vul_files_49 Scan Report

Project Name	vul_files_49
Scan Start	Wednesday, January 8, 2025 10:47:58 AM
Preset	Checkmarx Default
Scan Time	01h:26m:02s
Lines Of Code Scanned	298010
Files Scanned	116
Report Creation Time	Wednesday, January 8, 2025 12:19:17 PM
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051
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

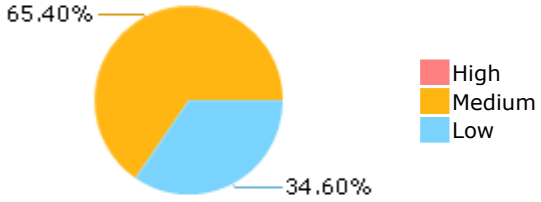
Results Limit

Results limit per query was set to 50

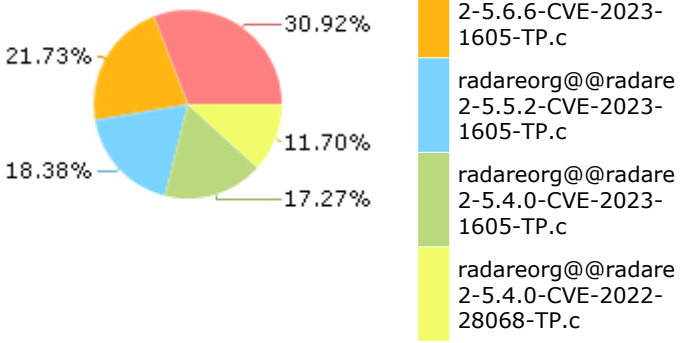
Selected Queries

Selected queries are listed in Result Summary

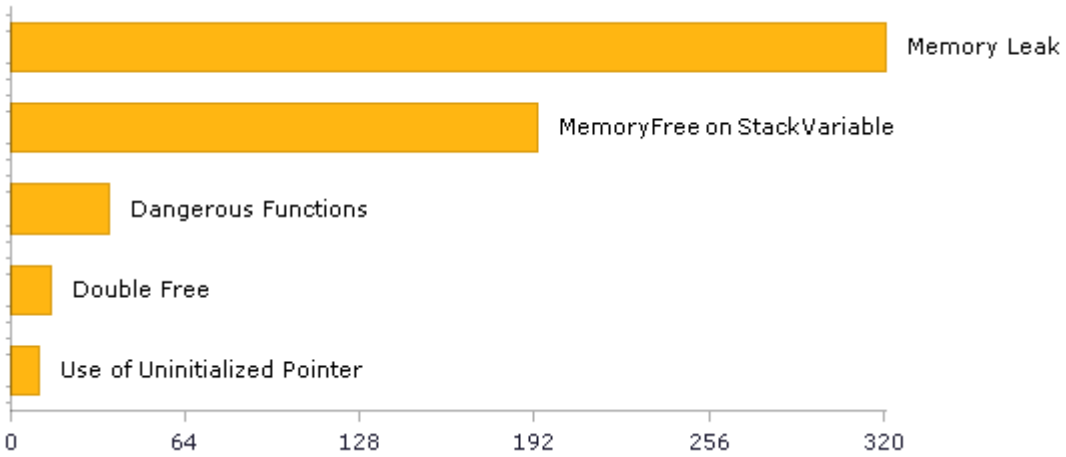
Result Summary



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	5	5
A2-Broken Authentication	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	3	3
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	36	36
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	36	36
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	2	2
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.	1	1
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)	3	3
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)*	338	332
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*	3	3
SI-11 Error Handling (P2)*	294	294
SI-15 Information Output Filtering (P0)	0	0
SI-16 Memory Protection (P1)	15	15

***** 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

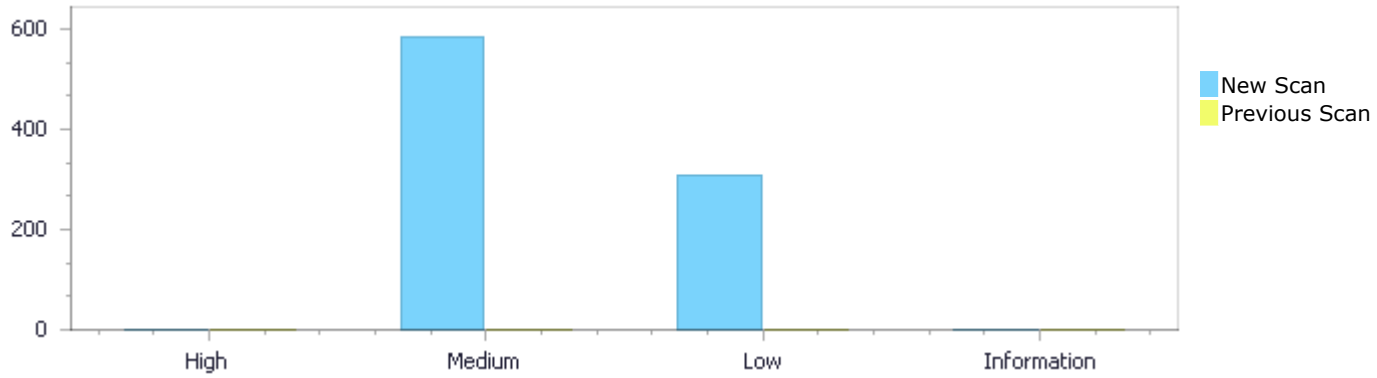
	modify the application's data and resources. This can provide the attacker a direct method of subverting the intended use of the software for personal or monetary gain.		
M9-Reverse Engineering	This category includes analysis of the final core binary to determine its source code, libraries, algorithms, 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	586	310	0	896
Recurrent Issues	0	0	0	0	0
Total	0	586	310	0	896
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	586	310	0	896
Urgent	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0
Total	0	586	310	0	896

Result Summary

Vulnerability Type	Occurrences	Severity
Memory Leak	321	Medium
MemoryFree on StackVariable	193	Medium
Dangerous Functions	36	Medium
Double Free	15	Medium
Use of Uninitialized Pointer	10	Medium

Wrong Size t Allocation	5	Medium
Use of Zero Initialized Pointer	3	Medium
Buffer Overflow AddressOfLocalVarReturned	1	Medium
Buffer Overflow boundcpy WrongSizeParam	1	Medium
Wrong Memory Allocation	1	Medium
Unchecked Return Value	294	Low
Sizeof Pointer Argument	6	Low
NULL Pointer Dereference	3	Low
Incorrect Permission Assignment For Critical Resources	2	Low
Unchecked Array Index	2	Low
Improper Resource Access Authorization	1	Low
TOCTOU	1	Low
Use of Sizeof On a Pointer Type	1	Low

10 Most Vulnerable Files

High and Medium Vulnerabilities

File Name	Issues Found
radareorg@@radare2-5.6.6-CVE-2022-0695-FP.c	59
radareorg@@radare2-5.6.6-CVE-2023-1605-TP.c	42
radareorg@@radare2-5.5.2-CVE-2023-1605-TP.c	36
radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	34
radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c	34
radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c	24
radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c	24
radareorg@@radare2-5.6.6-CVE-2022-0523-FP.c	24
radareorg@@radare2-5.6.6-CVE-2022-0520-FP.c	22
radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c	16

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=1020061&projectid=20051&pathid=558
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c
Line	283	283
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c
Method static bool __ne_get_resources(r_bin_ne_obj_t *bin) {

```
....
283.                    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=1020061&projectid=20051&pathid=559
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1238-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1238-TP.c
Line	283	283
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1238-TP.c

Method	static bool __ne_get_resources(r_bin_ne_obj_t *bin) {

```
....
283.                    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=1020061&projectid=20051&pathid=560
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1283-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1283-TP.c
Line	283	283
Object	name	name

Code Snippet	
File Name	radareorg@@radare2-5.5.2-CVE-2022-1283-TP.c
Method	static bool __ne_get_resources(r_bin_ne_obj_t *bin) {

```
....
283.                    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=1020061&projectid=20051&pathid=561
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1296-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1296-TP.c
Line	283	283
Object	name	name

Code Snippet	
File Name	radareorg@@radare2-5.5.2-CVE-2022-1296-TP.c
Method	static bool __ne_get_resources(r_bin_ne_obj_t *bin) {

```
....
283.                    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=1020061&projectid=20051&pathid=562
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1297-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1297-TP.c
Line	283	283
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1297-TP.c
Method static bool __ne_get_resources(r_bin_ne_obj_t *bin) {

```
....
283.                    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=1020061&projectid=20051&pathid=563
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1382-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1382-TP.c
Line	283	283
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1382-TP.c
Method static bool __ne_get_resources(r_bin_ne_obj_t *bin) {

```
....
283.                    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=1020061&projectid=20051&pathid=564
Status	New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-1237-TP.c	radareorg@@radare2-5.6.6-CVE-2022-1237-TP.c
Line	290	290
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-1237-TP.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 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=565
Status	New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-1238-TP.c	radareorg@@radare2-5.6.6-CVE-2022-1238-TP.c
Line	290	290
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-1238-TP.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 9:

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

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=566
Status	New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-1283-TP.c	radareorg@@radare2-5.6.6-CVE-2022-1283-TP.c
Line	290	290
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-1283-TP.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 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=567
Status	New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-1296-TP.c	radareorg@@radare2-5.6.6-CVE-2022-1296-TP.c
Line	290	290
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-1296-TP.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 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=568
Status	New

	Source	Destination

File	radareorg@@radare2-5.6.6-CVE-2022-1297-TP.c	radareorg@@radare2-5.6.6-CVE-2022-1297-TP.c
Line	290	290
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-1297-TP.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 12:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=569
Status New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-1382-TP.c	radareorg@@radare2-5.6.6-CVE-2022-1382-TP.c
Line	290	290
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-1382-TP.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 13:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=570
Status New

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c
Line	481	481

Object	buf	buf

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c
Method ut8 *buf = malloc (bsize);

```
....
481.        ut8 *buf = malloc (bsize);
```

Memory Leak\Path 14:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=571
Status New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Line	284	284
Object	s	s

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Method static pyc_object *get_float_object(RBuffer *buffer) {

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

Memory Leak\Path 15:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=572
Status New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Line	345	345
Object	s1	s1

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Method static pyc_object *get_complex_object(RBuffer *buffer) {

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

Memory Leak\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=573
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Line	366	366
Object	s2	s2

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Method static pyc_object *get_complex_object(RBuffer *buffer) {

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

Memory Leak\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=574
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Line	284	284
Object	s	s

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Method static pyc_object *get_float_object(RBuffer *buffer) {

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

Memory Leak\Path 18:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=575
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Line	345	345
Object	s1	s1

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Method static pyc_object *get_complex_object(RBuffer *buffer) {

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

Memory Leak\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=576
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Line	366	366
Object	s2	s2

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Method static pyc_object *get_complex_object(RBuffer *buffer) {

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

Memory Leak\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=577
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0712-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0712-TP.c
Line	192	192
Object	b	b

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0712-TP.c
Method ut8 *b = malloc (size);

```
....
192.        ut8 *b = malloc (size);
```

Memory Leak\Path 21:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=578
Status New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0713-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0713-TP.c
Line	151	151
Object	result	result

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0713-TP.c
Method static char *str_dup_safe_fixed(const ut8 *b, const ut8 *str, ut64 len, const ut8 *end) {

```
....
151.            char *result = calloc (1, len + 1);
```

Memory Leak\Path 22:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=579
Status New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1061-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1061-TP.c
Line	192	192

Object	b	b

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1061-TP.c
Method ut8 *b = malloc (size);

```
....
192.        ut8 *b = malloc (size);
```

Memory Leak\Path 23:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=580
Status New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c
Line	42	42
Object	str	str

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c
Method static char *__read_nonnull_str_at(RBuffer *buf, ut64 offset) {

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

Memory Leak\Path 24:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=581
Status New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c
Line	125	125
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c
Method RList *r_bin_ne_get_symbols(r_bin_ne_obj_t *bin) {

```
....
125.              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=1020061&projectid=20051&pathid=582
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c
Line	331	331
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c
Method RList *r_bin_ne_get_imports(r_bin_ne_obj_t *bin) {

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

Memory Leak\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=583
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1238-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1238-TP.c
Line	42	42
Object	str	str

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1238-TP.c
Method static char *__read_nonnull_str_at(RBuffer *buf, ut64 offset) {

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

Memory Leak\Path 27:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=584
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1238-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1238-TP.c
Line	125	125
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1238-TP.c
Method RList *r_bin_ne_get_symbols(r_bin_ne_obj_t *bin) {

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

Memory Leak\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=585
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1238-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1238-TP.c
Line	331	331
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1238-TP.c
Method RList *r_bin_ne_get_imports(r_bin_ne_obj_t *bin) {

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

Memory Leak\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=586
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1283-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1283-TP.c
Line	42	42
Object	str	str

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1283-TP.c
Method static char *__read_nonnull_str_at(RBuffer *buf, ut64 offset) {

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

Memory Leak\Path 30:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=587
Status New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1283-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1283-TP.c
Line	125	125
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1283-TP.c
Method RList *r_bin_ne_get_symbols(r_bin_ne_obj_t *bin) {

```
....
125.              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=1020061&projectid=20051&pathid=588
Status New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1283-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1283-TP.c
Line	331	331

Object	name	name

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1283-TP.c
Method RList *r_bin_ne_get_imports(r_bin_ne_obj_t *bin) {

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

Memory Leak\Path 32:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=589
Status New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1296-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1296-TP.c
Line	42	42
Object	str	str

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1296-TP.c
Method static char *__read_nonnull_str_at(RBuffer *buf, ut64 offset) {

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

Memory Leak\Path 33:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=590
Status New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1296-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1296-TP.c
Line	125	125
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1296-TP.c
Method RList *r_bin_ne_get_symbols(r_bin_ne_obj_t *bin) {

```
....
125.              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=1020061&projectid=20051&pathid=591
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1296-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1296-TP.c
Line	331	331
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1296-TP.c
Method RList *r_bin_ne_get_imports(r_bin_ne_obj_t *bin) {

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

Memory Leak\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=592
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1297-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1297-TP.c
Line	42	42
Object	str	str

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1297-TP.c
Method static char *__read_nonnull_str_at(RBuffer *buf, ut64 offset) {

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

Memory Leak\Path 36:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=593
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1297-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1297-TP.c
Line	125	125
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1297-TP.c
Method RList *r_bin_ne_get_symbols(r_bin_ne_obj_t *bin) {

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

Memory Leak\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=594
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1297-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1297-TP.c
Line	331	331
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1297-TP.c
Method RList *r_bin_ne_get_imports(r_bin_ne_obj_t *bin) {

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

Memory Leak\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=595
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1382-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1382-TP.c
Line	42	42
Object	str	str

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1382-TP.c
Method static char *__read_nonnull_str_at(RBuffer *buf, ut64 offset) {

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

Memory Leak\Path 39:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=596
Status New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1382-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1382-TP.c
Line	125	125
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1382-TP.c
Method RList *r_bin_ne_get_symbols(r_bin_ne_obj_t *bin) {

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

Memory Leak\Path 40:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=597
Status New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1382-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1382-TP.c
Line	331	331

Object	name	name

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1382-TP.c
Method RList *r_bin_ne_get_imports(r_bin_ne_obj_t *bin) {

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

Memory Leak\Path 41:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=598
Status New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1383-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1383-TP.c
Line	151	151
Object	result	result

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1383-TP.c
Method static char *str_dup_safe_fixed(const ut8 *b, const ut8 *str, ut64 len, const ut8 *end) {

```
....
151.              char *result = calloc (1, len + 1);
```

Memory Leak\Path 42:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=599
Status New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1437-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1437-TP.c
Line	192	192
Object	b	b

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1437-TP.c
Method ut8 *b = malloc (size);

```
....
192.        ut8 *b = malloc (size);
```

Memory Leak\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=600
Status	New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-0523-FP.c	radareorg@@radare2-5.6.6-CVE-2022-0523-FP.c
Line	280	280
Object	s	s

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-0523-FP.c
Method static pyc_object *get_float_object(RBuffer *buffer) {

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

Memory Leak\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=601
Status	New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-0523-FP.c	radareorg@@radare2-5.6.6-CVE-2022-0523-FP.c
Line	341	341
Object	s1	s1

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-0523-FP.c
Method static pyc_object *get_complex_object(RBuffer *buffer) {

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

Memory Leak\Path 45:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=602
Status	New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-0523-FP.c	radareorg@@radare2-5.6.6-CVE-2022-0523-FP.c
Line	362	362
Object	s2	s2

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-0523-FP.c
Method static pyc_object *get_complex_object(RBuffer *buffer) {

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

Memory Leak\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=603
Status	New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-1061-TP.c	radareorg@@radare2-5.6.6-CVE-2022-1061-TP.c
Line	192	192
Object	b	b

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-1061-TP.c
Method ut8 *b = malloc (size);

```
....
192.        ut8 *b = malloc (size);
```

Memory Leak\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=604
Status	New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-1237-TP.c	radareorg@@radare2-5.6.6-CVE-2022-1237-TP.c
Line	42	42
Object	str	str

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-1237-TP.c
Method static char *__read_nonnull_str_at(RBuffer *buf, ut64 offset) {

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

Memory Leak\Path 48:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=605
Status New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-1237-TP.c	radareorg@@radare2-5.6.6-CVE-2022-1237-TP.c
Line	132	132
Object	name	name

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-1237-TP.c
Method RList *r_bin_ne_get_symbols(r_bin_ne_obj_t *bin) {

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

Memory Leak\Path 49:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=606
Status New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-1237-TP.c	radareorg@@radare2-5.6.6-CVE-2022-1237-TP.c
Line	338	338

Object	name	name

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-1237-TP.c
Method RList *r_bin_ne_get_imports(r_bin_ne_obj_t *bin) {

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

Memory Leak\Path 50:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=607
Status New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-1238-TP.c	radareorg@@radare2-5.6.6-CVE-2022-1238-TP.c
Line	42	42
Object	str	str

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-1238-TP.c
Method static char *__read_nonnull_str_at(RBuffer *buf, ut64 offset) {

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

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=1020061&projectid=20051&pathid=298
Status New

Calling free() (line 1269) on a variable that was not dynamically allocated (line 1269) in file radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	1344	1344

Object	val	val

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method static void sdb_save_dwarf_function(Function *dwarf_fcn, RList/*<Variable*>*/ *variables, Sdb *sdb) {

```
....
1344.             free (val);
```

MemoryFree on StackVariable\Path 2:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=299
Status New

Calling free() (line 1269) on a variable that was not dynamically allocated (line 1269) in file radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	1353	1353
Object	vars_val	vars_val

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method static void sdb_save_dwarf_function(Function *dwarf_fcn, RList/*<Variable*>*/ *variables, Sdb *sdb) {

```
....
1353.       free (vars_val);
```

MemoryFree on StackVariable\Path 3:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=300
Status New

Calling free() (line 1612) on a variable that was not dynamically allocated (line 1612) in file radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c

Line	1617	1617
Object	addr_key	addr_key

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method ls_foreach (sdb_list, it, kv) {

```
....
1617.             free (addr_key);
```

MemoryFree on StackVariable\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=301
Status	New

Calling free() (line 1612) on a variable that was not dynamically allocated (line 1612) in file radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	1625	1625
Object	real_name_key	real_name_key

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method ls_foreach (sdb_list, it, kv) {

```
....
1625.                   free (real_name_key);
```

MemoryFree on StackVariable\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=302
Status	New

Calling free() (line 1612) on a variable that was not dynamically allocated (line 1612) in file radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c

Line	1628	1628
Object	real_name	real_name

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method ls_foreach (sdb_list, it, kv) {

```
....
1628.                   free (real_name);
```

MemoryFree on StackVariable\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=303
Status	New

Calling free() (line 1612) on a variable that was not dynamically allocated (line 1612) in file radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	1631	1631
Object	dwf_name	dwf_name

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method ls_foreach (sdb_list, it, kv) {

```
....
1631.                   free (dwf_name);
```

MemoryFree on StackVariable\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=304
Status	New

Calling free() (line 1612) on a variable that was not dynamically allocated (line 1612) in file radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c

Line	1635	1635
Object	tmp	tmp

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method ls_foreach (sdb_list, it, kv) {

```
....
1635.                    free (tmp);
```

MemoryFree on StackVariable\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=305
Status	New

Calling free() (line 1612) on a variable that was not dynamically allocated (line 1612) in file radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	1638	1638
Object	fcnstr	fcnstr

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method ls_foreach (sdb_list, it, kv) {

```
....
1638.                    free (fcnstr);
```

MemoryFree on StackVariable\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=306
Status	New

Calling free() (line 1612) on a variable that was not dynamically allocated (line 1612) in file radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c

Line	1661	1661
Object	global_name	global_name

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method ls_foreach (sdb_list, it, kv) {

```
....
1661.                           free (global_name);
```

MemoryFree on StackVariable\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=307
Status	New

Calling free() (line 1612) on a variable that was not dynamically allocated (line 1612) in file radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	1673	1673
Object	var_key	var_key

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method ls_foreach (sdb_list, it, kv) {

```
....
1673.                   free (var_key);
```

MemoryFree on StackVariable\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=308
Status	New

Calling free() (line 1612) on a variable that was not dynamically allocated (line 1612) in file radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c

Line	1674	1674
Object	var_data	var_data

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method ls_foreach (sdb_list, it, kv) {

```
....
1674.                   free (var_data);
```

MemoryFree on StackVariable\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=309
Status	New

Calling free() (line 1612) on a variable that was not dynamically allocated (line 1612) in file radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	1678	1678
Object	var_names_key	var_names_key

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method ls_foreach (sdb_list, it, kv) {

```
....
1678.               free (var_names_key);
```

MemoryFree on StackVariable\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=310
Status	New

Calling free() (line 1612) on a variable that was not dynamically allocated (line 1612) in file radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c

Line	1679	1679
Object	vars	vars

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method ls_foreach (sdb_list, it, kv) {

```
....
1679.             free (vars);
```

MemoryFree on StackVariable\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=311
Status	New

Calling free() (line 128) on a variable that was not dynamically allocated (line 128) in file radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c	radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c
Line	131	131
Object	ptr	ptr

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c
Method static RBinImport *_fill_bin_import(struct r_bin_coff_obj *bin, int idx) {

```
....
131.             free (ptr);
```

MemoryFree on StackVariable\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=312
Status	New

Calling free() (line 128) on a variable that was not dynamically allocated (line 128) in file radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c	radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c

Line	136	136
Object	ptr	ptr

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c
Method static RBinImport *_fill_bin_import(struct r_bin_coff_obj *bin, int idx) {

```
....
136.              free (ptr);
```

MemoryFree on StackVariable\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=313
Status	New

Calling free() (line 128) on a variable that was not dynamically allocated (line 128) in file radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c	radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c
Line	141	141
Object	ptr	ptr

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c
Method static RBinImport *_fill_bin_import(struct r_bin_coff_obj *bin, int idx) {

```
....
141.              free (ptr);
```

MemoryFree on StackVariable\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=314
Status	New

Calling free() (line 165) on a variable that was not dynamically allocated (line 165) in file radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c	radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c

Line	186	186
Object	tmp	tmp

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c
Method static RList *sections(RBinFile *bf) {

```
....
186.                            free (tmp);
```

MemoryFree on StackVariable\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=315
Status	New

Calling free() (line 165) on a variable that was not dynamically allocated (line 165) in file radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c	radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c
Line	190	190
Object	tmp	tmp

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c
Method static RList *sections(RBinFile *bf) {

```
....
190.                        free (tmp);
```

MemoryFree on StackVariable\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=316
Status	New

Calling free() (line 217) on a variable that was not dynamically allocated (line 217) in file radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c	radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c

Line	235	235
Object	ptr	ptr

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c
Method static RList *symbols(RBinFile *bf) {

```
....
235.                            free (ptr);
```

MemoryFree on StackVariable\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=317
Status	New

Calling free() (line 17) on a variable that was not dynamically allocated (line 17) in file radareorg@@radare2-5.4.0-CVE-2023-27590-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2023-27590-TP.c	radareorg@@radare2-5.4.0-CVE-2023-27590-TP.c
Line	62	62
Object	ostr	ostr

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2023-27590-TP.c
Method static RList *__io_maps(RDebug *dbg) {

```
....
62.    free (ostr);
```

MemoryFree on StackVariable\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=318
Status	New

Calling free() (line 113) on a variable that was not dynamically allocated (line 113) in file radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c

Line	155	155
Object	arg	arg

Code Snippet
File Name radareorg@@radare2-5.4.0-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) {

```
....
155.                    free (arg);
```

MemoryFree on StackVariable\Path 22:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=319
Status New

Calling free() (line 113) on a variable that was not dynamically allocated (line 113) in file radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	166	166
Object	arg	arg

Code Snippet
File Name radareorg@@radare2-5.4.0-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) {

```
....
166.                    free (arg);
```

MemoryFree on StackVariable\Path 23:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=320
Status New

Calling free() (line 113) on a variable that was not dynamically allocated (line 113) in file radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2023-	radareorg@@radare2-5.4.0-CVE-2023-

	5686-TP.c	5686-TP.c
Line	214	214
Object	arg	arg

Code Snippet
File Name radareorg@@radare2-5.4.0-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) {

```
....
214.                    free (arg);
```

MemoryFree on StackVariable\Path 24:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=321
Status New

Calling free() (line 113) on a variable that was not dynamically allocated (line 113) in file radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	227	227
Object	arg	arg

Code Snippet
File Name radareorg@@radare2-5.4.0-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) {

```
....
227.                    free (arg);
```

MemoryFree on StackVariable\Path 25:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=322
Status New

Calling free() (line 113) on a variable that was not dynamically allocated (line 113) in file radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c may result with a crash.

	Source	Destination

File	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	240	240
Object	arg	arg

Code Snippet
File Name radareorg@@radare2-5.4.0-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) {

```
....
240.                    free (arg);
```

MemoryFree on StackVariable\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=323
Status	New

Calling free() (line 78) on a variable that was not dynamically allocated (line 78) in file radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Line	84	84
Object	ret	ret

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Method static ut8 *get_bytes(RBuffer *buffer, ut32 size) {

```
....
84.         free (ret);
```

MemoryFree on StackVariable\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=324
Status	New

Calling free() (line 271) on a variable that was not dynamically allocated (line 271) in file radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c may result with a crash.

	Source	Destination

File	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Line	286	286
Object	ret	ret

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Method static pyc_object *get_float_object(RBuffer *buffer) {

```
....
286.                free (ret);
```

MemoryFree on StackVariable\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=325
Status	New

Calling free() (line 324) on a variable that was not dynamically allocated (line 324) in file radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Line	342	342
Object	ret	ret

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Method static pyc_object *get_complex_object(RBuffer *buffer) {

```
....
342.                free (ret);
```

MemoryFree on StackVariable\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=326
Status	New

Calling free() (line 324) on a variable that was not dynamically allocated (line 324) in file radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c may result with a crash.

	Source	Destination

File	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Line	347	347
Object	ret	ret

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Method static pyc_object *get_complex_object(RBuffer *buffer) {

```
....
347.                free (ret);
```

MemoryFree on StackVariable\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=327
Status	New

Calling free() (line 491) on a variable that was not dynamically allocated (line 491) in file radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Line	502	502
Object	ret	ret

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Method static pyc_object *get_array_object_generic(RBuffer *buffer, ut32 size) {

```
....
502.                free (ret);
```

MemoryFree on StackVariable\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=328
Status	New

Calling free() (line 491) on a variable that was not dynamically allocated (line 491) in file radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c may result with a crash.

	Source	Destination

File	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Line	515	515
Object	ret	ret

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Method static pyc_object *get_array_object_generic(RBuffer *buffer, ut32 size) {

```
....
515.                    free (ret);
```

MemoryFree on StackVariable\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=329
Status	New

Calling free() (line 846) on a variable that was not dynamically allocated (line 846) in file radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Line	852	852
Object	ret	ret

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Method static pyc_object *get_code_object(RBuffer *buffer) {

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

MemoryFree on StackVariable\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=330
Status	New

Calling free() (line 846) on a variable that was not dynamically allocated (line 846) in file radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c may result with a crash.

	Source	Destination

File	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Line	853	853
Object	cobj	cobj

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Method static pyc_object *get_code_object(RBuffer *buffer) {

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

MemoryFree on StackVariable\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=331
Status	New

Calling free() (line 846) on a variable that was not dynamically allocated (line 846) in file radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Line	869	869
Object	ret	ret

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Method static pyc_object *get_code_object(RBuffer *buffer) {

```
....
869.                free (ret);
```

MemoryFree on StackVariable\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=332
Status	New

Calling free() (line 846) on a variable that was not dynamically allocated (line 846) in file radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c may result with a crash.

	Source	Destination

File	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Line	870	870
Object	cobj	cobj

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Method static pyc_object *get_code_object(RBuffer *buffer) {

```
....
870.              free (cobj);
```

MemoryFree on StackVariable\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=333
Status	New

Calling free() (line 846) on a variable that was not dynamically allocated (line 846) in file radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Line	970	970
Object	cobj	cobj

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Method static pyc_object *get_code_object(RBuffer *buffer) {

```
....
970.              free (cobj);
```

MemoryFree on StackVariable\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=334
Status	New

Calling free() (line 78) on a variable that was not dynamically allocated (line 78) in file radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c may result with a crash.

	Source	Destination

File	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Line	84	84
Object	ret	ret

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Method static ut8 *get_bytes(RBuffer *buffer, ut32 size) {

```
....
84.          free (ret);
```

MemoryFree on StackVariable\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=335
Status	New

Calling free() (line 271) on a variable that was not dynamically allocated (line 271) in file radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Line	286	286
Object	ret	ret

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Method static pyc_object *get_float_object(RBuffer *buffer) {

```
....
286.              free (ret);
```

MemoryFree on StackVariable\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=336
Status	New

Calling free() (line 324) on a variable that was not dynamically allocated (line 324) in file radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c may result with a crash.

	Source	Destination

File	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Line	342	342
Object	ret	ret

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Method static pyc_object *get_complex_object(RBuffer *buffer) {

```
....
342.                free (ret);
```

MemoryFree on StackVariable\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=337
Status	New

Calling free() (line 324) on a variable that was not dynamically allocated (line 324) in file radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Line	347	347
Object	ret	ret

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Method static pyc_object *get_complex_object(RBuffer *buffer) {

```
....
347.                free (ret);
```

MemoryFree on StackVariable\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=338
Status	New

Calling free() (line 491) on a variable that was not dynamically allocated (line 491) in file radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c may result with a crash.

	Source	Destination

File	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Line	502	502
Object	ret	ret

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Method static pyc_object *get_array_object_generic(RBuffer *buffer, ut32 size) {

```
....
502.                free (ret);
```

MemoryFree on StackVariable\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=339
Status	New

Calling free() (line 491) on a variable that was not dynamically allocated (line 491) in file radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Line	515	515
Object	ret	ret

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Method static pyc_object *get_array_object_generic(RBuffer *buffer, ut32 size) {

```
....
515.                    free (ret);
```

MemoryFree on StackVariable\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=340
Status	New

Calling free() (line 846) on a variable that was not dynamically allocated (line 846) in file radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c may result with a crash.

	Source	Destination

File	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Line	852	852
Object	ret	ret

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Method static pyc_object *get_code_object(RBuffer *buffer) {

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

MemoryFree on StackVariable\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=341
Status	New

Calling free() (line 846) on a variable that was not dynamically allocated (line 846) in file radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Line	853	853
Object	cobj	cobj

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Method static pyc_object *get_code_object(RBuffer *buffer) {

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

MemoryFree on StackVariable\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=342
Status	New

Calling free() (line 846) on a variable that was not dynamically allocated (line 846) in file radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c may result with a crash.

	Source	Destination

File	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Line	869	869
Object	ret	ret

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Method static pyc_object *get_code_object(RBuffer *buffer) {

```
....
869.                free (ret);
```

MemoryFree on StackVariable\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=343
Status	New

Calling free() (line 846) on a variable that was not dynamically allocated (line 846) in file radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Line	870	870
Object	cobj	cobj

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Method static pyc_object *get_code_object(RBuffer *buffer) {

```
....
870.                free (cobj);
```

MemoryFree on StackVariable\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=344
Status	New

Calling free() (line 846) on a variable that was not dynamically allocated (line 846) in file radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c may result with a crash.

	Source	Destination

File	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Line	970	970
Object	cobj	cobj

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Method static pyc_object *get_code_object(RBuffer *buffer) {

```
....
970.              free (cobj);
```

MemoryFree on StackVariable\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=345
Status	New

Calling free() (line 16) on a variable that was not dynamically allocated (line 16) in file radareorg@@radare2-5.5.2-CVE-2022-0713-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0713-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0713-TP.c
Line	21	21
Object	hdr	hdr

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0713-TP.c
Method static RCoreSymCacheElementHdr *r_coresym_cache_element_header_new(RBuffer *buf, size_t off, int bits) {

```
....
21.   free (hdr);
```

MemoryFree on StackVariable\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=346
Status	New

Calling free() (line 51) on a variable that was not dynamically allocated (line 51) in file radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c may result with a crash.

	Source	Destination

File	radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c
Line	61	61
Object	ord	ord

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c
Method static char *__func_name_from_ord(char *module, ut16 ordinal) {

```
....
61.                free (ord);
```

MemoryFree on StackVariable\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=347
Status	New

Calling free() (line 51) on a variable that was not dynamically allocated (line 51) in file radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c may result with a crash.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c
Line	64	64
Object	sdb	sdb

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c
Method static char *__func_name_from_ord(char *module, ut16 ordinal) {

```
....
64.          free (sdb);
```

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=1020061&projectid=20051&pathid=507
Status	New

The dangerous function, memcpy, was found in use at line 92 in radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	110	110
Object	memcpy	memcpy

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method static bool strbuf_rev_prepend_char(RStrBuf *sb, const char *s, int c) {

```
....
110.              memcpy (ns, sb_str, idx);
```

Dangerous Functions\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=508
Status	New

The dangerous function, memcpy, was found in use at line 92 in radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	111	111
Object	memcpy	memcpy

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method static bool strbuf_rev_prepend_char(RStrBuf *sb, const char *s, int c) {

```
....
111.              memcpy (ns + idx, s, l);
```

Dangerous Functions\Path 3:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=509
Status	New

The dangerous function, memcpy, was found in use at line 92 in radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	112	112
Object	memcpy	memcpy

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method static bool strbuf_rev_prepend_char(RStrBuf *sb, const char *s, int c) {

```
....
112.              memcpy (ns + idx + l, sb_str + idx, sb->len - idx);
```

Dangerous Functions\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=510
Status	New

The dangerous function, memcpy, was found in use at line 127 in radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	145	145
Object	memcpy	memcpy

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method static bool strbuf_rev_append_char(RStrBuf *sb, const char *s, const char *needle) {

```
....
145.              memcpy (ns, sb_str, idx);
```

Dangerous Functions\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=511
Status	New

The dangerous function, memcpy, was found in use at line 127 in radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	146	146
Object	memcpy	memcpy

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method static bool strbuf_rev_append_char(RStrBuf *sb, const char *s, const char *needle) {

```
....
146.              memcpy (ns + idx, s, l);
```

Dangerous Functions\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=512
Status	New

The dangerous function, memcpy, was found in use at line 127 in radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	147	147
Object	memcpy	memcpy

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method static bool strbuf_rev_append_char(RStrBuf *sb, const char *s, const char *needle) {

```
....
147.              memcpy (ns + idx + l, sb_str + idx, sb->len - idx);
```

Dangerous Functions\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=513
Status	New

The dangerous function, memcpy, was found in use at line 93 in radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-27590-TP.c	radareorg@@radare2-5.4.0-CVE-2023-27590-TP.c
Line	112	112
Object	memcpy	memcpy

Code Snippet
File Name radareorg@@radare2-5.4.0-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 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=514
Status	New

The dangerous function, memcpy, was found in use at line 767 in radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Line	803	803
Object	memcpy	memcpy

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c

Method	static pyc_object *copy_object(pyc_object *object) {

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

Dangerous Functions\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=515
Status	New

The dangerous function, memcpy, was found in use at line 767 in radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Line	803	803
Object	memcpy	memcpy

Code Snippet	
File Name	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Method	static pyc_object *copy_object(pyc_object *object) {

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

Dangerous Functions\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=516
Status	New

The dangerous function, memcpy, was found in use at line 93 in radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-27590-TP.c	radareorg@@radare2-5.5.2-CVE-2023-27590-TP.c
Line	112	112
Object	memcpy	memcpy

Code Snippet

File Name	radareorg@@radare2-5.5.2-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 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=517
Status	New

The dangerous function, memcpy, was found in use at line 750 in radareorg@@radare2-5.6.6-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@@radare2-5.6.6-CVE-2022-0523-FP.c	radareorg@@radare2-5.6.6-CVE-2022-0523-FP.c
Line	786	786
Object	memcpy	memcpy

Code Snippet

File Name	radareorg@@radare2-5.6.6-CVE-2022-0523-FP.c
Method	static pyc_object *copy_object(pyc_object *object) {

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

Dangerous Functions\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=518
Status	New

The dangerous function, memcpy, was found in use at line 333 in radareorg@@radare2-5.6.6-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@@radare2-5.6.6-CVE-2022-0695-FP.c	radareorg@@radare2-5.6.6-CVE-2022-0695-FP.c
Line	348	348
Object	memcpy	memcpy

Code Snippet	
File Name	radareorg@@radare2-5.6.6-CVE-2022-0695-FP.c
Method	struct r_bin_te_section_t* r_bin_te_get_sections(struct r_bin_te_obj_t* bin) {

```
....
348.              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=1020061&projectid=20051&pathid=519
Status	New

The dangerous function, memcpy, was found in use at line 93 in radareorg@@radare2-5.6.6-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@@radare2-5.6.6-CVE-2023-27590-TP.c	radareorg@@radare2-5.6.6-CVE-2023-27590-TP.c
Line	112	112
Object	memcpy	memcpy

Code Snippet	
File Name	radareorg@@radare2-5.6.6-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 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=520
Status	New

The dangerous function, strlen, was found in use at line 92 in radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	94	94

Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method static bool strbuf_rev_prepend_char(RStrBuf *sb, const char *s, int c) {

```
....
94.   size_t l = strlen (s);
```

Dangerous Functions\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=521
Status	New

The dangerous function, strlen, was found in use at line 127 in radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	129	129
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method static bool strbuf_rev_append_char(RStrBuf *sb, const char *s, const char *needle) {

```
....
129.        size_t l = strlen (s);
```

Dangerous Functions\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=522
Status	New

The dangerous function, strlen, was found in use at line 127 in radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-	radareorg@@radare2-5.4.0-CVE-2022-

	28068-TP.c	28068-TP.c
Line	140	140
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method static bool strbuf_rev_append_char(RStrBuf *sb, const char *s, const char *needle) {

```
....
140.        pivot += strlen (needle);
```

Dangerous Functions\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=523
Status	New

The dangerous function, strlen, was found in use at line 53 in radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c
Line	66	66
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c
Method static void __var_rename(RAnal *anal, RAnalVar *v, const char *name, ut64 addr) {

```
....
66.   if (!is_default && (strlen (v->name) > strlen (name))) {
```

Dangerous Functions\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=524
Status	New

The dangerous function, strlen, was found in use at line 53 in radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c
Line	66	66
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c
Method static void __var_rename(RAnal *anal, RAnalVar *v, const char *name, ut64 addr) {

```
....
66.    if (!is_default && (strlen (v->name) > strlen (name))) {
```

Dangerous Functions\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=525
Status	New

The dangerous function, strlen, was found in use at line 202 in radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c
Line	214	214
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c
Method static RList *parse_format(RCore *core, char *fmt) {

```
....
214.         fmt[strlen (fmt) - 1] = '\0';
```

Dangerous Functions\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=526
Status	New

The dangerous function, strlen, was found in use at line 17 in radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-27590-TP.c	radareorg@@radare2-5.4.0-CVE-2023-27590-TP.c
Line	45	45
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2023-27590-TP.c
Method static RList *__io_maps(RDebug *dbg) {

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

Dangerous Functions\Path 21:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=527
Status New

The dangerous function, strlen, was found in use at line 17 in radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-27590-TP.c	radareorg@@radare2-5.4.0-CVE-2023-27590-TP.c
Line	49	49
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2023-27590-TP.c
Method static RList *__io_maps(RDebug *dbg) {

```
....
49.                     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=1020061&projectid=20051&pathid=528
Status New

The dangerous function, strlen, was found in use at line 93 in radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-27590-TP.c	radareorg@@radare2-5.4.0-CVE-2023-27590-TP.c
Line	104	104
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.4.0-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 23:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=529
Status New

The dangerous function, strlen, was found in use at line 17 in radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-27590-TP.c	radareorg@@radare2-5.5.2-CVE-2023-27590-TP.c
Line	45	45
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2023-27590-TP.c
Method static RList *__io_maps(RDebug *dbg) {

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

Dangerous Functions\Path 24:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=530

Status	New

The dangerous function, strlen, was found in use at line 17 in radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-27590-TP.c	radareorg@@radare2-5.5.2-CVE-2023-27590-TP.c
Line	49	49
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2023-27590-TP.c
Method static RList *__io_maps(RDebug *dbg) {

```
....
49.                     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=1020061&projectid=20051&pathid=531
Status	New

The dangerous function, strlen, was found in use at line 93 in radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-27590-TP.c	radareorg@@radare2-5.5.2-CVE-2023-27590-TP.c
Line	104	104
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.5.2-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 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20

	051&pathid=532
Status	New

The dangerous function, strlen, was found in use at line 59 in radareorg@@radare2-5.6.6-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@@radare2-5.6.6-CVE-2022-0520-FP.c	radareorg@@radare2-5.6.6-CVE-2022-0520-FP.c
Line	84	84
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-0520-FP.c
Method static int download(struct SPDBDownloader *pd) {

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

Dangerous Functions\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=533
Status	New

The dangerous function, strlen, was found in use at line 17 in radareorg@@radare2-5.6.6-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@@radare2-5.6.6-CVE-2023-27590-TP.c	radareorg@@radare2-5.6.6-CVE-2023-27590-TP.c
Line	45	45
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2023-27590-TP.c
Method static RList *__io_maps(RDebug *dbg) {

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

Dangerous Functions\Path 28:

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

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=534
Status	New

The dangerous function, strlen, was found in use at line 17 in radareorg@@radare2-5.6.6-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@@radare2-5.6.6-CVE-2023-27590-TP.c	radareorg@@radare2-5.6.6-CVE-2023-27590-TP.c
Line	49	49
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2023-27590-TP.c
Method static RList *__io_maps(RDebug *dbg) {

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

Dangerous Functions\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=535
Status	New

The dangerous function, strlen, was found in use at line 93 in radareorg@@radare2-5.6.6-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@@radare2-5.6.6-CVE-2023-27590-TP.c	radareorg@@radare2-5.6.6-CVE-2023-27590-TP.c
Line	104	104
Object	strlen	strlen

Code Snippet
File Name radareorg@@radare2-5.6.6-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 30:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=536
Status	New

The dangerous function, strncpy, was found in use at line 145 in radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c
Line	167	167
Object	strncpy	strncpy

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c
Method static void get_src_regname(RCore *core, ut64 addr, char *regname, int size) {

```
....
167.              strncpy (regname, op_esil, size - 1);
```

Dangerous Functions\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=537
Status	New

The dangerous function, strncpy, was found in use at line 519 in radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c
Line	746	746
Object	strncpy	strncpy

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c
Method r_list_foreach (fcn->bbs, it, bb) {

```
....
746.                                  strncpy (prev_type, var->type,
sizeof (prev_type) - 1);
```

Dangerous Functions\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=538
Status	New

The dangerous function, strncpy, was found in use at line 302 in radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	307	307
Object	strncpy	strncpy

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Method R_API int r_java_assemble(ut64 addr, ut8 *bytes, const char *string) {

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

Dangerous Functions\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=539
Status	New

The dangerous function, strncpy, was found in use at line 302 in radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	307	307
Object	strncpy	strncpy

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Method R_API int r_java_assemble(ut64 addr, ut8 *bytes, const char *string) {

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

Dangerous Functions\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=540
Status	New

The dangerous function, strncpy, was found in use at line 302 in radareorg@@radare2-5.6.6-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@@radare2-5.6.6-CVE-2023-5686-TP.c	radareorg@@radare2-5.6.6-CVE-2023-5686-TP.c
Line	307	307
Object	strncpy	strncpy

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2023-5686-TP.c
Method R_API int r_java_assemble(ut64 addr, ut8 *bytes, const char *string) {

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

Dangerous Functions\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=541
Status	New

The dangerous function, MultiByteToWideChar, was found in use at line 15 in radareorg@@radare2-5.5.2-CVE-2020-15121-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2020-15121-FP.c	radareorg@@radare2-5.5.2-CVE-2020-15121-FP.c
Line	22	22
Object	MultiByteToWideChar	MultiByteToWideChar

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2020-15121-FP.c
Method static wchar_t *r_utf8_to_utf16_l (const char *cstring, int len) {

```
....
22.   if ((wcsize = MultiByteToWideChar (CP_UTF8, 0, cstring, len, NULL,
0))) {
```

Dangerous Functions\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=542
Status	New

The dangerous function, MultiByteToWideChar, was found in use at line 15 in radareorg@@radare2-5.5.2-CVE-2020-15121-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2020-15121-FP.c	radareorg@@radare2-5.5.2-CVE-2020-15121-FP.c
Line	25	25
Object	MultiByteToWideChar	MultiByteToWideChar

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2020-15121-FP.c
Method static wchar_t *r_utf8_to_utf16_l (const char *cstring, int len) {

```
....
25.               MultiByteToWideChar (CP_UTF8, 0, cstring, len, rutf16,
wcsize);
```

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=1020061&projectid=20051&pathid=543
Status	New

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c	radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c

Line	320	451
Object	rel	rel

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c
Method static RList *_relocs_list(RBin *rbin, struct r_bin_coff_obj *bin, bool patch, ut64 imp_map) {

```
....
320.                    free (rel);
....
451.              free (rel);
```

Double Free\Path 2:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=544
Status New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c
Line	476	542
Object	reloc	reloc

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c
Method RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
476.                                free (reloc);
....
542.                           free (reloc);
```

Double Free\Path 3:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=545
Status New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1238-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1238-TP.c
Line	476	542

Object	reloc	reloc

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1238-TP.c
Method RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
476.                                    free (reloc);
....
542.                               free (reloc);
```

Double Free\Path 4:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=546
Status New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1283-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1283-TP.c
Line	476	542
Object	reloc	reloc

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1283-TP.c
Method RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
476.                                    free (reloc);
....
542.                               free (reloc);
```

Double Free\Path 5:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=547
Status New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1296-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1296-TP.c
Line	476	542
Object	reloc	reloc

Code Snippet

File Name	radareorg@@radare2-5.5.2-CVE-2022-1296-TP.c
Method	RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
476.                                    free (reloc);
....
542.                                free (reloc);
```

Double Free\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=548
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1297-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1297-TP.c
Line	476	542
Object	reloc	reloc

Code Snippet	
File Name	radareorg@@radare2-5.5.2-CVE-2022-1297-TP.c
Method	RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
476.                                    free (reloc);
....
542.                                free (reloc);
```

Double Free\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=549
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-1382-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1382-TP.c
Line	476	542
Object	reloc	reloc

Code Snippet	
File Name	radareorg@@radare2-5.5.2-CVE-2022-1382-TP.c
Method	RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
476.                                    free (reloc);
....
542.                                free (reloc);
```

Double Free\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=550
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2023-1605-TP.c	radareorg@@radare2-5.5.2-CVE-2023-1605-TP.c
Line	320	451
Object	rel	rel

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2023-1605-TP.c
Method static RList *_relocs_list(RBin *rbin, struct r_bin_coff_obj *bin, bool patch, ut64 imp_map) {

```
....
320.                    free (rel);
....
451.                free (rel);
```

Double Free\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=551
Status	New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-1237-TP.c	radareorg@@radare2-5.6.6-CVE-2022-1237-TP.c
Line	483	549
Object	reloc	reloc

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-1237-TP.c
Method RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
483.                                    free (reloc);
....
549.                                free (reloc);
```

Double Free\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=552
Status	New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-1238-TP.c	radareorg@@radare2-5.6.6-CVE-2022-1238-TP.c
Line	483	549
Object	reloc	reloc

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-1238-TP.c
Method RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
483.                                    free (reloc);
....
549.                                free (reloc);
```

Double Free\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=553
Status	New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-1283-TP.c	radareorg@@radare2-5.6.6-CVE-2022-1283-TP.c
Line	483	549
Object	reloc	reloc

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-1283-TP.c
Method RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
483.                                    free (reloc);
....
549.                                free (reloc);
```

Double Free\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=554
Status	New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-1296-TP.c	radareorg@@radare2-5.6.6-CVE-2022-1296-TP.c
Line	483	549
Object	reloc	reloc

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-1296-TP.c
Method RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
483.                                    free (reloc);
....
549.                                free (reloc);
```

Double Free\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=555
Status	New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-1297-TP.c	radareorg@@radare2-5.6.6-CVE-2022-1297-TP.c
Line	483	549
Object	reloc	reloc

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-1297-TP.c
Method RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
483.                                    free (reloc);
....
549.                                free (reloc);
```

Double Free\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=556
Status	New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-1382-TP.c	radareorg@@radare2-5.6.6-CVE-2022-1382-TP.c
Line	483	549
Object	reloc	reloc

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-1382-TP.c
Method RList *r_bin_ne_get_relocs(r_bin_ne_obj_t *bin) {

```
....
483.                                    free (reloc);
....
549.                                free (reloc);
```

Double Free\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=557
Status	New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2023-1605-TP.c	radareorg@@radare2-5.6.6-CVE-2023-1605-TP.c
Line	323	454
Object	rel	rel

Code Snippet
File Name radareorg@@radare2-5.6.6-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);
```

Use of Uninitialized Pointer

Query Path:
CPP\Cx\CPP Medium Threat\Use of Uninitialized Pointer Version:0

Categories

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

Description

Use of Uninitialized Pointer\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=879
Status	New

The variable declared in range at radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c in line 952 is not initialized when it is used by end at radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c in line 952.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	956	958
Object	range	end

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method static RBinDwarfLocRange *find_largest_loc_range (RList *loc_list) {

```
....
956.        RBinDwarfLocRange *range;
....
958.              ut64 diff = range->end - range->start;
```

Use of Uninitialized Pointer\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=880
Status	New

The variable declared in range at radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c in line 952 is not initialized when it is used by range at radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c in line 952.

Source	Destination

File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	956	961
Object	range	range

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method static RBinDwarfLocRange *find_largest_loc_range (RList *loc_list) {

```
....
956.        RBinDwarfLocRange *range;
....
961.                    largest = range;
```

Use of Uninitialized Pointer\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=881
Status	New

The variable declared in range at radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c in line 952 is not initialized when it is used by start at radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c in line 952.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	956	958
Object	range	start

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method static RBinDwarfLocRange *find_largest_loc_range (RList *loc_list) {

```
....
956.        RBinDwarfLocRange *range;
....
958.              ut64 diff = range->end - range->start;
```

Use of Uninitialized Pointer\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=882
Status	New

The variable declared in memory at radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c in line 9 is not initialized when it is used by data_size at radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c in line 9.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c
Line	12	23
Object	memory	data_size

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c
Method ut64 r_bin_mdmp_get_paddr(struct r_bin_mdmp_obj *obj, ut64 vaddr) {

```
....
12.   struct minidump_memory_descriptor64 *memory;
....
23.         index += memory->data_size;
```

Use of Uninitialized Pointer\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=883
Status	New

The variable declared in memory at radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c in line 9 is not initialized when it is used by start_of_memory_range at radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c in line 9.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c
Line	12	19
Object	memory	start_of_memory_range

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c
Method ut64 r_bin_mdmp_get_paddr(struct r_bin_mdmp_obj *obj, ut64 vaddr) {

```
....
12.   struct minidump_memory_descriptor64 *memory;
....
19.         if (vaddr == memory->start_of_memory_range) {
```

Use of Uninitialized Pointer\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=884
Status	New

The variable declared in mem_info at radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c in line 28 is not initialized when it is used by base_address at radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c in line 28.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c
Line	29	37
Object	mem_info	base_address

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c
Method struct minidump_memory_info *r_bin_mdmp_get_mem_info(struct r_bin_mdmp_obj *obj, ut64 vaddr) {

```
....
29.   struct minidump_memory_info *mem_info;
....
37.         if (mem_info->allocation_base && vaddr == mem_info->base_address) {
```

Use of Uninitialized Pointer\Path 7:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=885
Status New

The variable declared in mem_info at radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c in line 28 is not initialized when it is used by allocation_base at radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c in line 28.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c
Line	29	37
Object	mem_info	allocation_base

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c
Method struct minidump_memory_info *r_bin_mdmp_get_mem_info(struct r_bin_mdmp_obj *obj, ut64 vaddr) {

```
....
29.   struct minidump_memory_info *mem_info;
....
37.         if (mem_info->allocation_base && vaddr == mem_info->base_address) {
```

Use of Uninitialized Pointer\Path 8:

Severity Medium

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=886
Status	New

The variable declared in mem_info at radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c in line 28 is not initialized when it is used by mem_info at radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c in line 28.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c
Line	29	38
Object	mem_info	mem_info

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c
Method struct minidump_memory_info *r_bin_mdmp_get_mem_info(struct r_bin_mdmp_obj *obj, ut64 vaddr) {

```
....
29.   struct minidump_memory_info *mem_info;
....
38.               return mem_info;
```

Use of Uninitialized Pointer\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=887
Status	New

The variable declared in pe32_dup at radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c in line 978 is not initialized when it is used by vaddr at radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c in line 978.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c
Line	982	1003
Object	pe32_dup	vaddr

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c
Method static bool r_bin_mdmp_init_pe_bins(struct r_bin_mdmp_obj *obj) {

```
....
982.        struct Pe32_r_bin_mdmp_pe_bin *pe32_bin, *pe32_dup;
....
1003.                           if (pe32_dup->vaddr == module-
>base_of_image) {
```

Use of Uninitialized Pointer\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=888
Status	New

The variable declared in pe64_dup at radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c in line 978 is not initialized when it is used by pe64_dup at radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c in line 978.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c
Line	983	1022
Object	pe64_dup	pe64_dup

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0476-TP.c
Method static bool r_bin_mdmp_init_pe_bins(struct r_bin_mdmp_obj *obj) {

```
....
983.        struct Pe64_r_bin_mdmp_pe_bin *pe64_bin, *pe64_dup;
....
1022.                           if (pe64_dup->vaddr == module-
>base_of_image) {
```

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=1020061&projectid=20051&pathid=491
Status	New

The function size in radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c at line 168 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@@radare2-5.5.2-CVE-2022-0520-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Line	208	208
Object	size	size

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0520-TP.c
Method static pyc_object *get_long_object(RBuffer *buffer) {

```
....
208.              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=1020061&projectid=20051&pathid=492
Status	New

The function size in radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c at line 168 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@@radare2-5.5.2-CVE-2022-0523-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Line	208	208
Object	size	size

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0523-TP.c
Method static pyc_object *get_long_object(RBuffer *buffer) {

```
....
208.              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=1020061&projectid=20051&pathid=493
Status	New

The function size in radareorg@@radare2-5.6.6-CVE-2022-0523-FP.c at line 166 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@@radare2-5.6.6-CVE-2022-0523-FP.c	radareorg@@radare2-5.6.6-CVE-2022-0523-FP.c
Line	207	207
Object	size	size

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-0523-FP.c
Method static pyc_object *get_long_object(RBuffer *buffer) {

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

Wrong Size t Allocation\Path 4:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=494
Status New

The function newlen in radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c at line 92 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@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	100	100
Object	newlen	newlen

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method static bool strbuf_rev_prepend_char(RStrBuf *sb, const char *s, int c) {

```
....
100.        char *ns = malloc (newlen + 1);
```

Wrong Size t Allocation\Path 5:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=495
Status New

The function newlen in radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c at line 127 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@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	143	143
Object	newlen	newlen

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method static bool strbuf_rev_append_char(RStrBuf *sb, const char *s, const char *needle) {

```
....
143.        char *ns = malloc (newlen + 1);
```

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=1020061&projectid=20051&pathid=889
Status	New

The variable declared in extra at radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c in line 1612 is not initialized when it is used by extra at radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c in line 1612.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	1649	1652
Object	extra	extra

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method ls_foreach (sdb_list, it, kv) {

```
....
1649.                   char *extra = NULL;
....
1652.                   extra = sdb_anext (extra, &type);
```

Use of Zero Initialized Pointer\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=890
Status	New

The variable declared in type at radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c in line 1612 is not initialized when it is used by extra at radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c in line 1612.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	1651	1652
Object	type	extra

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method ls_foreach (sdb_list, it, kv) {

```
....
1651.                   char *type = NULL;
1652.                   extra = sdb_anext (extra, &type);
```

Use of Zero Initialized Pointer\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=891
Status	New

The variable declared in header at radareorg@@radare2-5.6.6-CVE-2022-0695-FP.c in line 124 is not initialized when it is used by header at radareorg@@radare2-5.6.6-CVE-2022-0695-FP.c in line 106.

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-0695-FP.c	radareorg@@radare2-5.6.6-CVE-2022-0695-FP.c
Line	125	107
Object	header	header

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-0695-FP.c
Method static int r_bin_te_init(struct r_bin_te_obj_t* bin) {

```
....
125.        bin->header = NULL;
```

File Name	radareorg@@radare2-5.6.6-CVE-2022-0695-FP.c
Method	static int r_bin_te_init_sections(struct r_bin_te_obj_t* bin) {

```
....
107.        int sections_size = sizeof(TE_image_section_header) * bin-
>header->NumberOfSections;
```

Buffer Overflow AddressOfLocalVarReturned

Query Path:
CPP\Cx\CPP Buffer Overflow\Buffer Overflow AddressOfLocalVarReturned Version:1

Categories

PCI DSS v3.2: PCI DSS (3.2) - 6.5.2 - Buffer overflows
NIST SP 800-53: SC-5 Denial of Service Protection (P1)
OWASP Top 10 2017: A1-Injection

Description
Buffer Overflow AddressOfLocalVarReturned\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=296
Status	New

The pointer b at radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c in line 446 is being used after it has been freed.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c
Line	448	448
Object	b	b

Code Snippet

File Name	radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c
Method	static int bb_cmpaddr(const void *_a, const void *_b) {

```
....
448.        return a->addr > b->addr ? 1 : (a->addr < b->addr ? -1 : 0);
```

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=1020061&projectid=20051&pathid=297
Status	New

The size of the buffer used by r_bin_te_get_sections in TE_IMAGE_SIZEOF_NAME, at line 333 of radareorg@@radare2-5.6.6-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 333 of radareorg@@radare2-5.6.6-CVE-2022-0695-FP.c, to overwrite the target buffer.

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-0695-FP.c	radareorg@@radare2-5.6.6-CVE-2022-0695-FP.c
Line	348	348
Object	TE_IMAGE_SIZEOF_NAME	TE_IMAGE_SIZEOF_NAME

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-0695-FP.c
Method struct r_bin_te_section_t* r_bin_te_get_sections(struct r_bin_te_obj_t* bin) {

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

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=1020061&projectid=20051&pathid=892
Status	New

The function malloc in radareorg@@radare2-5.6.6-CVE-2022-0695-FP.c at line 18 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@@radare2-5.6.6-CVE-2022-0695-FP.c	radareorg@@radare2-5.6.6-CVE-2022-0695-FP.c
Line	22	22

Object	sizeof	malloc

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-0695-FP.c
Method static int r_bin_te_init_hdr(struct r_bin_te_obj_t *bin) {

```
....
22.   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=1020061&projectid=20051&pathid=1
Status	New

The malloc method calls the malloc function, at line 481 of radareorg@@radare2-5.4.0-CVE-2022-28071-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@@radare2-5.4.0-CVE-2022-28071-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c
Line	481	481
Object	malloc	malloc

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28071-TP.c
Method ut8 *buf = malloc (bsize);

```
....
481.        ut8 *buf = malloc (bsize);
```

Unchecked Return Value\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=2
Status	New

The handle_switch_op method calls the snprintf function, at line 104 of radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	108	108
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Method static int handle_switch_op (ut64 addr, const ut8 * bytes, char *output, int outlen) {

```
....
108.        snprintf (output, outlen, "case %d: goto 0x%04x", ccase,
jmp);
```

Unchecked Return Value\Path 3:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=3
Status New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	129	129
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.4.0-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) {

```
....
129.             snprintf (output, outlen, "%s %d", JAVA_OPS[idx].name,
(char) bytes[1]);
```

Unchecked Return Value\Path 4:

Severity Low
Result State To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=4
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	133	133
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.4.0-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) {

```
....
133.              snprintf (output, outlen, "%s %d", JAVA_OPS[idx].name,
(int)USHORT (bytes, 1));
```

Unchecked Return Value\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=5
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	147	147
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.4.0-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) {

```
....
147.              snprintf (output, outlen, "%s %d", JAVA_OPS[idx].name,
bytes[1]);
```

Unchecked Return Value\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=6
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	154	154
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.4.0-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) {

```
....
154.                    snprintf (output, outlen, "%s %s",
JAVA_OPS[idx].name, arg);
```

Unchecked Return Value\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=7
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	157	157
Object	snprintf	snprintf

Code Snippet	
File Name	radareorg@@radare2-5.4.0-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) {

```
....
157.                    snprintf (output, outlen, "%s #%d",
JAVA_OPS[idx].name, USHORT (bytes, 1));
```

Unchecked Return Value\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=8
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	165	165
Object	snprintf	snprintf

Code Snippet	
File Name	radareorg@@radare2-5.4.0-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) {

```
....
165.                    snprintf (output, outlen, "%s %s",
JAVA_OPS[idx].name, arg);
```

Unchecked Return Value\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=9
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	168	168
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.4.0-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) {

```
....
168.                    snprintf (output, outlen, "%s #%d",
JAVA_OPS[idx].name, USHORT (bytes, 1));
```

Unchecked Return Value\Path 10:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=10
Status New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	175	175
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.4.0-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) {

```
....
175.                snprintf (output, outlen, "%s %d %d",
JAVA_OPS[idx].name, val_one, val_two);
```

Unchecked Return Value\Path 11:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=11
Status New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	213	213
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.4.0-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) {

```
....
213.                    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=1020061&projectid=20051&pathid=12
Status New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	216	216
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.4.0-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) {

```
....
216.                    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=1020061&projectid=20051&pathid=13
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	226	226
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.4.0-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) {

```
....
226.                    snprintf (output, outlen, "%s %s",
JAVA_OPS[idx].name, arg);
```

Unchecked Return Value\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=14
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	229	229
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.4.0-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) {

```
....
229.                    snprintf (output, outlen, "%s #%d",
JAVA_OPS[idx].name, USHORT (bytes, 1) );
```

Unchecked Return Value\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=15
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	239	239
Object	snprintf	snprintf

Code Snippet

File Name radareorg@@radare2-5.4.0-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) {

```
....
239.                    snprintf (output, outlen, "%s %s",
JAVA_OPS[idx].name, arg);
```

Unchecked Return Value\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=16
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	242	242
Object	snprintf	snprintf

Code Snippet	
File Name	radareorg@@radare2-5.4.0-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) {

```
....
242.                    snprintf (output, outlen, "%s #%d",
JAVA_OPS[idx].name, USHORT (bytes, 1) );
```

Unchecked Return Value\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=17
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	250	250
Object	snprintf	snprintf

Code Snippet	
File Name	radareorg@@radare2-5.4.0-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) {

```
....
250.        case 1: snprintf (output, outlen, "%s", JAVA_OPS[idx].name);
```

Unchecked Return Value\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=18
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-	radareorg@@radare2-5.4.0-CVE-2023-

	5686-TP.c	5686-TP.c
Line	252	252
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.4.0-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) {

```
....
252.        case 2: snprintf (output, outlen, "%s %d",
JAVA_OPS[idx].name, bytes[1]);
```

Unchecked Return Value\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=19
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	254	254
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.4.0-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) {

```
....
254.        case 3: snprintf (output, outlen, "%s 0x%04x 0x%04x",
JAVA_OPS[idx].name, bytes[0], bytes[1]);
```

Unchecked Return Value\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=20
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.4.0-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@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	256	256
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.4.0-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) {

```
....
256.        case 5: snprintf (output, outlen, "%s %d",
JAVA_OPS[idx].name, bytes[1]);
```

Unchecked Return Value\Path 21:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=21
Status New

The malloc method calls the malloc function, at line 192 of radareorg@@radare2-5.5.2-CVE-2022-0712-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@@radare2-5.5.2-CVE-2022-0712-TP.c	radareorg@@radare2-5.5.2-CVE-2022-0712-TP.c
Line	192	192
Object	malloc	malloc

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-0712-TP.c
Method ut8 *b = malloc (size);

```
....
192.        ut8 *b = malloc (size);
```

Unchecked Return Value\Path 22:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20

	051&pathid=22
Status	New

The malloc method calls the malloc function, at line 192 of radareorg@@radare2-5.5.2-CVE-2022-1061-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@@radare2-5.5.2-CVE-2022-1061-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1061-TP.c
Line	192	192
Object	malloc	malloc

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1061-TP.c
Method ut8 *b = malloc (size);

```
....
192.        ut8 *b = malloc (size);
```

Unchecked Return Value\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=23
Status	New

The *__resource_type_str method calls the strdup function, at line 174 of radareorg@@radare2-5.5.2-CVE-2022-1237-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@@radare2-5.5.2-CVE-2022-1237-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c
Line	246	246
Object	strdup	strdup

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1237-TP.c
Method static char *__resource_type_str(int type) {

```
....
246.        return strdup (typeName);
```

Unchecked Return Value\Path 24:

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

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=24
Status	New

The *__resource_type_str method calls the strdup function, at line 174 of radareorg@@radare2-5.5.2-CVE-2022-1238-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@@radare2-5.5.2-CVE-2022-1238-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1238-TP.c
Line	246	246
Object	strdup	strdup

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1238-TP.c
Method static char *__resource_type_str(int type) {

```
....
246.        return strdup (typeName);
```

Unchecked Return Value\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=25
Status	New

The *__resource_type_str method calls the strdup function, at line 174 of radareorg@@radare2-5.5.2-CVE-2022-1283-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@@radare2-5.5.2-CVE-2022-1283-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1283-TP.c
Line	246	246
Object	strdup	strdup

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1283-TP.c
Method static char *__resource_type_str(int type) {

```
....
246.        return strdup (typeName);
```

Unchecked Return Value\Path 26:

Severity	Low
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=26
Status	New

The *__resource_type_str method calls the strdup function, at line 174 of radareorg@@radare2-5.5.2-CVE-2022-1296-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@@radare2-5.5.2-CVE-2022-1296-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1296-TP.c
Line	246	246
Object	strdup	strdup

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1296-TP.c
Method static char *__resource_type_str(int type) {

```
....
246.        return strdup (typeName);
```

Unchecked Return Value\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=27
Status	New

The *__resource_type_str method calls the strdup function, at line 174 of radareorg@@radare2-5.5.2-CVE-2022-1297-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@@radare2-5.5.2-CVE-2022-1297-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1297-TP.c
Line	246	246
Object	strdup	strdup

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1297-TP.c
Method static char *__resource_type_str(int type) {

```
....
246.        return strdup (typeName);
```

Unchecked Return Value\Path 28:

Severity	Low

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=28
Status	New

The *__resource_type_str method calls the strdup function, at line 174 of radareorg@@radare2-5.5.2-CVE-2022-1382-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@@radare2-5.5.2-CVE-2022-1382-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1382-TP.c
Line	246	246
Object	strdup	strdup

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1382-TP.c
Method static char *__resource_type_str(int type) {

```
....
246.        return strdup (typeName);
```

Unchecked Return Value\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=29
Status	New

The malloc method calls the malloc function, at line 192 of radareorg@@radare2-5.5.2-CVE-2022-1437-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@@radare2-5.5.2-CVE-2022-1437-TP.c	radareorg@@radare2-5.5.2-CVE-2022-1437-TP.c
Line	192	192
Object	malloc	malloc

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2022-1437-TP.c
Method ut8 *b = malloc (size);

```
....
192.        ut8 *b = malloc (size);
```

Unchecked Return Value\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=30
Status	New

The handle_switch_op method calls the snprintf function, at line 104 of radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	108	108
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Method static int handle_switch_op (ut64 addr, const ut8 * bytes, char *output, int outlen) {

```
....
108.        snprintf (output, outlen, "case %d: goto 0x%04x", ccase,
jmp);
```

Unchecked Return Value\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=31
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	129	129
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.5.2-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) {

```
....
129.              snprintf (output, outlen, "%s %d", JAVA_OPS[idx].name,
(char) bytes[1]);
```

Unchecked Return Value\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=32
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	133	133
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.5.2-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) {

```
....
133.              snprintf (output, outlen, "%s %d", JAVA_OPS[idx].name,
(int)USHORT (bytes, 1));
```

Unchecked Return Value\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=33
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	147	147
Object	snprintf	snprintf

Code Snippet	
File Name	radareorg@@radare2-5.5.2-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) {

```
....
147.              snprintf (output, outlen, "%s %d", JAVA_OPS[idx].name,
bytes[1]);
```

Unchecked Return Value\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=34
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	154	154
Object	snprintf	snprintf

Code Snippet	
File Name	radareorg@@radare2-5.5.2-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) {

```
....
154.                    snprintf (output, outlen, "%s %s",
JAVA_OPS[idx].name, arg);
```

Unchecked Return Value\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=35
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	157	157
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.5.2-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) {

```
....
157.                    snprintf (output, outlen, "%s #%d",
JAVA_OPS[idx].name, USHORT (bytes, 1));
```

Unchecked Return Value\Path 36:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=36
Status New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	165	165
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.5.2-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) {

```
....
165.                    snprintf (output, outlen, "%s %s",
JAVA_OPS[idx].name, arg);
```

Unchecked Return Value\Path 37:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=37
Status New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	168	168
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.5.2-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) {

```
....
168.                    snprintf (output, outlen, "%s #%d",
JAVA_OPS[idx].name, USHORT (bytes, 1));
```

Unchecked Return Value\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=38
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	175	175
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.5.2-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) {

```
....
175.                snprintf (output, outlen, "%s %d %d",
JAVA_OPS[idx].name, val_one, val_two);
```

Unchecked Return Value\Path 39:

Severity	Low
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=39
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	213	213
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.5.2-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) {

```
....
213.                    snprintf (output, outlen, "%s %s",
JAVA_OPS[idx].name, arg);
```

Unchecked Return Value\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=40
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	216	216
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.5.2-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) {

```
....
216.                    snprintf (output, outlen, "%s #%d",
JAVA_OPS[idx].name, USHORT (bytes, 1) );
```

Unchecked Return Value\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=41
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	226	226
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.5.2-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) {

```
....
226.                    snprintf (output, outlen, "%s %s",
JAVA_OPS[idx].name, arg);
```

Unchecked Return Value\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=42
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	229	229
Object	snprintf	snprintf

Code Snippet	
File Name	radareorg@@radare2-5.5.2-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) {

```
....
229.                    snprintf (output, outlen, "%s #%d",
JAVA_OPS[idx].name, USHORT (bytes, 1) );
```

Unchecked Return Value\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=43
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	239	239
Object	snprintf	snprintf

Code Snippet	
File Name	radareorg@@radare2-5.5.2-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) {

```
....
239.                    snprintf (output, outlen, "%s %s",
JAVA_OPS[idx].name, arg);
```

Unchecked Return Value\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=44
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	242	242
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.5.2-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) {

```
....
242.                    snprintf (output, outlen, "%s #%d",
JAVA_OPS[idx].name, USHORT (bytes, 1) );
```

Unchecked Return Value\Path 45:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=45
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	250	250
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.5.2-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) {

```
....
250.        case 1: snprintf (output, outlen, "%s", JAVA_OPS[idx].name);
```

Unchecked Return Value\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=46
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	252	252
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.5.2-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) {

```
....
252.        case 2: snprintf (output, outlen, "%s %d",
JAVA_OPS[idx].name, bytes[1]);
```

Unchecked Return Value\Path 47:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=47
Status New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	254	254
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.5.2-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) {

```
....
254.        case 3: snprintf (output, outlen, "%s 0x%04x 0x%04x",
JAVA_OPS[idx].name, bytes[0], bytes[1]);
```

Unchecked Return Value\Path 48:

Severity Low
Result State To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=48
Status	New

The java_print_opcode method calls the snprintf function, at line 113 of radareorg@@radare2-5.5.2-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@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	256	256
Object	snprintf	snprintf

Code Snippet
File Name radareorg@@radare2-5.5.2-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) {

```
....
256.        case 5: snprintf (output, outlen, "%s %d",
JAVA_OPS[idx].name, bytes[1]);
```

Unchecked Return Value\Path 49:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=49
Status	New

The malloc method calls the malloc function, at line 192 of radareorg@@radare2-5.6.6-CVE-2022-1061-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@@radare2-5.6.6-CVE-2022-1061-TP.c	radareorg@@radare2-5.6.6-CVE-2022-1061-TP.c
Line	192	192
Object	malloc	malloc

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-1061-TP.c
Method ut8 *b = malloc (size);

```
....
192.        ut8 *b = malloc (size);
```

Unchecked Return Value\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=50
Status	New

The *__resource_type_str method calls the strdup function, at line 181 of radareorg@@radare2-5.6.6-CVE-2022-1237-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@@radare2-5.6.6-CVE-2022-1237-TP.c	radareorg@@radare2-5.6.6-CVE-2022-1237-TP.c
Line	253	253
Object	strdup	strdup

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-1237-TP.c
Method static char *__resource_type_str(int type) {

```
....
253.        return strdup (typeName);
```

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=1020061&projectid=20051&pathid=499
Status	New

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	308	308
Object	name	sizeof

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Method R_API int r_java_assemble(ut64 addr, ut8 *bytes, const char *string) {

```
....
308.        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=1020061&projectid=20051&pathid=500
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	308	308
Object	name	sizeof

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Method R_API int r_java_assemble(ut64 addr, ut8 *bytes, const char *string) {

```
....
308.        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=1020061&projectid=20051&pathid=501
Status	New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2023-5686-TP.c	radareorg@@radare2-5.6.6-CVE-2023-5686-TP.c
Line	308	308
Object	name	sizeof

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2023-5686-TP.c
Method R_API int r_java_assemble(ut64 addr, ut8 *bytes, const char *string) {

```
....
308.        name[sizeof (name) - 1] = 0;
```

Sizeof Pointer Argument\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=502

Status	New

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c	radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Line	307	307
Object	name	sizeof

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2023-5686-TP.c
Method R_API int r_java_assemble(ut64 addr, ut8 *bytes, const char *string) {

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

Sizeof Pointer Argument\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=503
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c	radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Line	307	307
Object	name	sizeof

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2023-5686-TP.c
Method R_API int r_java_assemble(ut64 addr, ut8 *bytes, const char *string) {

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

Sizeof Pointer Argument\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=504
Status	New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2023-5686-TP.c	radareorg@@radare2-5.6.6-CVE-2023-5686-TP.c

Line	307	307
Object	name	sizeof

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2023-5686-TP.c
Method R_API int r_java_assemble(ut64 addr, ut8 *bytes, const char *string) {

```
....
307.        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=1020061&projectid=20051&pathid=496
Status	New

The variable declared in null at radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c in line 517 is not initialized when it is used by ret at radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c in line 517.

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c	radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c
Line	521	521
Object	null	ret

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2023-1605-TP.c
Method static RBinInfo *info(RBinFile *bf) {

```
....
521.        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=1020061&projectid=20051&pathid=497
Status	New

The variable declared in null at radareorg@@radare2-5.5.2-CVE-2023-1605-TP.c in line 519 is not initialized when it is used by ret at radareorg@@radare2-5.5.2-CVE-2023-1605-TP.c in line 519.

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2023-1605-TP.c	radareorg@@radare2-5.5.2-CVE-2023-1605-TP.c
Line	523	523
Object	null	ret

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2023-1605-TP.c
Method static RBinInfo *info(RBinFile *bf) {

```
....
523.        ret->file = bf->file? strdup (bf->file): NULL;
```

NULL Pointer Dereference\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=498
Status	New

The variable declared in null at radareorg@@radare2-5.6.6-CVE-2023-1605-TP.c in line 522 is not initialized when it is used by ret at radareorg@@radare2-5.6.6-CVE-2023-1605-TP.c in line 522.

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2023-1605-TP.c	radareorg@@radare2-5.6.6-CVE-2023-1605-TP.c
Line	526	526
Object	null	ret

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2023-1605-TP.c
Method static RBinInfo *info(RBinFile *bf) {

```
....
526.        ret->file = bf->file? strdup (bf->file): NULL;
```

Unchecked Array Index

Query Path:
CPP\Cx\CPP Low Visibility\Unchecked Array Index Version:1

Categories

NIST SP 800-53: SI-10 Information Input Validation (P1)

Description

Unchecked Array Index\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=505
Status	New

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	113	113
Object	newlen	newlen

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method static bool strbuf_rev_prepend_char(RStrBuf *sb, const char *s, int c) {

```
....
113.                ns[newlen] = 0;
```

Unchecked Array Index\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=506
Status	New

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	148	148
Object	newlen	newlen

Code Snippet
File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Method static bool strbuf_rev_append_char(RStrBuf *sb, const char *s, const char *needle) {

```
....
148.                ns[newlen] = 0;
```

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=1020061&projectid=20051&pathid=894
Status	New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-0520-FP.c	radareorg@@radare2-5.6.6-CVE-2022-0520-FP.c
Line	48	48
Object	f	f

Code Snippet
File Name radareorg@@radare2-5.6.6-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=1020061&projectid=20051&pathid=895
Status	New

	Source	Destination
File	radareorg@@radare2-5.5.2-CVE-2020-15121-FP.c	radareorg@@radare2-5.5.2-CVE-2020-15121-FP.c
Line	38	38
Object	CreateDirectory	CreateDirectory

Code Snippet
File Name radareorg@@radare2-5.5.2-CVE-2020-15121-FP.c
Method static bool r_sys_mkdir(const char *path) {

```
....
38.   bool ret = CreateDirectory (path_, NULL);
```

Use of Sizeof On a Pointer Type

Query Path:
CPP\Cx\CPP Low Visibility\Use of Sizeof On a Pointer Type Version:1

Description

Use of Sizeof On a Pointer Type\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1020061&projectid=20051&pathid=295
Status	New

	Source	Destination
File	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c	radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c
Line	271	271
Object	sizeof	sizeof

Code Snippet

File Name radareorg@@radare2-5.4.0-CVE-2022-28068-TP.c

Method static st32 parse_type(Context *ctx, const ut64 offset, RStrBuf *strbuf, ut64 *size, HtUP **visited) {

```
....
271.              visited = malloc (sizeof (void*));
```

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=1020061&projectid=20051&pathid=893
Status	New

	Source	Destination
File	radareorg@@radare2-5.6.6-CVE-2022-0520-FP.c	radareorg@@radare2-5.6.6-CVE-2022-0520-FP.c
Line	50	50
Object	fwrite	fwrite

Code Snippet

File Name radareorg@@radare2-5.6.6-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);
```

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=1020061&projectid=20051&pathid=896
Status	New

The download_and_write method in radareorg@@radare2-5.6.6-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@@radare2-5.6.6-CVE-2022-0520-FP.c	radareorg@@radare2-5.6.6-CVE-2022-0520-FP.c
Line	48	48
Object	fopen	fopen

Code Snippet
File Name radareorg@@radare2-5.6.6-CVE-2022-0520-FP.c
Method static bool download_and_write(SPDBDownloaderOpt *opt, const char *file) {

```
....
48.   FILE *f = fopen (path, "wb");
```

Buffer Overflow AddressOfLocalVarReturned

Risk

What might happen

A use after free error will cause code to use an area of memory previously assigned with a specific value, which has since been freed and may have been overwritten by another value. This error will likely cause unexpected behavior, memory corruption and crash errors. In some cases where the freed and used section of memory is used to determine execution flow, and the error can be induced by an attacker, this may result in execution of malicious code.

Cause

How does it happen

Pointers to variables allow code to have an address with a set size to a dynamically allocated variable. Eventually, the pointer's destination may become free - either explicitly in code, such as when programmatically freeing this variable, or implicitly, such as when a local variable is returned - once it is returned, the variable's scope is released. Once freed, this memory will be re-used by the application,

overwritten with new data. At this point, dereferencing this pointer will potentially resolve newly written and unexpected data.

General Recommendations

How to avoid it

- Do not return local variables or pointers
- Review code to ensure no flow allows use of a pointer after it has been explicitly freed

Source Code Examples

CPP

Use of Variable after It was Freed

```
free(input);
printf("%s", input);
```

Use of Pointer to Local Variable That Was Freed On Return

```
int* func1()
{
	int i;
	i = 1;
	return &i;
}

void func2()
{
	int j;
	j = 5;
}

//..
	int * i = func1();
	printf("%d\r\n", *i); // Output could be 1 or Segmentation Fault
	func2();
	printf("%d\r\n", *i); // Output is 5, which is j's value, as func2() overwrote data in
the stack
//..
```

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

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 Uninitialized 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

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

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());
```

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

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);
```

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';
```

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

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

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 Validation of Array Index

Weakness ID: 129 *(Weakness Base)* **Status:** Draft

Description

Description Summary

The product uses untrusted input when calculating or using an array index, but the product does not validate or incorrectly validates the index to ensure the index references a valid position within the array.

Alternate Terms

out-of-bounds array index

index-out-of-range

array index underflow

Time of Introduction

- Implementation

Applicable Platforms

Languages

C: *(Often)*

C++: *(Often)*

Language-independent

Common Consequences

Scope	Effect
Integrity Availability	Unchecked array indexing will very likely result in the corruption of relevant memory and perhaps instructions, leading to a crash, if the values are outside of the valid memory area.
Integrity	If the memory corrupted is data, rather than instructions, the system will continue to function with improper values.
Confidentiality Integrity	Unchecked array indexing can also trigger out-of-bounds read or write operations, or operations on the wrong objects; i.e., "buffer overflows" are not always the result. This may result in the exposure or modification of sensitive data.
Integrity	If the memory accessible by the attacker can be effectively controlled, it may be possible to execute arbitrary code, as with a standard buffer overflow and possibly without the use of large inputs if a precise index can be controlled.
Integrity Availability Confidentiality	A single fault could allow either an overflow (CWE-788) or underflow (CWE-786) of the array index. What happens next will depend on the type of operation being performed out of bounds, but can expose sensitive information, cause a system crash, or possibly lead to arbitrary code execution.

Likelihood of Exploit

High

Detection Methods

Automated Static Analysis

This weakness can often be detected using automated static analysis tools. Many modern tools use data flow analysis or constraint-based techniques to minimize the number of false positives.

Automated static analysis generally does not account for environmental considerations when reporting out-of-bounds memory operations. This can make it difficult for users to determine which warnings should be investigated first. For example, an analysis tool might report array index errors that originate from command line arguments in a program that is not expected to run with setuid or other special privileges.

Effectiveness: High

This is not a perfect solution, since 100% accuracy and coverage are not feasible.

Automated Dynamic Analysis

This weakness can be detected using dynamic tools and techniques that interact with the software using large test suites with many diverse inputs, such as fuzz testing (fuzzing), robustness testing, and fault injection. The software's operation may slow down, but it should not become unstable, crash, or generate incorrect results.

Black Box

Black box methods might not get the needed code coverage within limited time constraints, and a dynamic test might not produce any noticeable side effects even if it is successful.

Demonstrative Examples

Example 1

The following C/C++ example retrieves the sizes of messages for a pop3 mail server. The message sizes are retrieved from a socket that returns in a buffer the message number and the message size, the message number (num) and size (size) are extracted from the buffer and the message size is placed into an array using the message number for the array index.

(Bad Code)

Example Language: **C**

```
/* capture the sizes of all messages */
int getsizes(int sock, int count, int *sizes) {
...
char buf[BUFFER_SIZE];
int ok;
int num, size;

// read values from socket and added to sizes array
while ((ok = gen_recv(sock, buf, sizeof(buf))) == 0)
{

// continue read from socket until buf only contains '.'
if (DOTLINE(buf))
break;
else if (sscanf(buf, "%d %d", &num, &size) == 2)
sizes[num - 1] = size;
}
...
}
```

In this example the message number retrieved from the buffer could be a value that is outside the allowable range of indices for the array and could possibly be a negative number. Without proper validation of the value to be used for the array index an array overflow could occur and could potentially lead to unauthorized access to memory addresses and system crashes. The value of the array index should be validated to ensure that it is within the allowable range of indices for the array as in the following code.

(Good Code)

Example Language: **C**

```
/* capture the sizes of all messages */
int getsizes(int sock, int count, int *sizes) {
...
char buf[BUFFER_SIZE];
int ok;
int num, size;

// read values from socket and added to sizes array
while ((ok = gen_recv(sock, buf, sizeof(buf))) == 0)
{

// continue read from socket until buf only contains '.'
if (DOTLINE(buf))
```

```
break;
else if (sscanf(buf, "%d %d", &num, &size) == 2) {
if (num > 0 && num <= (unsigned)count)
sizes[num - 1] = size;
else
/* warn about possible attempt to induce buffer overflow */
report(stderr, "Warning: ignoring bogus data for message sizes returned by server.\n");
}
}
...
}
```

Example 2

In the code snippet below, an unchecked integer value is used to reference an object in an array.

(Bad Code)

Example Language: **Java**

```
public String getValue(int index) {
return array[index];
}
```

If index is outside of the range of the array, this may result in an ArrayIndexOutOfBounds Exception being raised.

Example 3

In the following Java example the method displayProductSummary is called from a Web service servlet to retrieve product summary information for display to the user. The servlet obtains the integer value of the product number from the user and passes it to the displayProductSummary method. The displayProductSummary method passes the integer value of the product number to the getProductSummary method which obtains the product summary from the array object containing the project summaries using the integer value of the product number as the array index.

(Bad Code)

Example Language: **Java**

```
// Method called from servlet to obtain product information
public String displayProductSummary(int index) {

String productSummary = new String("");

try {
String productSummary = getProductSummary(index);

} catch (Exception ex) {...}

return productSummary;
}

public String getProductSummary(int index) {
return products[index];
}
```

In this example the integer value used as the array index that is provided by the user may be outside the allowable range of indices for the array which may provide unexpected results or may comes the application to fail. The integer value used for the array index should be validated to ensure that it is within the allowable range of indices for the array as in the following code.

(Good Code)

Example Language: **Java**

```
// Method called from servlet to obtain product information
public String displayProductSummary(int index) {

String productSummary = new String("");
```

```
try {
String productSummary = getProductSummary(index);

} catch (Exception ex) {...}

return productSummary;
}

public String getProductSummary(int index) {
String productSummary = "";

if ((index >= 0) && (index < MAX_PRODUCTS)) {
productSummary = products[index];
}
else {
System.err.println("index is out of bounds");
throw new IndexOutOfBoundsException();
}

return productSummary;
}
```

An alternative in Java would be to use one of the collection objects such as ArrayList that will automatically generate an exception if an attempt is made to access an array index that is out of bounds.

(Good Code)

Example Language: **Java**

```
ArrayList productArray = new ArrayList(MAX_PRODUCTS);
...
try {
productSummary = (String) productArray.get(index);
} catch (IndexOutOfBoundsException ex) {...}
```

Observed Examples

Reference	Description
CVE-2005-0369	large ID in packet used as array index
CVE-2001-1009	negative array index as argument to POP LIST command
CVE-2003-0721	Integer signedness error leads to negative array index
CVE-2004-1189	product does not properly track a count and a maximum number, which can lead to resultant array index overflow.
CVE-2007-5756	chain: device driver for packet-capturing software allows access to an unintended IOCTL with resultant array index error.

Potential Mitigations

Phase: Architecture and Design

Strategies: Input Validation; Libraries or Frameworks

Use an input validation framework such as Struts or the OWASP ESAPI Validation API. If you use Struts, be mindful of weaknesses covered by the CWE-101 category.

Phase: Architecture and Design

For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side, in order to avoid CWE-602. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the server.

Even though client-side checks provide minimal benefits with respect to server-side security, they are still useful. First, they can support intrusion detection. If the server receives input that should have been rejected by the client, then it may be an indication of an attack. Second, client-side error-checking can provide helpful feedback to the user about the expectations for valid input. Third, there may be a reduction in server-side processing time for accidental input errors, although this is typically a small savings.

Phase: Requirements

Strategy: Language Selection

Use a language with features that can automatically mitigate or eliminate out-of-bounds indexing errors.

For example, Ada allows the programmer to constrain the values of a variable and languages such as Java and Ruby will allow the programmer to handle exceptions when an out-of-bounds index is accessed.

Phase: Implementation

Strategy: Input Validation

Assume all input is malicious. Use an "accept known good" input validation strategy (i.e., use a whitelist). Reject any input that does not strictly conform to specifications, or transform it into something that does. Use a blacklist to reject any unexpected inputs and detect potential attacks.

When accessing a user-controlled array index, use a stringent range of values that are within the target array. Make sure that you do not allow negative values to be used. That is, verify the minimum as well as the maximum of the range of acceptable values.

Phase: Implementation

Be especially careful to validate your input when you invoke code that crosses language boundaries, such as from an interpreted language to native code. This could create an unexpected interaction between the language boundaries. Ensure that you are not violating any of the expectations of the language with which you are interfacing. For example, even though Java may not be susceptible to buffer overflows, providing a large argument in a call to native code might trigger an overflow.

Weakness Ordinalities

Ordinality	Description
Resultant	The most common condition situation leading to unchecked array indexing is the use of loop index variables as buffer indexes. If the end condition for the loop is subject to a flaw, the index can grow or shrink unbounded, therefore causing a buffer overflow or underflow. Another common situation leading to this condition is the use of a function's return value, or the resulting value of a calculation directly as an index in to a buffer.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	20	Improper Input Validation	**Development Concepts (primary)699 Research Concepts (primary)1000**
ChildOf	Category	189	Numeric Errors	Development Concepts699
ChildOf	Category	633	Weaknesses that Affect Memory	**Resource-specific Weaknesses (primary)631**
ChildOf	Category	738	CERT C Secure Coding Section 04 - Integers (INT)	**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
ChildOf	Category	802	2010 Top 25 - Risky Resource Management	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
CanPrecede	Weakness Class	119	Failure to Constrain Operations within the Bounds of a Memory Buffer	Research Concepts1000
CanPrecede	Weakness Variant	789	Uncontrolled Memory Allocation	Research Concepts1000
PeerOf	Weakness Base	124	Buffer Underwrite ('Buffer Underflow')	Research Concepts1000

Theoretical Notes

An improperly validated array index might lead directly to the always-incorrect behavior of "access of array using out-of-bounds index."

Affected Resources

- Memory

f Causal Nature

Explicit

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Unchecked array indexing
PLOVER			INDEX - Array index overflow
CERT C Secure Coding	ARR00-C		Understand how arrays work
CERT C Secure Coding	ARR30-C		Guarantee that array indices are within the valid range
CERT C Secure Coding	ARR38-C		Do not add or subtract an integer to a pointer if the resulting value does not refer to a valid array element
CERT C Secure Coding	INT32-C		Ensure that operations on signed integers do not result in overflow

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
100	Overflow Buffers	

References

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 5, "Array Indexing Errors" Page 144. 2nd Edition. Microsoft. 2002.

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	CLASP		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Sean Eidemiller	Cigital	External
	added/updated demonstrative examples		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Alternate Terms, Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Description, Name, Relationships		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Observed Examples, Other Notes, Potential Mitigations, Theoretical Notes, Weakness Ordinalities		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Demonstrative Examples, Detection Factors, Likelihood of Exploit, Potential Mitigations, References, Related Attack Patterns, Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Related Attack Patterns		
Previous Entry Names			
Change Date	**Previous Entry Name**		
2009-10-29	Unchecked Array Indexing		

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