParam\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=435
Status	New

The size of the buffer used by r_debug_qnx_reg_read in buflen, at line 66 of radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that r_debug_qnx_reg_read passes to buflen, at line 66 of radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c, to overwrite the target buffer.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Line	102	102
Object	buflen	buflen

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Method static bool r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
102.         memset ((void *)(volatile void *) reg_buf, 0, buflen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=436
Status	New

The size of the buffer used by r_debug_qnx_reg_read in size, at line 66 of radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that r_debug_qnx_reg_read passes to size, at line 66 of radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c, to overwrite the target buffer.

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c	radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c
Line	100	100
Object	size	size

Code Snippet
File Name radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c
Method static bool r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
100.         memset ((void *)(volatile void *) buf, 0, size);
```

Buffer Overflow boundcpy WrongSizeParam\Path 15:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=437
Status	New

The size of the buffer used by r_debug_qnx_reg_read in buflen, at line 66 of radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that r_debug_qnx_reg_read passes to buflen, at line 66 of radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c, to overwrite the target buffer.

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c	radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c
Line	102	102
Object	buflen	buflen

Code Snippet

File Name	radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c
Method	static bool r_debug_qnx_reg_read(RDebug *dbg, int type, ut8 *buf, int size) {

```
....
102.        memset ((void *)(volatile void *) reg_buf, 0, buflen);
```

Double Free

Query Path:
CPP\Cx\CPP Medium Threat\Double Free Version:1

Categories

NIST SP 800-53: SI-16 Memory Protection (P1)

Description

Double Free\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=629
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Line	513	586
Object	reloc	reloc

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Method	RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
513.                                  free (reloc);
....
586.                            free (reloc);
```

Double Free\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=630
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Line	513	586
Object	reloc	reloc

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Method	RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
513.                                  free (reloc);
....
586.                            free (reloc);
```

Double Free\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=631
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c	radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c
Line	323	454
Object	rel	rel

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c
Method	static RList *_relocs_list(RBin *rbin, struct r_bin_coff_obj *bin, bool patch, ut64 imp_map) {

```
....
323.                        free (rel);
....
454.                  free (rel);
```

Double Free\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=632
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1237-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1237-FP.c
Line	513	586
Object	reloc	reloc

Code Snippet	
File Name	radareorg@@radare2-5.8.0-CVE-2022-1237-FP.c
Method	RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
513.                                    free (reloc);
....
586.                              free (reloc);
```

Double Free\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=633
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1238-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1238-FP.c
Line	513	586
Object	reloc	reloc

Code Snippet	
File Name	radareorg@@radare2-5.8.0-CVE-2022-1238-FP.c
Method	RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
513.                                    free (reloc);
....
586.                              free (reloc);
```

Double Free\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=634
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2023-1605-TP.c	radareorg@@radare2-5.8.0-CVE-2023-1605-TP.c
Line	360	491
Object	rel	rel

Code Snippet	
File Name	radareorg@@radare2-5.8.0-CVE-2023-1605-TP.c
Method	static RList *_relocs_list(RBin *rbin, struct r_bin_coff_obj *bin, bool patch, ut64 imp_map) {

```
....
360.                        free (rel);
....
491.                  free (rel);
```

Double Free\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=635
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-1237-FP.c	radareorg@@radare2-5.8.6-CVE-2022-1237-FP.c
Line	513	586
Object	reloc	reloc

Code Snippet	
File Name	radareorg@@radare2-5.8.6-CVE-2022-1237-FP.c
Method	RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
513.                                        free (reloc);
....
586.                                 free (reloc);
```

Double Free\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=636
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-1238-FP.c	radareorg@@radare2-5.8.6-CVE-2022-1238-FP.c
Line	513	586
Object	reloc	reloc

Code Snippet

File Name	radareorg@@radare2-5.8.6-CVE-2022-1238-FP.c
Method	RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
513.                                        free (reloc);
....
586.                                 free (reloc);
```

Use of Zero Initialized Pointer

Query Path:
CPP\Cx\CPP Medium Threat\Use of Zero Initialized Pointer Version:1

Categories

NIST SP 800-53: SC-5 Denial of Service Protection (P1)

Description

Use of Zero Initialized Pointer\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=945
Status	New

The variable declared in current at radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c in line 114 is not initialized when it is used by current at radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c in line 114.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c

Line	144	146
Object	current	current

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c
Method static int r_debug_qnx_reg_write(RDebug *dbg, int type, const ut8 *buf, int size) {

```
....
144.        RRegItem *current = NULL;
....
146.              current = r_reg_next_diff (dbg->reg, type, reg_buf,
buflen, current, bits);
```

Use of Zero Initialized Pointer\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=946
Status	New

The variable declared in current at radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c in line 111 is not initialized when it is used by current at radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c in line 111.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Line	141	143
Object	current	current

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Method static bool r_debug_qnx_reg_write(RDebug *dbg, int type, const ut8 *buf, int size) {

```
....
141.        RRegItem *current = NULL;
....
143.              current = r_reg_next_diff (dbg->reg, type, reg_buf,
buflen, current, bits);
```

Use of Zero Initialized Pointer\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=947
Status	New

The variable declared in flag_str at radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c in line 269 is not initialized when it is used by flag_str at radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c in line 269.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	274	778
Object	flag_str	flag_str

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
274.        const char *flag_str = NULL;
....
778.        free ((char *)flag_str);
```

Use of Zero Initialized Pointer\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=948
Status	New

The variable declared in current at radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c in line 111 is not initialized when it is used by current at radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c in line 111.

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c	radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c
Line	141	143
Object	current	current

Code Snippet
File Name radareorg@@radare2-5.8.6-CVE-2022-1207-FP.c
Method static bool r_debug_qnx_reg_write(RDebug *dbg, int type, const ut8 *buf, int size) {

```
....
141.        RRegItem *current = NULL;
....
143.              current = r_reg_next_diff (dbg->reg, type, reg_buf,
buflen, current, bits);
```

Use of Zero Initialized Pointer\Path 5:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=949
Status	New

The variable declared in sect at radareorg@@radare2-5.8.6-CVE-2023-0302-FP.c in line 89 is not initialized when it is used by sect at radareorg@@radare2-5.8.6-CVE-2023-0302-FP.c in line 89.

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2023-0302-FP.c	radareorg@@radare2-5.8.6-CVE-2023-0302-FP.c
Line	90	114
Object	sect	sect

Code Snippet

File Name radareorg@@radare2-5.8.6-CVE-2023-0302-FP.c

Method static RList *sections(RBinFile *bf) {

```
....
90.   xbe_section *sect = NULL;
....
114.        sect = calloc (h->sections, sizeof (xbe_section));
```

Use of Zero Initialized Pointer\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=950
Status	New

The variable declared in header at radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c in line 116 is not initialized when it is used by header at radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c in line 99.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c
Line	117	100
Object	header	header

Code Snippet

File Name radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c

Method static int r_bin_te_init(struct r_bin_te_obj_t* bin) {

```
....
117.        bin->header = NULL;
```

File Name radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c

Method	static bool r_bin_te_init_sections(struct r_bin_te_obj_t* bin) {

```
....
100.        int sections_size = sizeof (TE_image_section_header) * bin-
>header->NumberOfSections;
```

Use of Zero Initialized Pointer\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=951
Status	New

The variable declared in header at radareorg@@radare2-5.8.0-CVE-2022-0695-FP.c in line 116 is not initialized when it is used by header at radareorg@@radare2-5.8.0-CVE-2022-0695-FP.c in line 99.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-0695-FP.c	radareorg@@radare2-5.8.0-CVE-2022-0695-FP.c
Line	117	100
Object	header	header

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-0695-FP.c
Method	static int r_bin_te_init(struct r_bin_te_obj_t* bin) {

```
....
117.        bin->header = NULL;
```

▼

File Name	radareorg@@radare2-5.8.0-CVE-2022-0695-FP.c
Method	static bool r_bin_te_init_sections(struct r_bin_te_obj_t* bin) {

```
....
100.        int sections_size = sizeof (TE_image_section_header) * bin-
>header->NumberOfSections;
```

Use of Zero Initialized Pointer\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=952
Status	New

The variable declared in header at radareorg@@radare2-5.8.6-CVE-2022-0695-FP.c in line 116 is not initialized when it is used by header at radareorg@@radare2-5.8.6-CVE-2022-0695-FP.c in line 99.

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-0695-FP.c	radareorg@@radare2-5.8.6-CVE-2022-0695-FP.c
Line	117	100
Object	header	header

Code Snippet
File Name radareorg@@radare2-5.8.6-CVE-2022-0695-FP.c
Method static int r_bin_te_init(struct r_bin_te_obj_t* bin) {

```
....
117.        bin->header = NULL;
```

File Name radareorg@@radare2-5.8.6-CVE-2022-0695-FP.c

Method static bool r_bin_te_init_sections(struct r_bin_te_obj_t* bin) {

```
....
100.        int sections_size = sizeof (TE_image_section_header) * bin-
>header->NumberOfSections;
```

Wrong Size t Allocation

Query Path:
CPP\Cx\CPP Integer Overflow\Wrong Size t Allocation Version:0

Description

Wrong Size t Allocation\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=586
Status	New

The function size in radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c at line 169 assigns an incorrectly calculated size to a buffer, resulting in a mismatch between the value being written and the size of the buffer it is being written into.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Line	210	210
Object	size	size

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0523-FP.c
Method static pyc_object *get_long_object(RBuffer *buffer) {

```
....
210.                hexstr = calloc (size, sizeof (char));
```

Wrong Size t Allocation\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=587
Status	New

The function size in radareorg@@radare2-5.8.0-CVE-2022-0523-FP.c at line 169 assigns an incorrectly calculated size to a buffer, resulting in a mismatch between the value being written and the size of the buffer it is being written into.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-0523-FP.c	radareorg@@radare2-5.8.0-CVE-2022-0523-FP.c
Line	210	210
Object	size	size

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-0523-FP.c
Method	static pyc_object *get_long_object(RBuffer *buffer) {

```
....
210.                hexstr = calloc (size, sizeof (char));
```

Wrong Size t Allocation\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=588
Status	New

The function size in radareorg@@radare2-5.8.6-CVE-2022-0523-FP.c at line 169 assigns an incorrectly calculated size to a buffer, resulting in a mismatch between the value being written and the size of the buffer it is being written into.

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-0523-FP.c	radareorg@@radare2-5.8.6-CVE-2022-0523-FP.c
Line	210	210
Object	size	size

Code Snippet

File Name	radareorg@@radare2-5.8.6-CVE-2022-0523-FP.c

Method static pyc_object *get_long_object(RBuffer *buffer) {

```
....
210.            hexstr = calloc (size, sizeof (char));
```

Wrong Memory Allocation

Query Path:
CPP\Cx\CPP Medium Threat\Wrong Memory Allocation Version:0

Categories

NIST SP 800-53: SI-10 Information Input Validation (P1)

Description

Wrong Memory Allocation\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=953
Status	New

The function malloc in radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c at line 14 assigns an incorrectly calculated size to a buffer, resulting in a mismatch between the value being written and the size of the buffer it is being written into.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c
Line	18	18
Object	sizeof	malloc

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0695-FP.c
Method static int r_bin_te_init_hdr(struct r_bin_te_obj_t *bin) {

```
....
18.    if (!(bin->header = malloc (sizeof (TE_image_file_header)))) {
```

Wrong Memory Allocation\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=954
Status	New

The function malloc in radareorg@@radare2-5.8.0-CVE-2022-0695-FP.c at line 14 assigns an incorrectly calculated size to a buffer, resulting in a mismatch between the value being written and the size of the buffer it is being written into.

	Source	Destination

File	radareorg@@radare2-5.8.0-CVE-2022-0695-FP.c	radareorg@@radare2-5.8.0-CVE-2022-0695-FP.c
Line	18	18
Object	sizeof	malloc

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-0695-FP.c
Method static int r_bin_te_init_hdr(struct r_bin_te_obj_t *bin) {

```
....
18.    if (!(bin->header = malloc (sizeof (TE_image_file_header)))) {
```

Wrong Memory Allocation\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=955
Status	New

The function malloc in radareorg@@radare2-5.8.6-CVE-2022-0695-FP.c at line 14 assigns an incorrectly calculated size to a buffer, resulting in a mismatch between the value being written and the size of the buffer it is being written into.

	Source	Destination
File	radareorg@@radare2-5.8.6-CVE-2022-0695-FP.c	radareorg@@radare2-5.8.6-CVE-2022-0695-FP.c
Line	18	18
Object	sizeof	malloc

Code Snippet
File Name radareorg@@radare2-5.8.6-CVE-2022-0695-FP.c
Method static int r_bin_te_init_hdr(struct r_bin_te_obj_t *bin) {

```
....
18.    if (!(bin->header = malloc (sizeof (TE_image_file_header)))) {
```

Unchecked Return Value

Query Path:
CPP\Cx\CPP Low Visibility\Unchecked Return Value Version:1

Categories

NIST SP 800-53: SI-11 Error Handling (P2)

Description

Unchecked Return Value\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=1
Status	New

The *r_debug_qnx_reg_profile method calls the strdup function, at line 229 of radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c
Line	234	234
Object	strdup	strdup

Code Snippet

File Name radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c

Method static const char *r_debug_qnx_reg_profile(RDebug *dbg) {

```
....
234.                return strdup (
```

Unchecked Return Value\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=2
Status	New

The *r_debug_qnx_reg_profile method calls the strdup function, at line 229 of radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c
Line	263	263
Object	strdup	strdup

Code Snippet

File Name radareorg@@radare2-5.7.4-CVE-2022-1207-FP.c

Method static const char *r_debug_qnx_reg_profile(RDebug *dbg) {

```
....
263.                        return strdup (
```

Unchecked Return Value\Path 3:

Severity	Low
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=3
Status	New

The *__resource_type_str method calls the strdup function, at line 181 of radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Line	253	253
Object	strdup	strdup

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1237-FP.c
Method	static char *__resource_type_str(int type) {

```
....
253.         return strdup (typeName);
```

Unchecked Return Value\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=4
Status	New

The *__resource_type_str method calls the strdup function, at line 181 of radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Line	253	253
Object	strdup	strdup

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2022-1238-FP.c
Method	static char *__resource_type_str(int type) {

```
....
253.         return strdup (typeName);
```

Unchecked Return Value\Path 5:

Severity	Low

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=5
Status	New

The handle_switch_op method calls the snprintf function, at line 96 of radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	101	101
Object	snprintf	snprintf

Code Snippet	
File Name	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Method	static int handle_switch_op(ut64 addr, const ut8 * bytes, char *output, int outlen) {

```
....
101.        snprintf (output, outlen, "case %d: goto 0x%04x", ccase, jmp);
```

Unchecked Return Value\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=6
Status	New

The java_print_opcode method calls the snprintf function, at line 105 of radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	124	124
Object	snprintf	snprintf

Code Snippet	
File Name	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Method	R_API int java_print_opcode(RBinJavaObj *obj, ut64 addr, int idx, const ut8 *bytes, int len, char *output, int outlen) {

```
....
124.                snprintf (output, outlen, "%s %d", JAVA_OPS[idx].name,
(char) bytes[1]);
```

Unchecked Return Value\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=7
Status	New

The java_print_opcode method calls the snprintf function, at line 105 of radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	128	128
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Method R_API int java_print_opcode(RBinJavaObj *obj, ut64 addr, int idx, const ut8 *bytes, int len, char *output, int outlen) {

```
....
128.                snprintf (output, outlen, "%s %d", JAVA_OPS[idx].name,
(int)USHORT (bytes, 1));
```

Unchecked Return Value\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=8
Status	New

The java_print_opcode method calls the snprintf function, at line 105 of radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	142	142
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Method	R_API int java_print_opcode(RBinJavaObj *obj, ut64 addr, int idx, const ut8 *bytes, int len, char *output, int outlen) {

```
....
142.                snprintf (output, outlen, "%s %d", JAVA_OPS[idx].name,
bytes[1]);
```

Unchecked Return Value\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=9
Status	New

The java_print_opcode method calls the snprintf function, at line 105 of radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	149	149
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Method	R_API int java_print_opcode(RBinJavaObj *obj, ut64 addr, int idx, const ut8 *bytes, int len, char *output, int outlen) {

```
....
149.                       snprintf (output, outlen, "%s %s",
JAVA_OPS[idx].name, arg);
```

Unchecked Return Value\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=10
Status	New

The java_print_opcode method calls the snprintf function, at line 105 of radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination

File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	152	152
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Method R_API int java_print_opcode(RBinJavaObj *obj, ut64 addr, int idx, const ut8 *bytes, int len, char *output, int outlen) {

```
....
152.                          snprintf (output, outlen, "%s #%d",
JAVA_OPS[idx].name, USHORT (bytes, 1));
```

Unchecked Return Value\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=11
Status	New

The java_print_opcode method calls the snprintf function, at line 105 of radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	160	160
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Method R_API int java_print_opcode(RBinJavaObj *obj, ut64 addr, int idx, const ut8 *bytes, int len, char *output, int outlen) {

```
....
160.                          snprintf (output, outlen, "%s %s",
JAVA_OPS[idx].name, arg);
```

Unchecked Return Value\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=12
Status	New

The java_print_opcode method calls the snprintf function, at line 105 of radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	163	163
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Method R_API int java_print_opcode(RBinJavaObj *obj, ut64 addr, int idx, const ut8 *bytes, int len, char *output, int outlen) {

```
....
163.                    snprintf (output, outlen, "%s #%d",
JAVA_OPS[idx].name, USHORT (bytes, 1));
```

Unchecked Return Value\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=13
Status	New

The java_print_opcode method calls the snprintf function, at line 105 of radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	170	170
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Method R_API int java_print_opcode(RBinJavaObj *obj, ut64 addr, int idx, const ut8 *bytes, int len, char *output, int outlen) {

```
....
170.              snprintf (output, outlen, "%s %d %d",
JAVA_OPS[idx].name, val_one, val_two);
```

Unchecked Return Value\Path 14:

Severity	Low
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=14
Status	New

The java_print_opcode method calls the snprintf function, at line 105 of radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	208	208
Object	snprintf	snprintf

Code Snippet

File Name radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c

Method R_API int java_print_opcode(RBinJavaObj *obj, ut64 addr, int idx, const ut8 *bytes, int len, char *output, int outlen) {

```
....
208.                         snprintf (output, outlen, "%s %s",
JAVA_OPS[idx].name, arg);
```

Unchecked Return Value\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=15
Status	New

The java_print_opcode method calls the snprintf function, at line 105 of radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	211	211
Object	snprintf	snprintf

Code Snippet

File Name radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c

Method R_API int java_print_opcode(RBinJavaObj *obj, ut64 addr, int idx, const ut8 *bytes, int len, char *output, int outlen) {

```
....
211.                    snprintf (output, outlen, "%s #%d",
JAVA_OPS[idx].name, USHORT (bytes, 1) );
```

Unchecked Return Value\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=16
Status	New

The java_print_opcode method calls the snprintf function, at line 105 of radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	221	221
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Method	R_API int java_print_opcode(RBinJavaObj *obj, ut64 addr, int idx, const ut8 *bytes, int len, char *output, int outlen) {

```
....
221.                    snprintf (output, outlen, "%s %s",
JAVA_OPS[idx].name, arg);
```

Unchecked Return Value\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=17
Status	New

The java_print_opcode method calls the snprintf function, at line 105 of radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	224	224
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Method	R_API int java_print_opcode(RBinJavaObj *obj, ut64 addr, int idx, const ut8 *bytes, int len, char *output, int outlen) {

```
....
224.                        snprintf (output, outlen, "%s #%d",
JAVA_OPS[idx].name, USHORT (bytes, 1) );
```

Unchecked Return Value\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=18
Status	New

The java_print_opcode method calls the snprintf function, at line 105 of radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	234	234
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Method	R_API int java_print_opcode(RBinJavaObj *obj, ut64 addr, int idx, const ut8 *bytes, int len, char *output, int outlen) {

```
....
234.                        snprintf (output, outlen, "%s %s",
JAVA_OPS[idx].name, arg);
```

Unchecked Return Value\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=19
Status	New

The java_print_opcode method calls the snprintf function, at line 105 of radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination

File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	237	237
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Method	R_API int java_print_opcode(RBinJavaObj *obj, ut64 addr, int idx, const ut8 *bytes, int len, char *output, int outlen) {

```
....
237.                    snprintf (output, outlen, "%s #%d",
JAVA_OPS[idx].name, USHORT (bytes, 1) );
```

Unchecked Return Value\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=20
Status	New

The java_print_opcode method calls the snprintf function, at line 105 of radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	245	245
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Method	R_API int java_print_opcode(RBinJavaObj *obj, ut64 addr, int idx, const ut8 *bytes, int len, char *output, int outlen) {

```
....
245.          case 1: snprintf (output, outlen, "%s", JAVA_OPS[idx].name);
```

Unchecked Return Value\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=21
Status	New

The java_print_opcode method calls the snprintf function, at line 105 of radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	247	247
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Method	R_API int java_print_opcode(RBinJavaObj *obj, ut64 addr, int idx, const ut8 *bytes, int len, char *output, int outlen) {

```
....
247.         case 2: snprintf (output, outlen, "%s %d",
JAVA_OPS[idx].name, bytes[1]);
```

Unchecked Return Value\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=22
Status	New

The java_print_opcode method calls the snprintf function, at line 105 of radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	249	249
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Method	R_API int java_print_opcode(RBinJavaObj *obj, ut64 addr, int idx, const ut8 *bytes, int len, char *output, int outlen) {

```
....
249.         case 3: snprintf (output, outlen, "%s 0x%04x 0x%04x",
JAVA_OPS[idx].name, bytes[0], bytes[1]);
```

Unchecked Return Value\Path 23:

Severity	Low
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=23
Status	New

The java_print_opcode method calls the snprintf function, at line 105 of radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	251	251
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Method	R_API int java_print_opcode(RBinJavaObj *obj, ut64 addr, int idx, const ut8 *bytes, int len, char *output, int outlen) {

```
....
251.         case 5: snprintf (output, outlen, "%s %d",
JAVA_OPS[idx].name, bytes[1]);
```

Unchecked Return Value\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=24
Status	New

The *r_debug_qnx_reg_profile method calls the strdup function, at line 226 of radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Line	231	231
Object	strdup	strdup

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Method	static const char *r_debug_qnx_reg_profile(RDebug *dbg) {

```
....
231.              return strdup (
```

Unchecked Return Value\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=25
Status	New

The *r_debug_qnx_reg_profile method calls the strdup function, at line 226 of radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Line	260	260
Object	strdup	strdup

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-1207-FP.c
Method	static const char *r_debug_qnx_reg_profile(RDebug *dbg) {

```
....
260.                    return strdup (
```

Unchecked Return Value\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=26
Status	New

The *__resource_type_str method calls the strdup function, at line 181 of radareorg@@radare2-5.8.0-CVE-2022-1237-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1237-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1237-FP.c
Line	253	253
Object	strdup	strdup

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-1237-FP.c
Method	static char *__resource_type_str(int type) {

```
....
253.        return strdup (typeName);
```

Unchecked Return Value\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=27
Status	New

The *__resource_type_str method calls the strdup function, at line 181 of radareorg@@radare2-5.8.0-CVE-2022-1238-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-1238-FP.c	radareorg@@radare2-5.8.0-CVE-2022-1238-FP.c
Line	253	253
Object	strdup	strdup

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-1238-FP.c
Method static char *__resource_type_str(int type) {

```
....
253.        return strdup (typeName);
```

Unchecked Return Value\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=28
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	339	339
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
339.                    snprintf (str, sizeof (str), " v%i, v%i", vA,
vB);
```

Unchecked Return Value\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=29
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	345	345
Object	snprintf	snprintf

Code Snippet

File Name radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c

Method static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
345.                    snprintf (str, sizeof (str), " v%i, v%i", vA,
vB);
```

Unchecked Return Value\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=30
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	351	351
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method	static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
351.                    snprintf (str, sizeof (str), " v%i, v%i", vA,
vB);
```

Unchecked Return Value\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=31
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	356	356
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method	static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
356.                    snprintf (str, sizeof (str), " v%i", vA);
```

Unchecked Return Value\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=32
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-	radareorg@@radare2-5.8.0-CVE-2022-

	28069-FP.c	28069-FP.c
Line	362	362
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
362.                    snprintf (str, sizeof (str), " v%i, %#x", vA,
vB);
```

Unchecked Return Value\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=33
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	369	369
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
369.                         snprintf (str, sizeof (str), " v%i,
%#04hx", vA, sB);
```

Unchecked Return Value\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=34
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	377	377
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method	static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
377.                          snprintf (str, sizeof (str), " v%i:v%i,
0x%08x", vA, vA + 1, vB);
```

Unchecked Return Value\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=35
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	379	379
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method	static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
379.                          snprintf (str, sizeof (str), " v%i,
0x%08x", vA, vB);
```

Unchecked Return Value\Path 36:

Severity	Low
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=36
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	388	388
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method	static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
388.                    snprintf (str, sizeof (str), " v%i:v%i,
0x%08x", vA, vA + 1, vB);
```

Unchecked Return Value\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=37
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	390	390
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method	static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
390.                     snprintf (str, sizeof (str), " v%i,
0x%08x", vA, vB);
```

Unchecked Return Value\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=38
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	407	407
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method	static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
407.                snprintf (str, sizeof (str), " v%i, v%i, v%i",
vA, vB, vC);
```

Unchecked Return Value\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=39
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	414	414
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method	static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
414.                    snprintf (str, sizeof (str), " v%i, v%i, %#x",
vA, vB, vC);
```

Unchecked Return Value\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=40
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	421	421
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method	static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
421.                    snprintf (str, sizeof (str), " v%i, v%i, %#x",
vA, vB, vC);
```

Unchecked Return Value\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=41
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination

File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	468	468
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
468.                               snprintf (str, sizeof (str), " {v%i}",
buf[4] & 0x0f);
```

Unchecked Return Value\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=42
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	471	471
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
471.                               snprintf (str, sizeof (str), " {v%i,
v%i}", buf[4] & 0x0f, (buf[4] & 0xf0) >> 4);
```

Unchecked Return Value\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=43
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	474	474
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
474.                                    snprintf (str, sizeof (str), " {v%i, v%i,
v%i}", buf[4] & 0x0f, (buf[4] & 0xf0) >> 4, buf[5] & 0x0f);
```

Unchecked Return Value\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=44
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	477	477
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
477.                                    snprintf (str, sizeof (str), " {v%i, v%i,
v%i, v%i}", buf[4] & 0x0f,
```

Unchecked Return Value\Path 45:

Severity	Low
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=45
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	481	481
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method	static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
481.                                snprintf (str, sizeof (str), " {}");
```

Unchecked Return Value\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=46
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	484	484
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method	static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
484.                          snprintf (str, sizeof (str), ", [%04x]", vB);
```

Unchecked Return Value\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=47
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	492	492
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method	static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
492.                        snprintf (str, sizeof (str), " {v%i..v%i},
[%04x]", vC, vC + vA - 1, vB);
```

Unchecked Return Value\Path 48:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=48
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	500	500
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method	static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
500.                          snprintf (str, sizeof (str), " {v%i}",
buf[4] & 0x0f);
```

Unchecked Return Value\Path 49:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=49
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	503	503
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method	static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
503.                          snprintf (str, sizeof (str), " {v%i,
v%i}", buf[4] & 0x0f, (buf[4] & 0xf0) >> 4);
```

Unchecked Return Value\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=50
Status	New

The dalvik_disassemble method calls the snprintf function, at line 269 of radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Line	506	506
Object	snprintf	snprintf

Code Snippet

File Name	radareorg@@radare2-5.8.0-CVE-2022-28069-FP.c
Method	static int dalvik_disassemble(RAnal *a, RAnalOp *op, ut64 addr, const ut8 *buf, int len, int size) {

```
....
506.                              snprintf (str, sizeof (str), " {v%i, v%i,
v%i}", buf[4] & 0x0f,
```

Sizeof Pointer Argument

Query Path:

CPP\Cx\CPP Low Visibility\Sizeof Pointer Argument Version:0

Description

Sizeof Pointer Argument\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=591
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	307	307
Object	name	sizeof

Code Snippet

File Name	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Method	R_API int r_java_assemble(ut64 addr, ut8 *bytes, const char *string) {

```
....
307.          name[sizeof (name) - 1] = 0;
```

Sizeof Pointer Argument\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=592
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.8.0-CVE-2023-5686-TP.c
Line	307	307
Object	name	sizeof

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2023-5686-TP.c
Method R_API int r_java_assemble(ut64 addr, ut8 *bytes, const char *string) {

```
....
307.          name[sizeof (name) - 1] = 0;
```

Sizeof Pointer Argument\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=593
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c	radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Line	306	306
Object	name	sizeof

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2023-5686-TP.c
Method R_API int r_java_assemble(ut64 addr, ut8 *bytes, const char *string) {

```
....
306.          strncpy (name, string, sizeof (name) - 1);
```

Sizeof Pointer Argument\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=594
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.8.0-CVE-2023-5686-TP.c
Line	306	306
Object	name	sizeof

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2023-5686-TP.c
Method R_API int r_java_assemble(ut64 addr, ut8 *bytes, const char *string) {

```
....
306.          strncpy (name, string, sizeof (name) - 1);
```

NULL Pointer Dereference

Query Path:
CPP\Cx\CPP Low Visibility\NULL Pointer Dereference Version:1

Categories

NIST SP 800-53: SC-5 Denial of Service Protection (P1)
OWASP Top 10 2017: A1-Injection

Description

NULL Pointer Dereference\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=589
Status	New

The variable declared in null at radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c in line 522 is not initialized when it is used by ret at radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c in line 522.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c	radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c
Line	526	526
Object	null	ret

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2023-1605-TP.c
Method static RBinInfo *info(RBinFile *bf) {

```
....
526.         ret->file = bf->file? strdup (bf->file): NULL;
```

NULL Pointer Dereference\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=590
Status	New

The variable declared in null at radareorg@@radare2-5.8.0-CVE-2023-1605-TP.c in line 559 is not initialized when it is used by ret at radareorg@@radare2-5.8.0-CVE-2023-1605-TP.c in line 559.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2023-1605-TP.c	radareorg@@radare2-5.8.0-CVE-2023-1605-TP.c
Line	563	563
Object	null	ret

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2023-1605-TP.c
Method static RBinInfo *info(RBinFile *bf) {

```
....
563.        ret->file = bf->file? strdup (bf->file): NULL;
```

Improper Resource Access Authorization

Query Path:
CPP\Cx\CPP Low Visibility\Improper Resource Access Authorization Version:1

Categories

FISMA 2014: Identification And Authentication
NIST SP 800-53: AC-3 Access Enforcement (P1)
OWASP Top 10 2017: A2-Broken Authentication

Description

Improper Resource Access Authorization\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=956
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	50	50
Object	fwrite	fwrite

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method static bool download_and_write(SPDBDownloaderOpt *opt, const char *file) {

```
....
50.         fwrite (file_buf, sizeof (char), (size_t)len, f);
```

Improper Resource Access Authorization\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=957
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-0520-FP.c	radareorg@@radare2-5.8.0-CVE-2022-0520-FP.c

Line	50	50
Object	fwrite	fwrite

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-0520-FP.c
Method static bool download_and_write(SPDBDownloaderOpt *opt, const char *file) {

```
....
50.          fwrite (file_buf, sizeof (char), (size_t)len, f);
```

Incorrect Permission Assignment For Critical Resources

Query Path:
CPP\Cx\CPP Low Visibility\Incorrect Permission Assignment For Critical Resources Version:1

Categories

FISMA 2014: Access Control
NIST SP 800-53: AC-3 Access Enforcement (P1)
OWASP Top 10 2017: A2-Broken Authentication

Description

Incorrect Permission Assignment For Critical Resources\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=958
Status	New

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	48	48
Object	f	f

Code Snippet
File Name radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Method static bool download_and_write(SPDBDownloaderOpt *opt, const char *file) {

```
....
48.    FILE *f = fopen (path, "wb");
```

Incorrect Permission Assignment For Critical Resources\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=959
Status	New

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-0520-FP.c	radareorg@@radare2-5.8.0-CVE-2022-0520-FP.c
Line	48	48
Object	f	f

Code Snippet

File Name radareorg@@radare2-5.8.0-CVE-2022-0520-FP.c

Method static bool download_and_write(SPDBDownloaderOpt *opt, const char *file) {

```
....
48.    FILE *f = fopen (path, "wb");
```

TOCTOU

Query Path:

CPP\Cx\CPP Low Visibility\TOCTOU Version:1

Description

TOCTOU\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=960
Status	New

The download_and_write method in radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c file utilizes fopen that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c	radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c
Line	48	48
Object	fopen	fopen

Code Snippet

File Name radareorg@@radare2-5.7.4-CVE-2022-0520-FP.c

Method static bool download_and_write(SPDBDownloaderOpt *opt, const char *file) {

```
....
48.    FILE *f = fopen (path, "wb");
```

TOCTOU\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020063&projectid=20052&pathid=961
Status	New

The download_and_write method in radareorg@@radare2-5.8.0-CVE-2022-0520-FP.c file utilizes fopen that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	radareorg@@radare2-5.8.0-CVE-2022-0520-FP.c	radareorg@@radare2-5.8.0-CVE-2022-0520-FP.c
Line	48	48
Object	fopen	fopen

Code Snippet
File Name radareorg@@radare2-5.8.0-CVE-2022-0520-FP.c
Method static bool download_and_write(SPDBDownloaderOpt *opt, const char *file) {

```
....
48.    FILE *f = fopen (path, "wb");
```

Buffer Overflow boundcpy WrongSizeParam

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.
Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

CPP

Overflowing Buffers

```
const int BUFFER_SIZE = 10;
char buffer[BUFFER_SIZE];

void copyStringToBuffer(char* inputString)
{
    strcpy(buffer, inputString);
}
```

Checked Buffers

```
const int BUFFER_SIZE = 10;
const int MAX_INPUT_SIZE = 256;
char buffer[BUFFER_SIZE];

void copyStringToBuffer(char* inputString)
{
    if (strnlen(inputString, MAX_INPUT_SIZE) < sizeof(buffer))
    {
        strncpy(buffer, inputString, sizeof(buffer));
    }
}
```

MemoryFree on StackVariable

Risk

What might happen

Undefined Behavior may result with a crash. Crashes may give an attacker valuable information about the system and the program internals. Furthermore, it may leave unprotected files (e.g memory) that may be exploited.

Cause

How does it happen

Calling free() on a variable that was not dynamically allocated (e.g. malloc) will result with an Undefined Behavior.

General Recommendations

How to avoid it

Use free() only on dynamically allocated variables in order to prevent unexpected behavior from the compiler.

Source Code Examples

CPP

Bad - Calling free() on a static variable

```
void clean_up(){
  char temp[256];
  do_something();
  free(tmp);
  return;
}
```

Good - Calling free() only on variables that were dynamically allocated

```
void clean_up(){
  char *buff;
  buff = (char*) malloc(1024);
  free(buff);
  return;
}
```

Wrong Size t Allocation

Risk

What might happen

Incorrect allocation of memory may result in unexpected behavior by either overwriting sections of memory with unexpected values. Under certain conditions where both an incorrect allocation of memory and the values being written can be controlled by an attacker, such an issue may result in execution of malicious code.

Cause

How does it happen

Some memory allocation functions require a size value to be provided as a parameter. The allocated size should be derived from the provided value, by providing the length value of the intended source, multiplied by the size of that length. Failure to perform the correct arithmetic to obtain the exact size of the value will likely result in the source overflowing its destination.

General Recommendations

How to avoid it

- Always perform the correct arithmetic to determine size.
- Specifically for memory allocation, calculate the allocation size from the allocation source:
    - Derive the size value from the length of intended source to determine the amount of units to be processed.
    - Always programmatically consider the size of the each unit and their conversion to memory units - for example, by using sizeof() on the unit's type.
    - Memory allocation should be a multiplication of the amount of units being written, times the size of each unit.

Source Code Examples

CPP

Allocating and Assigning Memory without Sizeof Arithmetic

```
int *ptr;
ptr = (int*)malloc(5);
for (int i = 0; i < 5; i++)
{
      ptr[i] = i * 2 + 1;
}
```

Allocating and Assigning Memory with Sizeof Arithmetic

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
```

```
for (int i = 0; i < 5; i++)
{
      ptr[i] = i * 2 + 1;
}
```

Incorrect Arithmetic of Multi-Byte String Allocation

```
wchar_t * dest;
dest = (wchar_t *)malloc(wcslen(source) + 1); // Would not crash for a short "source"
wcscpy((wchar_t *)dest, source);
wprintf(L"Dest: %s\r\n", dest);
```

Correct Arithmetic of Multi-Byte String Allocation

```
wchar_t * dest;
dest = (wchar_t *)malloc((wcslen(source) + 1) * sizeof(wchar_t));
wcscpy((wchar_t *)dest, source);
wprintf(L"Dest: %s\r\n", dest);
```

Dangerous Functions

Risk

What might happen

Use of dangerous functions may expose varying risks associated with each particular function, with potential impact of improper usage of these functions varying significantly. The presence of such functions indicates a flaw in code maintenance policies and adherence to secure coding practices, in a way that has allowed introducing known dangerous code into the application.

Cause

How does it happen

A dangerous function has been identified within the code. Functions are often deemed dangerous to use for numerous reasons, as there are different sets of vulnerabilities associated with usage of such functions. For example, some string copy and concatenation functions are vulnerable to Buffer Overflow, Memory Disclosure, Denial of Service and more. Use of these functions is not recommended.

General Recommendations

How to avoid it

- Deploy a secure and recommended alternative to any functions that were identified as dangerous.
    - If no secure alternative is found, conduct further researching and testing to identify whether current usage successfully sanitizes and verifies values, and thus successfully avoids the use-cases for whom the function is indeed dangerous
- Conduct a periodical review of methods that are in use, to ensure that all external libraries and built-in functions are up-to-date and whose use has not been excluded from best secure coding practices.

Source Code Examples

CPP

Buffer Overflow in gets()

```
int main()
{
      char buf[10];

      printf("Please enter your name: ");
      gets(buf); // veryveryverylongname
      if (buf == ACCEPTED_NAME)
      {
             // Do something
      }
      return 0;
}
```

Safe reading from user

```
int main()
{
      char buf[10];

      printf("Please enter your name: ");
      fgets(buf, sizeof(buf), stdin); //setting the amount of bytes to read
      if (buf == ACCEPTED_NAME)
      {
             //Do something
      }
      return 0;
}
```

Unsafe function for string copy

```
int main(int argc, char* argv[])
{
      char buf[10];
      strcpy(buf, argv[1]); // overflow occurs when len(argv[1]) > 10 bytes

      return 0;
}
```

Safe string copy

```
int main(int argc, char* argv[])
{
      char buf[10];
      strncpy(buf, argv[1], sizeof(buf));
      buf[9]= '\0'; //strncpy doesn't NULL terminates

      return 0;
}
```

Unsafe format string

```
int main(int argc, char* argv[])
{
      printf(argv[1]); // If argv[1] contains a format token, such as %s,%x or %d, will cause
an access violation
      return 0;
}
```

Safe format string

```
int main(int argc, char* argv[])
{
      printf("%s", argv[1]); // Second parameter is not a formattable string

      return 0;
}
```

Double Free

Weakness ID: 415 *(Weakness Variant)* **Status:** Draft

Description

Description Summary

The product calls free() twice on the same memory address, potentially leading to modification of unexpected memory locations.

Extended Description

When a program calls free() twice with the same argument, the program's memory management data structures become corrupted. This corruption can cause the program to crash or, in some circumstances, cause two later calls to malloc() to return the same pointer. If malloc() returns the same value twice and the program later gives the attacker control over the data that is written into this doubly-allocated memory, the program becomes vulnerable to a buffer overflow attack.

Alternate Terms

Double-free

Time of Introduction

- Architecture and Design
- Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

Scope	Effect
Access Control	Doubly freeing memory may result in a write-what-where condition, allowing an attacker to execute arbitrary code.

Likelihood of Exploit

Low to Medium

Demonstrative Examples

Example 1

The following code shows a simple example of a double free vulnerability.

(Bad Code)

Example Language: ***C***

```
char* ptr = (char*)malloc (SIZE);
...
if (abrt) {
free(ptr);
}
...
free(ptr);
```

Double free vulnerabilities have two common (and sometimes overlapping) causes:

- Error conditions and other exceptional circumstances
- Confusion over which part of the program is responsible for freeing the memory

Although some double free vulnerabilities are not much more complicated than the previous example, most are spread out across hundreds of lines of code or even different files. Programmers seem particularly susceptible to freeing global variables more than once.

Example 2

While contrived, this code should be exploitable on Linux distributions which do not ship with heap-chunk check summing turned on.

(Bad Code)

Example Language: **C**

```
#include <stdio.h>
#include <unistd.h>
#define BUFSIZE1 512
#define BUFSIZE2 ((BUFSIZE1/2) - 8)

int main(int argc, char **argv) {
char *buf1R1;
char *buf2R1;
char *buf1R2;
buf1R1 = (char *) malloc(BUFSIZE2);
buf2R1 = (char *) malloc(BUFSIZE2);
free(buf1R1);
free(buf2R1);
buf1R2 = (char *) malloc(BUFSIZE1);
strncpy(buf1R2, argv[1], BUFSIZE1-1);
free(buf2R1);
free(buf1R2);
}
```

Observed Examples

Reference	Description
CVE-2004-0642	Double free resultant from certain error conditions.
CVE-2004-0772	Double free resultant from certain error conditions.
CVE-2005-1689	Double free resultant from certain error conditions.
CVE-2003-0545	Double free from invalid ASN.1 encoding.
CVE-2003-1048	Double free from malformed GIF.
CVE-2005-0891	Double free from malformed GIF.
CVE-2002-0059	Double free from malformed compressed data.

Potential Mitigations

Phase: Architecture and Design

Choose a language that provides automatic memory management.

Phase: Implementation

Ensure that each allocation is freed only once. After freeing a chunk, set the pointer to NULL to ensure the pointer cannot be freed again. In complicated error conditions, be sure that clean-up routines respect the state of allocation properly. If the language is object oriented, ensure that object destructors delete each chunk of memory only once.

Phase: Implementation

Use a static analysis tool to find double free instances.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	398	Indicator of Poor Code Quality	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Category	399	Resource Management Errors	**Development Concepts (primary)699**
ChildOf	Category	633	Weaknesses that Affect Memory	**Resource-specific Weaknesses (primary)631**
ChildOf	Weakness Base	666	Operation on Resource in Wrong Phase of	**Research Concepts (primary)1000**

			Lifetime	
ChildOf	Weakness Class	675	Duplicate Operations on Resource	Research Concepts1000
ChildOf	Category	742	CERT C Secure Coding Section 08 - Memory Management (MEM)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
PeerOf	Weakness Base	123	Write-what-where Condition	Research Concepts1000
PeerOf	Weakness Base	416	Use After Free	Development Concepts699 Research Concepts1000
MemberOf	View	630	Weaknesses Examined by SAMATE	**Weaknesses Examined by SAMATE (primary)630**
PeerOf	Weakness Base	364	Signal Handler Race Condition	Research Concepts1000

Relationship Notes

This is usually resultant from another weakness, such as an unhandled error or race condition between threads. It could also be primary to weaknesses such as buffer overflows.

Affected Resources

- Memory

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
PLOVER			DFREE - Double-Free Vulnerability
7 Pernicious Kingdoms			Double Free
CLASP			Doubly freeing memory
CERT C Secure Coding	MEM00-C		Allocate and free memory in the same module, at the same level of abstraction
CERT C Secure Coding	MEM01-C		Store a new value in pointers immediately after free()
CERT C Secure Coding	MEM31-C		Free dynamically allocated memory exactly once

White Box Definitions

A weakness where code path has:

1. start statement that relinquishes a dynamically allocated memory resource
2. end statement that relinquishes the dynamically allocated memory resource

Maintenance Notes

It could be argued that Double Free would be most appropriately located as a child of "Use after Free", but "Use" and "Release" are considered to be distinct operations within vulnerability theory, therefore this is more accurately "Release of a Resource after Expiration or Release", which doesn't exist yet.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	PLOVER		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Potential Mitigations, Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Description, Maintenance Notes, Relationships, Other Notes, Relationship Notes, Taxonomy Mappings		
2008-11-24	CWE Content Team	MITRE	Internal

	updated Relationships, Taxonomy Mappings		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Other Notes		

BACK TO TOP

Failure to Release Memory Before Removing Last Reference ('Memory Leak')

Weakness ID: 401 *(Weakness Base)* **Status:** Draft

Description

Description Summary

The software does not sufficiently track and release allocated memory after it has been used, which slowly consumes remaining memory.

Extended Description

This is often triggered by improper handling of malformed data or unexpectedly interrupted sessions.

Terminology Notes

"memory leak" has sometimes been used to describe other kinds of issues, e.g. for information leaks in which the contents of memory are inadvertently leaked (CVE-2003-0400 is one such example of this terminology conflict).

Time of Introduction

- Architecture and Design
- Implementation

Applicable Platforms

Languages

C

C++

Modes of Introduction

Memory leaks have two common and sometimes overlapping causes:

- Error conditions and other exceptional circumstances
- Confusion over which part of the program is responsible for freeing the memory

Common Consequences

Scope	Effect
Availability	Most memory leaks result in general software reliability problems, but if an attacker can intentionally trigger a memory leak, the attacker might be able to launch a denial of service attack (by crashing or hanging the program) or take advantage of other unexpected program behavior resulting from a low memory condition.

Likelihood of Exploit

Medium

Demonstrative Examples

Example 1

The following C function leaks a block of allocated memory if the call to read() fails to return the expected number of bytes:

(Bad Code)

Example Language: **C**

```
char* getBlock(int fd) {
char* buf = (char*) malloc(BLOCK_SIZE);
if (!buf) {
return NULL;
}
if (read(fd, buf, BLOCK_SIZE) != BLOCK_SIZE) {

return NULL;
}
return buf;
}
```

Example 2

Here the problem is that every time a connection is made, more memory is allocated. So if one just opened up more and more connections, eventually the machine would run out of memory.

(Bad Code)

Example Language: ***C***

```
bar connection(){
foo = malloc(1024);
return foo;
}
endConnection(bar foo) {

free(foo);
}
int main() {

while(1) //thread 1
//On a connection
foo=connection(); //thread 2
//When the connection ends
endConnection(foo)
}
```

Observed Examples

Reference	Description
CVE-2005-3119	Memory leak because function does not free() an element of a data structure.
CVE-2004-0427	Memory leak when counter variable is not decremented.
CVE-2002-0574	Memory leak when counter variable is not decremented.
CVE-2005-3181	Kernel uses wrong function to release a data structure, preventing data from being properly tracked by other code.
CVE-2004-0222	Memory leak via unknown manipulations as part of protocol test suite.
CVE-2001-0136	Memory leak via a series of the same command.

Potential Mitigations

Pre-design: Use a language or compiler that performs automatic bounds checking.

Phase: Architecture and Design

Use an abstraction library to abstract away risky APIs. Not a complete solution.

Pre-design through Build: The Boehm-Demers-Weiser Garbage Collector or valgrind can be used to detect leaks in code. This is not a complete solution as it is not 100% effective.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	398	Indicator of Poor Code Quality	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Category	399	Resource Management Errors	**Development Concepts (primary)699**
ChildOf	Category	633	Weaknesses that Affect Memory	**Resource-specific Weaknesses (primary)631**
ChildOf	Category	730	OWASP Top Ten 2004 Category A9 - Denial of Service	**Weaknesses in OWASP Top Ten (2004) (primary)711**
ChildOf	Weakness Base	772	Missing Release of Resource after Effective	**Research Concepts (primary)1000**

			Lifetime	
MemberOf	View	630	Weaknesses Examined by SAMATE	**Weaknesses Examined by SAMATE (primary)630**
CanFollow	Weakness Class	390	Detection of Error Condition Without Action	Research Concepts1000

Relationship Notes

This is often a resultant weakness due to improper handling of malformed data or early termination of sessions.

Affected Resources

- Memory

Functional Areas

- Memory management

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
PLOVER			Memory leak
7 Pernicious Kingdoms			Memory Leak
CLASP			Failure to deallocate data
OWASP Top Ten 2004	A9	CWE More Specific	Denial of Service

White Box Definitions

A weakness where the code path has:

1. start statement that allocates dynamically allocated memory resource

2. end statement that loses identity of the dynamically allocated memory resource creating situation where dynamically allocated memory resource is never relinquished

Where "loses" is defined through the following scenarios:

1. identity of the dynamic allocated memory resource never obtained

2. the statement assigns another value to the data element that stored the identity of the dynamically allocated memory resource and there are no aliases of that data element

3. identity of the dynamic allocated memory resource obtained but never passed on to function for memory resource release

4. the data element that stored the identity of the dynamically allocated resource has reached the end of its scope at the statement and there are no aliases of that data element

References

J. Whittaker and H. Thompson. "How to Break Software Security". Addison Wesley. 2003.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	PLOVER		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-08-15		Veracode	External
	Suggested OWASP Top Ten 2004 mapping		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Relationships, Other Notes, References, Relationship Notes, Taxonomy Mappings, Terminology Notes		
2008-10-14	CWE Content Team	MITRE	Internal
	updated Description		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Other Notes		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Name		
2009-07-17	KDM Analytics		External
	Improved the White Box Definition		

2009-07-27	CWE Content Team	MITRE	Internal
	updated White Box Definitions		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Modes of Introduction, Other Notes		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		

Previous Entry Names

Change Date	Previous Entry Name
2008-04-11	Memory Leak
2009-05-27	Failure to Release Memory Before Removing Last Reference (aka 'Memory Leak')

BACK TO TOP

Use of Zero Initialized Pointer

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
- Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
- Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.

Source Code Examples

Wrong Memory Allocation

Risk

What might happen

Incorrect allocation of memory may result in unexpected behavior by either overwriting sections of memory with unexpected values. Under certain conditions where both an incorrect allocation of memory and the values being written can be controlled by an attacker, such an issue may result in execution of malicious code.

Cause

How does it happen

Some memory allocation functions require a size value to be provided as a parameter. The allocated size should be derived from the provided value, by providing the length value of the intended source, multiplied by the size of that length. Failure to perform the correct arithmetic to obtain the exact size of the value will likely result in the source overflowing its destination.

General Recommendations

How to avoid it

- Always perform the correct arithmetic to determine size.
- Specifically for memory allocation, calculate the allocation size from the allocation source:
  - Derive the size value from the length of intended source to determine the amount of units to be processed.
  - Always programmatically consider the size of the each unit and their conversion to memory units - for example, by using sizeof() on the unit's type.
  - Memory allocation should be a multiplication of the amount of units being written, times the size of each unit.

Source Code Examples

Unchecked Return Value

Risk

What might happen

A program that does not check function return values could cause the application to enter an undefined state. This could lead to unexpected behavior and unintended consequences, including inconsistent data, system crashes or other error-based exploits.

Cause

How does it happen

The application calls a system function, but does not receive or check the result of this funciton. These functions often return error codes in the result, or share other status codes with it's caller. The application simply ignores this result value, losing this vital information.

General Recommendations

How to avoid it

- Always check the result of any called function that returns a value, and verify the result is an expected value.

- Ensure the calling function responds to all possible return values.

- Expect runtime errors and handle them gracefully. Explicitly define a mechanism for handling unexpected errors.

Source Code Examples

CPP

Unchecked Memory Allocation

```
buff = (char*) malloc(size);
strncpy(buff, source, size);
```

Safer Memory Allocation

```
buff = (char*) malloc(size+1);
if (buff==NULL) exit(1);

strncpy(buff, source, size);
buff[size] = '\0';
```

NULL Pointer Dereference

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
- Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
- Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.

Source Code Examples

CPP

Explicit NULL Dereference

```
char * input = NULL;
printf("%s", input);
```

Implicit NULL Dereference

```
char * input;
printf("%s", input);
```

Java

Explicit Null Dereference

```
Object o = null;
out.println(o.getClass());
```

Use of sizeof() on a Pointer Type

Weakness ID: 467 *(Weakness Variant)* **Status:** Draft

Description

Description Summary

The code calls sizeof() on a malloced pointer type, which always returns the wordsize/8. This can produce an unexpected result if the programmer intended to determine how much memory has been allocated.

Time of Introduction

- Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

Scope	Effect
Integrity	This error can often cause one to allocate a buffer that is much smaller than what is needed, leading to resultant weaknesses such as buffer overflows.

Likelihood of Exploit

High

Demonstrative Examples

Example 1

Care should be taken to ensure sizeof returns the size of the data structure itself, and not the size of the pointer to the data structure.

In this example, sizeof(foo) returns the size of the pointer.

(Bad Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(foo));
```

In this example, sizeof(*foo) returns the size of the data structure and not the size of the pointer.

(Good Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(*foo));
```

Example 2

This example defines a fixed username and password. The AuthenticateUser() function is intended to accept a username and a password from an untrusted user, and check to ensure that it matches the username and password. If the username and password match, AuthenticateUser() is intended to indicate that authentication succeeded.

(Bad Code)

```
/* Ignore CWE-259 (hard-coded password) and CWE-309 (use of password system for authentication) for this example. */

char *username = "admin";
char *pass = "password";

int AuthenticateUser(char *inUser, char *inPass) {
```

```
printf("Sizeof username = %d\n", sizeof(username));
printf("Sizeof pass = %d\n", sizeof(pass));

if (strncmp(username, inUser, sizeof(username))) {
printf("Auth failure of username using sizeof\n");
return(AUTH_FAIL);
}
/* Because of CWE-467, the sizeof returns 4 on many platforms and architectures. */
if (! strncmp(pass, inPass, sizeof(pass))) {
printf("Auth success of password using sizeof\n");
return(AUTH_SUCCESS);
}
else {
printf("Auth fail of password using sizeof\n");
return(AUTH_FAIL);
}
}

int main (int argc, char **argv)
{
int authResult;

if (argc < 3) {
ExitError("Usage: Provide a username and password");
}
authResult = AuthenticateUser(argv[1], argv[2]);
if (authResult != AUTH_SUCCESS) {
ExitError("Authentication failed");
}
else {
DoAuthenticatedTask(argv[1]);
}
}
```

In AuthenticateUser(), because sizeof() is applied to a parameter with an array type, the sizeof() call might return 4 on many modern architectures. As a result, the strncmp() call only checks the first four characters of the input password, resulting in a partial comparison (CWE-187), leading to improper authentication (CWE-287).

Because of the partial comparison, any of these passwords would still cause authentication to succeed for the "admin" user:

(Attack)

```
pass5
passABCDEFGH
passWORD
```

Because only 4 characters are checked, this significantly reduces the search space for an attacker, making brute force attacks more feasible.

The same problem also applies to the username, so values such as "adminXYZ" and "administrator" will succeed for the username.

Potential Mitigations

Phase: Implementation

Use expressions such as "sizeof(*pointer)" instead of "sizeof(pointer)", unless you intend to run sizeof() on a pointer type to gain some platform independence or if you are allocating a variable on the stack.

Other Notes

The use of sizeof() on a pointer can sometimes generate useful information. An obvious case is to find out the wordsize on a platform. More often than not, the appearance of sizeof(pointer) indicates a bug.

Weakness Ordinalities

Ordinality	Description
Primary	*(where the weakness exists independent of other weaknesses)*

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	465	Pointer Issues	**Development Concepts (primary)699**
ChildOf	Weakness Class	682	Incorrect Calculation	**Research Concepts (primary)1000**
ChildOf	Category	737	CERT C Secure Coding Section 03 - Expressions (EXP)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
ChildOf	Category	740	CERT C Secure Coding Section 06 - Arrays (ARR)	Weaknesses Addressed by the CERT C Secure Coding Standard734
CanPrecede	Weakness Base	131	Incorrect Calculation of Buffer Size	Research Concepts1000

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Use of sizeof() on a pointer type
CERT C Secure Coding	ARR01-C		Do not apply the sizeof operator to a pointer when taking the size of an array
CERT C Secure Coding	EXP01-C		Do not take the size of a pointer to determine the size of the pointed-to type

White Box Definitions

A weakness where code path has:

1. end statement that passes an identity of a dynamically allocated memory resource to a sizeof operator
2. start statement that allocates the dynamically allocated memory resource

References

Robert Seacord. "EXP01-A. Do not take the sizeof a pointer to determine the size of a type". <https://www.securecoding.cert.org/confluence/display/seccode/EXP01-A.+Do+not+take+the+sizeof+a+pointer+to+determine+the+size+of+a+type>.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	CLASP		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		

BACK TO TOP

Improper Access Control (Authorization)

Weakness ID: 285 *(Weakness Class)* **Status:** Draft

Description

Description Summary

The software does not perform or incorrectly performs access control checks across all potential execution paths.

Extended Description

When access control checks are not applied consistently - or not at all - users are able to access data or perform actions that they should not be allowed to perform. This can lead to a wide range of problems, including information leaks, denial of service, and arbitrary code execution.

Alternate Terms

AuthZ:	"AuthZ" is typically used as an abbreviation of "authorization" within the web application security community. It is also distinct from "AuthC," which is an abbreviation of "authentication." The use of "Auth" as an abbreviation is discouraged, since it could be used for either authentication or authorization.

Time of Introduction

- Architecture and Design
- Implementation
- Operation

Applicable Platforms

Languages

Language-independent

Technology Classes

Web-Server: *(Often)*

Database-Server: *(Often)*

Modes of Introduction

A developer may introduce authorization weaknesses because of a lack of understanding about the underlying technologies. For example, a developer may assume that attackers cannot modify certain inputs such as headers or cookies.

Authorization weaknesses may arise when a single-user application is ported to a multi-user environment.

Common Consequences

Scope	Effect
Confidentiality	An attacker could read sensitive data, either by reading the data directly from a data store that is not properly restricted, or by accessing insufficiently-protected, privileged functionality to read the data.
Integrity	An attacker could modify sensitive data, either by writing the data directly to a data store that is not properly restricted, or by accessing insufficiently-protected, privileged functionality to write the data.
Integrity	An attacker could gain privileges by modifying or reading critical data directly, or by accessing insufficiently-protected, privileged functionality.

Likelihood of Exploit

High

Detection Methods

Automated Static Analysis

Automated static analysis is useful for detecting commonly-used idioms for authorization. A tool may be able to analyze related configuration files, such as .htaccess in Apache web servers, or detect the usage of commonly-used authorization libraries.

Generally, automated static analysis tools have difficulty detecting custom authorization schemes. In addition, the software's design may include some functionality that is accessible to any user and does not require an authorization check; an automated technique that detects the absence of authorization may report false positives.

Effectiveness: Limited

Automated Dynamic Analysis

Automated dynamic analysis may find many or all possible interfaces that do not require authorization, but manual analysis is required to determine if the lack of authorization violates business logic

Manual Analysis

This weakness can be detected using tools and techniques that require manual (human) analysis, such as penetration testing, threat modeling, and interactive tools that allow the tester to record and modify an active session.

Specifically, manual static analysis is useful for evaluating the correctness of custom authorization mechanisms.

Effectiveness: Moderate

These may be more effective than strictly automated techniques. This is especially the case with weaknesses that are related to design and business rules. However, manual efforts might not achieve desired code coverage within limited time constraints.

Demonstrative Examples

Example 1

The following program could be part of a bulletin board system that allows users to send private messages to each other. This program intends to authenticate the user before deciding whether a private message should be displayed. Assume that LookupMessageObject() ensures that the $id argument is numeric, constructs a filename based on that id, and reads the message details from that file. Also assume that the program stores all private messages for all users in the same directory.

(Bad Code)

Example Language: **Perl**

```
sub DisplayPrivateMessage {
my($id) = @_;
my $Message = LookupMessageObject($id);
print "From: " . encodeHTML($Message->{from}) . "<br>\n";
print "Subject: " . encodeHTML($Message->{subject}) . "\n";
print "<hr>\n";
print "Body: " . encodeHTML($Message->{body}) . "\n";
}

my $q = new CGI;
# For purposes of this example, assume that CWE-309 and
# CWE-523 do not apply.
if (! AuthenticateUser($q->param('username'), $q->param('password'))) {
ExitError("invalid username or password");
}

my $id = $q->param('id');
DisplayPrivateMessage($id);
```

While the program properly exits if authentication fails, it does not ensure that the message is addressed to the user. As a result, an authenticated attacker could provide any arbitrary identifier and read private messages that were intended for other users.

One way to avoid this problem would be to ensure that the "to" field in the message object matches the username of the authenticated user.

Observed Examples

Reference	Description
CVE-2009-3168	Web application does not restrict access to admin scripts, allowing authenticated users to reset administrative passwords.

CVE-2009-2960	Web application does not restrict access to admin scripts, allowing authenticated users to modify passwords of other users.
CVE-2009-3597	Web application stores database file under the web root with insufficient access control (CWE-219), allowing direct request.
CVE-2009-2282	Terminal server does not check authorization for guest access.
CVE-2009-3230	Database server does not use appropriate privileges for certain sensitive operations.
CVE-2009-2213	Gateway uses default "Allow" configuration for its authorization settings.
CVE-2009-0034	Chain: product does not properly interpret a configuration option for a system group, allowing users to gain privileges.
CVE-2008-6123	Chain: SNMP product does not properly parse a configuration option for which hosts are allowed to connect, allowing unauthorized IP addresses to connect.
CVE-2008-5027	System monitoring software allows users to bypass authorization by creating custom forms.
CVE-2008-7109	Chain: reliance on client-side security (CWE-602) allows attackers to bypass authorization using a custom client.
CVE-2008-3424	Chain: product does not properly handle wildcards in an authorization policy list, allowing unintended access.
CVE-2009-3781	Content management system does not check access permissions for private files, allowing others to view those files.
CVE-2008-4577	ACL-based protection mechanism treats negative access rights as if they are positive, allowing bypass of intended restrictions.
CVE-2008-6548	Product does not check the ACL of a page accessed using an "include" directive, allowing attackers to read unauthorized files.
CVE-2007-2925	Default ACL list for a DNS server does not set certain ACLs, allowing unauthorized DNS queries.
CVE-2006-6679	Product relies on the X-Forwarded-For HTTP header for authorization, allowing unintended access by spoofing the header.
CVE-2005-3623	OS kernel does not check for a certain privilege before setting ACLs for files.
CVE-2005-2801	Chain: file-system code performs an incorrect comparison (CWE-697), preventing defauls ACLs from being properly applied.
CVE-2001-1155	Chain: product does not properly check the result of a reverse DNS lookup because of operator precedence (CWE-783), allowing bypass of DNS-based access restrictions.

Potential Mitigations

Phase: Architecture and Design

Divide your application into anonymous, normal, privileged, and administrative areas. Reduce the attack surface by carefully mapping roles with data and functionality. Use role-based access control (RBAC) to enforce the roles at the appropriate boundaries.

Note that this approach may not protect against horizontal authorization, i.e., it will not protect a user from attacking others with the same role.

Phase: Architecture and Design

Ensure that you perform access control checks related to your business logic. These checks may be different than the access control checks that you apply to more generic resources such as files, connections, processes, memory, and database records. For example, a database may restrict access for medical records to a specific database user, but each record might only be intended to be accessible to the patient and the patient's doctor.

Phase: Architecture and Design

Strategy: Libraries or Frameworks

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.

For example, consider using authorization frameworks such as the JAAS Authorization Framework and the OWASP ESAPI Access Control feature.

Phase: Architecture and Design

For web applications, make sure that the access control mechanism is enforced correctly at the server side on every page. Users should not be able to access any unauthorized functionality or information by simply requesting direct access to that page.

One way to do this is to ensure that all pages containing sensitive information are not cached, and that all such pages restrict access to requests that are accompanied by an active and authenticated session token associated with a user who has the required permissions to access that page.

Phases: System Configuration; Installation

Use the access control capabilities of your operating system and server environment and define your access control lists accordingly. Use a "default deny" policy when defining these ACLs.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	254	Security Features	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Weakness Class	284	Access Control (Authorization) Issues	**Development Concepts (primary)699** **Research Concepts (primary)1000**
ChildOf	Category	721	OWASP Top Ten 2007 Category A10 - Failure to Restrict URL Access	**Weaknesses in OWASP Top Ten (2007) (primary)629**
ChildOf	Category	723	OWASP Top Ten 2004 Category A2 - Broken Access Control	**Weaknesses in OWASP Top Ten (2004) (primary)711**
ChildOf	Category	753	2009 Top 25 - Porous Defenses	**Weaknesses in the 2009 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)750**
ChildOf	Category	803	2010 Top 25 - Porous Defenses	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
ParentOf	Weakness Variant	219	Sensitive Data Under Web Root	**Research Concepts (primary)1000**
ParentOf	Weakness Base	551	Incorrect Behavior Order: Authorization Before Parsing and Canonicalization	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Class	638	Failure to Use Complete Mediation	Research Concepts1000
ParentOf	Weakness Base	804	Guessable CAPTCHA	**Development Concepts (primary)699** **Research Concepts (primary)1000**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
7 Pernicious Kingdoms			Missing Access Control
OWASP Top Ten 2007	A10	CWE More Specific	Failure to Restrict URL Access
OWASP Top Ten 2004	A2	CWE More Specific	Broken Access Control

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
1	Accessing Functionality Not Properly Constrained by ACLs	
13	Subverting Environment Variable Values	

17	Accessing, Modifying or Executing Executable Files
87	Forceful Browsing
39	Manipulating Opaque Client-based Data Tokens
45	Buffer Overflow via Symbolic Links
51	Poison Web Service Registry
59	Session Credential Falsification through Prediction
60	Reusing Session IDs (aka Session Replay)
77	Manipulating User-Controlled Variables
76	Manipulating Input to File System Calls
104	Cross Zone Scripting

References

NIST. "Role Based Access Control and Role Based Security". <http://csrc.nist.gov/groups/SNS/rbac/>.

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 4, "Authorization" Page 114; Chapter 6, "Determining Appropriate Access Control" Page 171. 2nd Edition. Microsoft. 2002.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	7 Pernicious Kingdoms		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-15		Veracode	External
	Suggested OWASP Top Ten 2004 mapping		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Relationships, Other Notes, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences, Description, Likelihood of Exploit, Name, Other Notes, Potential Mitigations, References, Relationships		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Potential Mitigations		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Description, Related Attack Patterns		
2009-07-27	CWE Content Team	MITRE	Internal
	updated Relationships		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Type		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Demonstrative Examples, Detection Factors, Modes of Introduction, Observed Examples, Relationships		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Alternate Terms, Detection Factors, Potential Mitigations, References, Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Potential Mitigations		

Previous Entry Names

Change Date	Previous Entry Name
2009-01-12	Missing or Inconsistent Access Control

BACK TO TOP

Incorrect Permission Assignment for Critical Resource

Weakness ID: 732 *(Weakness Class)* **Status:** Draft

Description

Description Summary

The software specifies permissions for a security-critical resource in a way that allows that resource to be read or modified by unintended actors.

Extended Description

When a resource is given a permissions setting that provides access to a wider range of actors than required, it could lead to the disclosure of sensitive information, or the modification of that resource by unintended parties. This is especially dangerous when the resource is related to program configuration, execution or sensitive user data.

Time of Introduction

- Architecture and Design
- Implementation
- Installation
- Operation

Applicable Platforms

Languages

Language-independent

Modes of Introduction

The developer may set loose permissions in order to minimize problems when the user first runs the program, then create documentation stating that permissions should be tightened. Since system administrators and users do not always read the documentation, this can result in insecure permissions being left unchanged.

The developer might make certain assumptions about the environment in which the software runs - e.g., that the software is running on a single-user system, or the software is only accessible to trusted administrators. When the software is running in a different environment, the permissions become a problem.

Common Consequences

Scope	Effect
Confidentiality	An attacker may be able to read sensitive information from the associated resource, such as credentials or configuration information stored in a file.
Integrity	An attacker may be able to modify critical properties of the associated resource to gain privileges, such as replacing a world-writable executable with a Trojan horse.
Availability	An attacker may be able to destroy or corrupt critical data in the associated resource, such as deletion of records from a database.

Likelihood of Exploit

Medium to High

Detection Methods

Automated Static Analysis

Automated static analysis may be effective in detecting permission problems for system resources such as files, directories, shared memory, device interfaces, etc. Automated techniques may be able to detect the use of library functions that modify permissions, then analyze function calls for arguments that contain potentially insecure values.

However, since the software's intended security policy might allow loose permissions for certain operations (such as publishing a file on a web server), automated static analysis may produce some false positives - i.e., warnings that do not have any security consequences or require any code changes.

When custom permissions models are used - such as defining who can read messages in a particular forum in a bulletin board system - these can be difficult to detect using automated static analysis. It may be possible to define custom signatures that identify any custom functions that implement the permission checks and assignments.

Automated Dynamic Analysis

Automated dynamic analysis may be effective in detecting permission problems for system resources such as files, directories, shared memory, device interfaces, etc.

However, since the software's intended security policy might allow loose permissions for certain operations (such as publishing a file on a web server), automated dynamic analysis may produce some false positives - i.e., warnings that do not have any security consequences or require any code changes.

When custom permissions models are used - such as defining who can read messages in a particular forum in a bulletin board system - these can be difficult to detect using automated dynamic analysis. It may be possible to define custom signatures that identify any custom functions that implement the permission checks and assignments.

Manual Static Analysis

Manual static analysis may be effective in detecting the use of custom permissions models and functions. The code could then be examined to identifying usage of the related functions. Then the human analyst could evaluate permission assignments in the context of the intended security model of the software.

Manual Dynamic Analysis

Manual dynamic analysis may be effective in detecting the use of custom permissions models and functions. The program could then be executed with a focus on exercising code paths that are related to the custom permissions. Then the human analyst could evaluate permission assignments in the context of the intended security model of the software.

Fuzzing

Fuzzing is not effective in detecting this weakness.

Demonstrative Examples

Example 1

The following code sets the umask of the process to 0 before creating a file and writing "Hello world" into the file.

(Bad Code)

Example Language: **C**

```
#define OUTFILE "hello.out"

umask(0);
FILE *out;
/* Ignore CWE-59 (link following) for brevity */
out = fopen(OUTFILE, "w");
if (out) {
fprintf(out, "hello world!\n");
fclose(out);
}
```

After running this program on a UNIX system, running the "ls -l" command might return the following output:

(Result)

```
-rw-rw-rw- 1 username 13 Nov 24 17:58 hello.out
```

The "rw-rw-rw-" string indicates that the owner, group, and world (all users) can read the file and write to it.

Example 2

The following code snippet might be used as a monitor to periodically record whether a web site is alive. To ensure that the file can always be modified, the code uses chmod() to make the file world-writable.

(Bad Code)

Example Language: **Perl**

```
$fileName = "secretFile.out";

if (-e $fileName) {
chmod 0777, $fileName;
}
```

```
my $outFH;
if (! open($outFH, ">>$fileName")) {
ExitError("Couldn't append to $fileName: $!");
}
my $dateString = FormatCurrentTime();
my $status = IsHostAlive("cwe.mitre.org");
print $outFH "$dateString cwe status: $status!\n";
close($outFH);
```

The first time the program runs, it might create a new file that inherits the permissions from its environment. A file listing might look like:

(Result)

```
-rw-r--r-- 1 username 13 Nov 24 17:58 secretFile.out
```

This listing might occur when the user has a default umask of 022, which is a common setting. Depending on the nature of the file, the user might not have intended to make it readable by everyone on the system.

The next time the program runs, however - and all subsequent executions - the chmod will set the file's permissions so that the owner, group, and world (all users) can read the file and write to it:

(Result)

```
-rw-rw-rw- 1 username 13 Nov 24 17:58 secretFile.out
```

Perhaps the programmer tried to do this because a different process uses different permissions that might prevent the file from being updated.

Example 3

The following command recursively sets world-readable permissions for a directory and all of its children:

(Bad Code)

Example Language: **Shell**

```
chmod -R ugo+r DIRNAME
```

If this command is run from a program, the person calling the program might not expect that all the files under the directory will be world-readable. If the directory is expected to contain private data, this could become a security problem.

Observed Examples

Reference	Description
CVE-2009-3482	Anti-virus product sets insecure "Everyone: Full Control" permissions for files under the "Program Files" folder, allowing attackers to replace executables with Trojan horses.
CVE-2009-3897	Product creates directories with 0777 permissions at installation, allowing users to gain privileges and access a socket used for authentication.
CVE-2009-3489	Photo editor installs a service with an insecure security descriptor, allowing users to stop or start the service, or execute commands as SYSTEM.
CVE-2009-3289	Library function copies a file to a new target and uses the source file's permissions for the target, which is incorrect when the source file is a symbolic link, which typically has 0777 permissions.
CVE-2009-0115	Device driver uses world-writable permissions for a socket file, allowing attackers to inject arbitrary commands.
CVE-2009-1073	LDAP server stores a cleartext password in a world-readable file.
CVE-2009-0141	Terminal emulator creates TTY devices with world-writable permissions, allowing an attacker to write to the terminals of other users.

CVE-2008-0662	VPN product stores user credentials in a registry key with "Everyone: Full Control" permissions, allowing attackers to steal the credentials.
CVE-2008-0322	Driver installs its device interface with "Everyone: Write" permissions.
CVE-2009-3939	Driver installs a file with world-writable permissions.
CVE-2009-3611	Product changes permissions to 0777 before deleting a backup; the permissions stay insecure for subsequent backups.
CVE-2007-6033	Product creates a share with "Everyone: Full Control" permissions, allowing arbitrary program execution.
CVE-2007-5544	Product uses "Everyone: Full Control" permissions for memory-mapped files (shared memory) in inter-process communication, allowing attackers to tamper with a session.
CVE-2005-4868	Database product uses read/write permissions for everyone for its shared memory, allowing theft of credentials.
CVE-2004-1714	Security product uses "Everyone: Full Control" permissions for its configuration files.
CVE-2001-0006	"Everyone: Full Control" permissions assigned to a mutex allows users to disable network connectivity.
CVE-2002-0969	Chain: database product contains buffer overflow that is only reachable through a .ini configuration file - which has "Everyone: Full Control" permissions.

Potential Mitigations

Phase: Implementation

When using a critical resource such as a configuration file, check to see if the resource has insecure permissions (such as being modifiable by any regular user), and generate an error or even exit the software if there is a possibility that the resource could have been modified by an unauthorized party.

Phase: Architecture and Design

Divide your application into anonymous, normal, privileged, and administrative areas. Reduce the attack surface by carefully defining distinct user groups, privileges, and/or roles. Map these against data, functionality, and the related resources. Then set the permissions accordingly. This will allow you to maintain more fine-grained control over your resources.

Phases: Implementation; Installation

During program startup, explicitly set the default permissions or umask to the most restrictive setting possible. Also set the appropriate permissions during program installation. This will prevent you from inheriting insecure permissions from any user who installs or runs the program.

Phase: System Configuration

For all configuration files, executables, and libraries, make sure that they are only readable and writable by the software's administrator.

Phase: Documentation

Do not suggest insecure configuration changes in your documentation, especially if those configurations can extend to resources and other software that are outside the scope of your own software.

Phase: Installation

Do not assume that the system administrator will manually change the configuration to the settings that you recommend in the manual.

Phase: Testing

Use tools and techniques that require manual (human) analysis, such as penetration testing, threat modeling, and interactive tools that allow the tester to record and modify an active session. These may be more effective than strictly automated techniques. This is especially the case with weaknesses that are related to design and business rules.

Phase: Testing

Use monitoring tools that examine the software's process as it interacts with the operating system and the network. This technique is useful in cases when source code is unavailable, if the software was not developed by you, or if you want to verify that the build phase did not introduce any new weaknesses. Examples include debuggers that directly attach to the running process; system-call tracing utilities such as truss (Solaris) and strace (Linux); system activity monitors such as FileMon, RegMon, Process Monitor, and other Sysinternals utilities (Windows); and sniffers and protocol analyzers that monitor network traffic.

Attach the monitor to the process and watch for library functions or system calls on OS resources such as files, directories, and shared memory. Examine the arguments to these calls to infer which permissions are being used.

Note that this technique is only useful for permissions issues related to system resources. It is not likely to detect application-level business rules that are related to permissions, such as if a user of a blog system marks a post as "private," but the blog system inadvertently marks it as "public."

Phases: Testing; System Configuration

Ensure that your software runs properly under the Federal Desktop Core Configuration (FDCC) or an equivalent hardening configuration guide, which many organizations use to limit the attack surface and potential risk of deployed software.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	275	Permission Issues	**Development Concepts (primary)699**
ChildOf	Weakness Class	668	Exposure of Resource to Wrong Sphere	**Research Concepts (primary)1000**
ChildOf	Category	753	2009 Top 25 - Porous Defenses	**Weaknesses in the 2009 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)750**
ChildOf	Category	803	2010 Top 25 - Porous Defenses	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
RequiredBy	Compound Element: Composite	689	Permission Race Condition During Resource Copy	Research Concepts1000
ParentOf	Weakness Variant	276	Incorrect Default Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	277	Insecure Inherited Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	278	Insecure Preserved Inherited Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	279	Incorrect Execution-Assigned Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Base	281	Improper Preservation of Permissions	**Research Concepts (primary)1000**

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
232	Exploitation of Privilege/Trust	
1	Accessing Functionality Not Properly Constrained by ACLs	
17	Accessing, Modifying or Executing Executable Files	
60	Reusing Session IDs (aka Session Replay)	
61	Session Fixation	
62	Cross Site Request Forgery (aka Session Riding)	
122	Exploitation of Authorization	
180	Exploiting Incorrectly Configured Access Control Security Levels	
234	Hijacking a privileged process	

References

Mark Dowd, John McDonald and Justin Schuh. "The Art of Software Security Assessment". Chapter 9, "File Permissions." Page 495.. 1st Edition. Addison Wesley. 2006.

John Viega and Gary McGraw. "Building Secure Software". Chapter 8, "Access Control." Page 194.. 1st Edition. Addison-Wesley. 2002.

Maintenance Notes

The relationships between privileges, permissions, and actors (e.g. users and groups) need further refinement within the Research view. One complication is that these concepts apply to two different pillars, related to control of resources (CWE-664) and protection mechanism failures (CWE-396).

Content History

Submissions

Submission Date	Submitter	Organization	Source
2008-09-08			Internal CWE Team
	new weakness-focused entry for Research view.		

Modifications

Modification Date	Modifier	Organization	Source
2009-01-12	CWE Content Team	MITRE	Internal
	updated Description, Likelihood of Exploit, Name, Potential Mitigations, Relationships		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Potential Mitigations, Related Attack Patterns		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Name		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Demonstrative Examples, Detection Factors, Modes of Introduction, Observed Examples, Potential Mitigations, References		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Potential Mitigations, Related Attack Patterns		

Previous Entry Names

Change Date	Previous Entry Name
2009-01-12	Insecure Permission Assignment for Resource
2009-05-27	Insecure Permission Assignment for Critical Resource

BACK TO TOP

TOCTOU

Risk

What might happen

At best, a Race Condition may cause errors in accuracy, overidden values or unexpected behavior that may result in denial-of-service. At worst, it may allow attackers to retrieve data or bypass security processes by replaying a controllable Race Condition until it plays out in their favor.

Cause

How does it happen

Race Conditions occur when a public, single instance of a resource is used by multiple concurrent logical processes. If the these logical processes attempt to retrieve and update the resource without a timely management system, such as a lock, a Race Condition will occur.

An example for when a Race Condition occurs is a resource that may return a certain value to a process for further editing, and then updated by a second process, resulting in the original process' data no longer being valid. Once the original process edits and updates the incorrect value back into the resource, the second process' update has been overwritten and lost.

General Recommendations

How to avoid it

When sharing resources between concurrent processes across the application ensure that these resources are either thread-safe, or implement a locking mechanism to ensure expected concurrent activity.

Source Code Examples

Java

Different Threads Increment and Decrement The Same Counter Repeatedly, Resulting in a Race Condition

```
      public static int counter = 0;
      public static void start() throws InterruptedException {
             incrementCounter ic;
             decrementCounter dc;
             while(counter == 0) {
                    counter = 0;
                    ic = new incrementCounter();
                    dc = new decrementCounter();
                    ic.start();
                    dc.start();
                    ic.join();
                    dc.join();
             }
             System.out.println(counter); //Will stop and return either -1 or 1 due to race
condition over counter
      }

      public static class incrementCounter extends Thread {
          public void run() {
             counter++;
          }
      }

      public static class decrementCounter extends Thread {
          public void run() {
             counter--;
          }
      }
```

Different Threads Increment and Decrement The Same Thread-Safe Counter Repeatedly, Never Resulting in a Race Condition

```
public static int counter = 0;
public static Object lock = new Object();

public static void start() throws InterruptedException {
       incrementCounter ic;
       decrementCounter dc;
       while(counter == 0) { // because of proper locking, this condition is never false
              counter = 0;
              ic = new incrementCounter();
              dc = new decrementCounter();
              ic.start();
              dc.start();
              ic.join();
              dc.join();
       }
       System.out.println(counter); // Never reached
}

public static class incrementCounter extends Thread {
    public void run() {
       synchronized (lock) {
              counter++;
       }
    }
}

public static class decrementCounter extends Thread {
    public void run() {
       synchronized (lock) {
              counter--;
       }
    }
}
```

Scanned Languages

Language	Hash Number	Change Date
CPP	4541647240435660	1/6/2025
Common	0105849645654507	1/6/2025