

vul_files_85 Scan Report

Project Name	vul_files_85
Scan Start	Thursday, January 9, 2025 2:03:42 PM
Preset	Checkmarx Default
Scan Time	01h:04m:29s
Lines Of Code Scanned	295033
Files Scanned	47
Report Creation Time	Thursday, January 9, 2025 3:16:04 PM
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079
Team	CxServer
Checkmarx Version	8.7.0
Scan Type	Full
Source Origin	LocalPath
Density	2/1000 (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

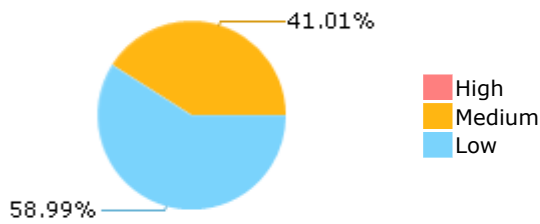
Results Limit

Results limit per query was set to 50

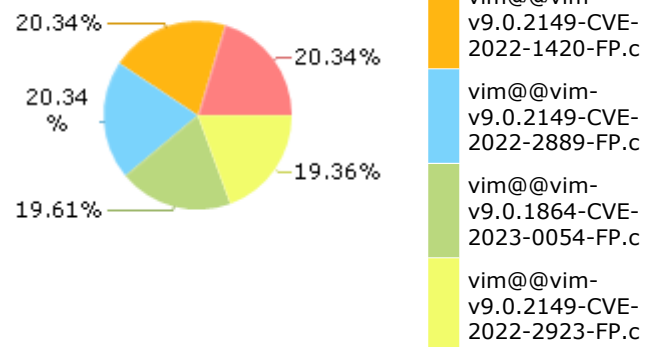
Selected Queries

Selected queries are listed in [Result Summary](#)

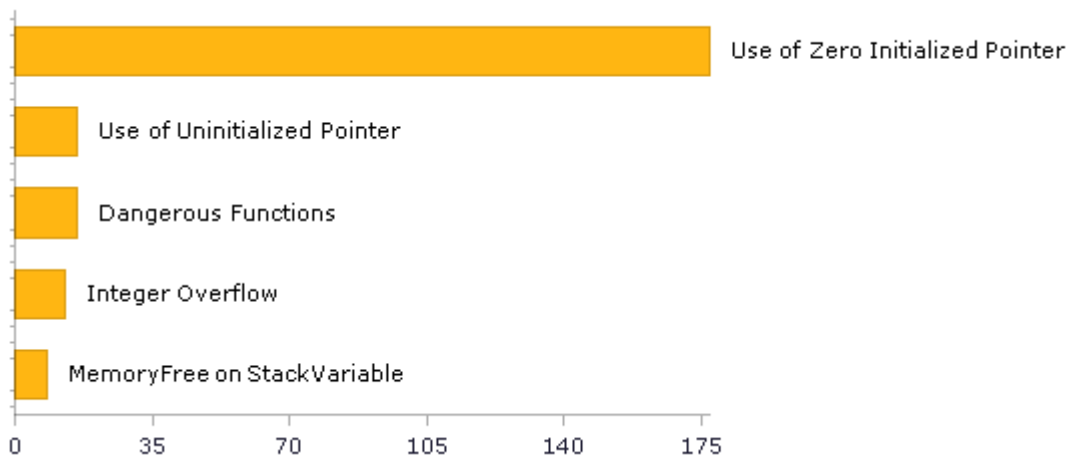
Result Summary



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	210	54
A2-Broken Authentication	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	38	38
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	16	16
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	16	16
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	1	1
PCI DSS (3.2) - 6.5.2 - Buffer overflows	14	14
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.	3	3
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.	3	3
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.	35	35
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.	13	13

* 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)	38	38
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)*	401	123
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*	41	41
SI-11 Error Handling (P2)*	10	10
SI-15 Information Output Filtering (P0)	0	0
SI-16 Memory Protection (P1)	1	1

* 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 wasn't 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

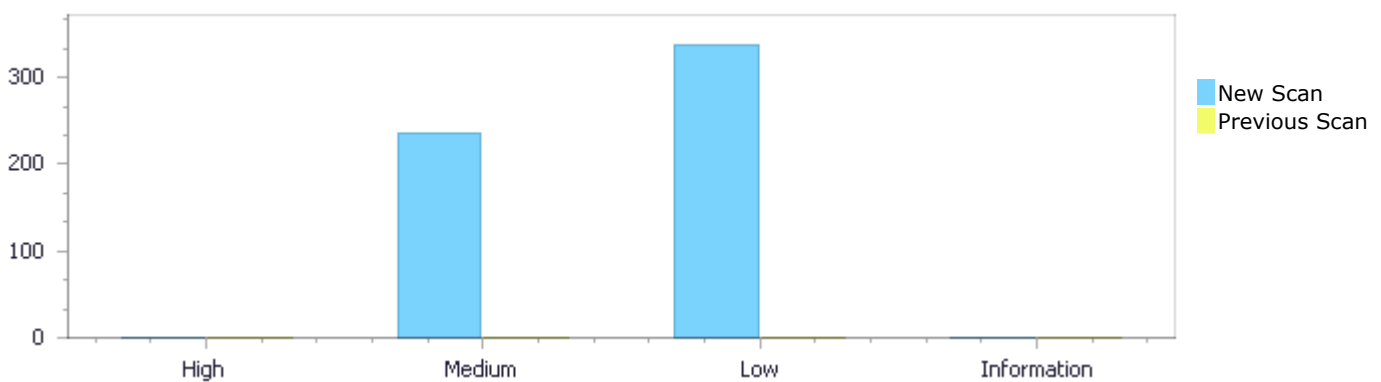
Scan Summary - Custom

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	235	338	0	573
Recurrent Issues	0	0	0	0	0
Total	0	235	338	0	573

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	235	338	0	573
Urgent	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0
Total	0	235	338	0	573

Result Summary

Vulnerability Type	Occurrences	Severity
Use of Zero Initialized Pointer	177	Medium
Dangerous Functions	16	Medium
Use of Uninitialized Pointer	16	Medium
Integer Overflow	13	Medium
MemoryFree on StackVariable	8	Medium

Divide By Zero	4	Medium
Char Overflow	1	Medium
NULL Pointer Dereference	205	Low
Use of Sizeof On a Pointer Type	51	Low
Improper Resource Access Authorization	35	Low
Unchecked Array Index	23	Low
Unchecked Return Value	10	Low
Potential Precision Problem	4	Low
Arithmetic Operation On Boolean	3	Low
Incorrect Permission Assignment For Critical Resources	3	Low
TOCTOU	3	Low
Potential Off by One Error in Loops	1	Low

10 Most Vulnerable Files

High and Medium Vulnerabilities

File Name	Issues Found
vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	43
vim@@vim-v9.0.2149-CVE-2022-1420-FP.c	43
vim@@vim-v9.0.2149-CVE-2022-2889-FP.c	43
vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	42
vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	23
vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	10
vim@@vim-v9.0.2149-CVE-2022-2208-FP.c	7
vim@@vim-v9.0.2149-CVE-2022-2210-FP.c	7
vim@@vim-v9.0.2149-CVE-2022-2289-FP.c	7
vim@@vim-v9.0.2149-CVE-2022-2598-FP.c	7

Scan Results Details

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=1050090&projectid=50079&pathid=356
Status	New

The variable declared in ht at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 1008 is not initialized when it is used by ll_tv at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 1008.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	1026	1181
Object	ht	ll_tv

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method get_lval(

```
....
1026.      hashtable_T      *ht = NULL;
....
1181.      lp->ll_tv = &v->di_tv;
```

Use of Zero Initialized Pointer\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=357
Status	New

The variable declared in ll_tv at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 1008 is not initialized when it is used by ll_tv at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 1008.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-	vim@@vim-v9.0.1864-CVE-2023-0054-

	FP.c	FP.c
Line	1470	1198
Object	ll_tv	ll_tv

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method get_lval(

```
....  
1470.          lp->ll_tv = NULL;  
....  
1198.          vartype_T v_type = lp->ll_tv->v_type;
```

Use of Zero Initialized Pointer\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=358
Status	New

The variable declared in ht_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5399 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5399.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5405	5432
Object	ht_stack	ht_stack

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....  
5405.          ht_stack_T *ht_stack = NULL;  
....  
5432.          ht_stack = ht_stack->prev;
```

Use of Zero Initialized Pointer\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=359
Status	New

The variable declared in ht_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5399 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5399.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5405	5430
Object	ht_stack	ht_stack

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....  
5405.      ht_stack_T    *ht_stack = NULL;  
....  
5430.      cur_ht = ht_stack->ht;
```

Use of Zero Initialized Pointer\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=360
Status	New

The variable declared in list_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5479 is not initialized when it is used by list_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5479.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5484	5503
Object	list_stack	list_stack

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....  
5484.      list_stack_T *list_stack = NULL;  
....  
5503.      list_stack = list_stack->prev;
```

Use of Zero Initialized Pointer\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=361
Status	New

The variable declared in list_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5479 is not initialized when it is used by list_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5479.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5484	5501
Object	list_stack	list_stack

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....  
5484.         list_stack_T *list_stack = NULL;  
....  
5501.         cur_l = list_stack->list;
```

Use of Zero Initialized Pointer\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=362
Status	New

The variable declared in tf at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5863 is not initialized when it is used by r at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5863.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	6100	5903
Object	tf	r

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method echo_string_core(

```
....  
6100.         char_u *tf = NULL;  
....  
5903.         r = *tofree;
```

Use of Zero Initialized Pointer\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=363

Status New

The variable declared in tf at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5863 is not initialized when it is used by r at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5863.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	6100	5930
Object	tf	r

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method echo_string_core(

```
....  
6100.                char_u *tf = NULL;  
....  
5930.                r = *tfree;
```

Use of Zero Initialized Pointer\Path 9:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=364>

Status New

The variable declared in tf at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5863 is not initialized when it is used by r at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5863.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	6100	5986
Object	tf	r

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method echo_string_core(

```
....  
6100.                char_u *tf = NULL;  
....  
5986.                r = *tfree;
```

Use of Zero Initialized Pointer\Path 10:

Severity Medium

Result State To Verify

Online Results <http://WIN->

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=365
Status	New

The variable declared in tf at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5863 is not initialized when it is used by r at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5863.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	6100	5991
Object	tf	r

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method echo_string_core(

```
....  
6100.                char_u *tf = NULL;  
....  
5991.                r = blob2string(tv->vval.v_blob, tofree, numbuf);
```

Use of Zero Initialized Pointer\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=366
Status	New

The variable declared in tf at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5863 is not initialized when it is used by r at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5863.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	6100	6015
Object	tf	r

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method echo_string_core(

```
....  
6100.                char_u *tf = NULL;  
....  
6015.                r = *tofree;
```

Use of Zero Initialized Pointer\Path 12:

Severity	Medium
----------	--------

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=367
Status	New

The variable declared in tf at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5863 is not initialized when it is used by r at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5863.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	6100	6040
Object	tf	r

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method echo_string_core(

```
....  
6100.                char_u *tf = NULL;  
....  
6040.                r = *tfree;
```

Use of Zero Initialized Pointer\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=368
Status	New

The variable declared in Pointer at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5863 is not initialized when it is used by r at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5863.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	6055	6061
Object	Pointer	r

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method echo_string_core(

```
....  
6055.                *tfree = NULL;  
....  
6061.                r = *tfree;
```

Use of Zero Initialized Pointer\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=369
Status	New

The variable declared in tf at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5863 is not initialized when it is used by r at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5863.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	6100	6061
Object	tf	r

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method echo_string_core(

```
....  
6100.                char_u *tf = NULL;  
....  
6061.                r = *tofree;
```

Use of Zero Initialized Pointer\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=370
Status	New

The variable declared in tf at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5863 is not initialized when it is used by r at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5863.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	6100	6075
Object	tf	r

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method echo_string_core(

```
.....
6100.                char_u *tf = NULL;
.....
6075.                r = *tfree = alloc(len);
```

Use of Zero Initialized Pointer\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=371
Status	New

The variable declared in ht at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 1157 is not initialized when it is used by ll_tv at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 1157.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	1175	1357
Object	ht	ll_tv

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method get_lval(

```
.....
1175.        hashtab_T    *ht = NULL;
.....
1357.        lp->ll_tv = &v->di_tv;
```

Use of Zero Initialized Pointer\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=372
Status	New

The variable declared in ll_tv at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 1157 is not initialized when it is used by ll_tv at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 1157.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	1666	1374
Object	ll_tv	ll_tv

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method get_lval(

```
....
1666.          lp->ll_tv = NULL;
....
1374.          vartype_T v_type = lp->ll_tv->v_type;
```

Use of Zero Initialized Pointer\Path 18:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=373>
Status New

The variable declared in ht_stack at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5604 is not initialized when it is used by ht_stack at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5604.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	5610	5637
Object	ht_stack	ht_stack

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
5610.          ht_stack_T *ht_stack = NULL;
....
5637.          ht_stack = ht_stack->prev;
```

Use of Zero Initialized Pointer\Path 19:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=374>
Status New

The variable declared in ht_stack at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5604 is not initialized when it is used by ht_stack at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5604.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	5610	5635
Object	ht_stack	ht_stack

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....  
5610.      ht_stack_T      *ht_stack = NULL;  
....  
5635.      cur_ht = ht_stack->ht;
```

Use of Zero Initialized Pointer\Path 20:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=375>
Status New

The variable declared in list_stack at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5684 is not initialized when it is used by list_stack at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5684.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	5689	5708
Object	list_stack	list_stack

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....  
5689.      list_stack_T *list_stack = NULL;  
....  
5708.      list_stack = list_stack->prev;
```

Use of Zero Initialized Pointer\Path 21:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=376>
Status New

The variable declared in list_stack at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5684 is not initialized when it is used by list_stack at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5684.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c

Line	5689	5706
Object	list_stack	list_stack

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
5689.      list_stack_T *list_stack = NULL;
....
5706.      cur_l = list_stack->list;
```

Use of Zero Initialized Pointer\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=377
Status	New

The variable declared in tf at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 6083 is not initialized when it is used by r at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 6083.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	6320	6123
Object	tf	r

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method echo_string_core(

```
....
6320.      char_u *tf = NULL;
....
6123.      r = *tfree;
```

Use of Zero Initialized Pointer\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=378
Status	New

The variable declared in tf at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 6083 is not initialized when it is used by r at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 6083.

Source	Destination
--------	-------------

File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	6320	6150
Object	tf	r

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method echo_string_core(

```
....  
6320.                char_u *tf = NULL;  
....  
6150.                r = *tfree;
```

Use of Zero Initialized Pointer\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=379
Status	New

The variable declared in tf at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 6083 is not initialized when it is used by r at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 6083.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	6320	6206
Object	tf	r

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method echo_string_core(

```
....  
6320.                char_u *tf = NULL;  
....  
6206.                r = *tfree;
```

Use of Zero Initialized Pointer\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=380
Status	New

The variable declared in tf at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 6083 is not initialized when it is used by r at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 6083.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	6320	6211
Object	tf	r

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method echo_string_core(

```
....
6320.                char_u *tf = NULL;
....
6211.                r = blob2string(tv->vval.v_blob, tofree, numbuf);
```

Use of Zero Initialized Pointer\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=381
Status	New

The variable declared in tf at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 6083 is not initialized when it is used by r at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 6083.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	6320	6235
Object	tf	r

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method echo_string_core(

```
....
6320.                char_u *tf = NULL;
....
6235.                r = *tofree;
```

Use of Zero Initialized Pointer\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=382
Status	New

The variable declared in tf at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 6083 is not initialized when it is used by r at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 6083.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	6320	6260
Object	tf	r

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c

Method echo_string_core(

```
....  
6320.                char_u *tf = NULL;  
....  
6260.                r = *tofree;
```

Use of Zero Initialized Pointer\Path 28:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=383>

Status New

The variable declared in Pointer at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 6083 is not initialized when it is used by r at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 6083.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	6275	6281
Object	Pointer	r

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c

Method echo_string_core(

```
....  
6275.                *tofree = NULL;  
....  
6281.                r = *tofree;
```

Use of Zero Initialized Pointer\Path 29:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=384>

Status New

The variable declared in tf at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 6083 is not initialized when it is used by r at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 6083.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	6320	6281
Object	tf	r

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c

Method echo_string_core(

```
....  
6320.                char_u *tf = NULL;  
....  
6281.                r = *tfree;
```

Use of Zero Initialized Pointer\Path 30:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=385>

Status New

The variable declared in tf at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 6083 is not initialized when it is used by r at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 6083.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	6320	6295
Object	tf	r

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c

Method echo_string_core(

```
....  
6320.                char_u *tf = NULL;  
....  
6295.                r = *tfree = alloc(len);
```

Use of Zero Initialized Pointer\Path 31:

Severity Medium

Result State To Verify

Online Results <http://WIN->

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=386
Status	New

The variable declared in tf at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 6083 is not initialized when it is used by r at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 6083.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	6320	6348
Object	tf	r

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method echo_string_core(

```
....  
6320.                char_u *tf = NULL;  
....  
6348.                r = *tofree;
```

Use of Zero Initialized Pointer\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=387
Status	New

The variable declared in type at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 66 is not initialized when it is used by ga_data at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 66.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	146	179
Object	type	ga_data

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method one_function_arg(

```
....  
146.                char_u *type = NULL;  
....  
179.                ((char_u **)argtypes->ga_data)[argtypes->ga_len++] =  
type;
```

Use of Zero Initialized Pointer\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=388
Status	New

The variable declared in cmdline at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 1305 is not initialized when it is used by ga_data at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 1305.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	1322	1409
Object	cmdline	ga_data

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method lambda_function_body(

```
....  
1322.      char_u *cmdline = NULL;  
....  
1409.      ((char_u **) (tfgap->ga_data)) [tfgap->ga_len++] =  
cmdline;
```

Use of Zero Initialized Pointer\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=389
Status	New

The variable declared in cmdline at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 1305 is not initialized when it is used by ga_data at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 1305.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	1322	1396
Object	cmdline	ga_data

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method lambda_function_body(

```
.....
1322.      char_u *cmdline = NULL;
.....
1396.      ((char_u **)freegap->ga_data)[freegap->ga_len++] = pnl;
```

Use of Zero Initialized Pointer\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=390
Status	New

The variable declared in cmdline at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 1305 is not initialized when it is used by ga_data at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 1305.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	1322	1395
Object	cmdline	ga_data

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method lambda_function_body(

```
.....
1322.      char_u *cmdline = NULL;
.....
1395.      ((char_u **)gap->ga_data)[gap->ga_len++] = pnl;
```

Use of Zero Initialized Pointer\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=391
Status	New

The variable declared in cmdline at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 1305 is not initialized when it is used by ga_data at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 856.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	1322	1274
Object	cmdline	ga_data

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method lambda_function_body(

```
....
1322.      char_u *cmdline = NULL;
```

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method get_function_body(

```
....
1274.      ((char_u **) (newlines->ga_data)) [newlines->ga_len++] = p;
```

Use of Zero Initialized Pointer\Path 37:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=392>
Status New

The variable declared in cmdline at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 1305 is not initialized when it is used by heredoc_trimmed at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 856.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	1322	1233
Object	cmdline	heredoc_trimmed

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method lambda_function_body(

```
....
1322.      char_u *cmdline = NULL;
```

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method get_function_body(

```
....
1233.      heredoc_trimmed = vim_strnsave(theline,
```

Use of Zero Initialized Pointer\Path 38:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=392>

Status	079&pathid=393 New
--------	---

The variable declared in cmdline at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 1305 is not initialized when it is used by heredoc_trimmed at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 856.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	1322	1192
Object	cmdline	heredoc_trimmed

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method lambda_function_body(

```
....
1322.      char_u *cmdline = NULL;
```

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method get_function_body(

```
....
1192.      heredoc_trimmed = vim_strnsave(theline,
```

Use of Zero Initialized Pointer\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=394
Status	New

The variable declared in cmdline at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 1305 is not initialized when it is used by name at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 4182.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	1322	4344
Object	cmdline	name

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method lambda_function_body(

```
.....
1322.      char_u *cmdline = NULL;
```

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c

Method trans_function_name_ext(

```
.....
4344.      name = deref_func_name(*pp, &len, partial, type,
```

Use of Zero Initialized Pointer\Path 40:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=395>

Status New

The variable declared in cmdline at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 1305 is not initialized when it is used by nextcmd at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 856.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	1322	1023
Object	cmdline	nextcmd

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c

Method lambda_function_body(

```
.....
1322.      char_u *cmdline = NULL;
```

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c

Method get_function_body(

```
.....
1023.      nextcmd = p + 1;
```

Use of Zero Initialized Pointer\Path 41:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=396>

Status New

The variable declared in pt at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 1498 is not initialized when it is used by pt at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 1498.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	1513	1637
Object	pt	pt

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c

Method get_lambda_tv(

```
....
1513.      partial_T    *pt = NULL;
....
1637.      pt = ALLOC_CLEAR_ONE(partial_T);
```

Use of Zero Initialized Pointer\Path 42:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=397>

Status New

The variable declared in tofree at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 6528 is not initialized when it is used by s at vim@@vim-v9.0.2149-CVE-2021-4187-FP.c in line 6528.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	6531	6535
Object	tofree	s

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c

Method get_return_cmd(void *rettv)

```
....
6531.      char_u *tofree = NULL;
....
6535.      s = echo_string((typval_T *)rettv, &tofree, numbuf, 0);
```

Use of Zero Initialized Pointer\Path 43:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=397>

Status	079&pathid=398 New
--------	---

The variable declared in ht at vim@@vim-v9.0.2149-CVE-2022-1420-FP.c in line 1157 is not initialized when it is used by ll_tv at vim@@vim-v9.0.2149-CVE-2022-1420-FP.c in line 1157.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Line	1175	1357
Object	ht	ll_tv

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Method get_lval(

```
....
1175.      hashtable_T      *ht = NULL;
....
1357.      lp->ll_tv = &v->di_tv;
```

Use of Zero Initialized Pointer\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=399
Status	New

The variable declared in ll_tv at vim@@vim-v9.0.2149-CVE-2022-1420-FP.c in line 1157 is not initialized when it is used by ll_tv at vim@@vim-v9.0.2149-CVE-2022-1420-FP.c in line 1157.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Line	1666	1374
Object	ll_tv	ll_tv

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Method get_lval(

```
....
1666.      lp->ll_tv = NULL;
....
1374.      vartype_T v_type = lp->ll_tv->v_type;
```

Use of Zero Initialized Pointer\Path 45:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=400
Status	New

The variable declared in ht_stack at vim@@vim-v9.0.2149-CVE-2022-1420-FP.c in line 5604 is not initialized when it is used by ht_stack at vim@@vim-v9.0.2149-CVE-2022-1420-FP.c in line 5604.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Line	5610	5637
Object	ht_stack	ht_stack

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....  
5610.      ht_stack_T      *ht_stack = NULL;  
....  
5637.      ht_stack = ht_stack->prev;
```

Use of Zero Initialized Pointer\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=401
Status	New

The variable declared in ht_stack at vim@@vim-v9.0.2149-CVE-2022-1420-FP.c in line 5604 is not initialized when it is used by ht_stack at vim@@vim-v9.0.2149-CVE-2022-1420-FP.c in line 5604.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Line	5610	5635
Object	ht_stack	ht_stack

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....  
5610.      ht_stack_T      *ht_stack = NULL;  
....  
5635.      cur_ht = ht_stack->ht;
```

Use of Zero Initialized Pointer\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=402
Status	New

The variable declared in list_stack at vim@@vim-v9.0.2149-CVE-2022-1420-FP.c in line 5684 is not initialized when it is used by list_stack at vim@@vim-v9.0.2149-CVE-2022-1420-FP.c in line 5684.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Line	5689	5708
Object	list_stack	list_stack

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....  
5689.      list_stack_T *list_stack = NULL;  
....  
5708.      list_stack = list_stack->prev;
```

Use of Zero Initialized Pointer\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=403
Status	New

The variable declared in list_stack at vim@@vim-v9.0.2149-CVE-2022-1420-FP.c in line 5684 is not initialized when it is used by list_stack at vim@@vim-v9.0.2149-CVE-2022-1420-FP.c in line 5684.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Line	5689	5706
Object	list_stack	list_stack

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....  
5689.      list_stack_T *list_stack = NULL;  
....  
5706.      cur_l = list_stack->list;
```

Use of Zero Initialized Pointer\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=404
Status	New

The variable declared in tf at vim@@vim-v9.0.2149-CVE-2022-1420-FP.c in line 6083 is not initialized when it is used by r at vim@@vim-v9.0.2149-CVE-2022-1420-FP.c in line 6083.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Line	6320	6123
Object	tf	r

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Method echo_string_core(

```
....  
6320.                char_u *tf = NULL;  
....  
6123.                r = *tfree;
```

Use of Zero Initialized Pointer\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=405
Status	New

The variable declared in tf at vim@@vim-v9.0.2149-CVE-2022-1420-FP.c in line 6083 is not initialized when it is used by r at vim@@vim-v9.0.2149-CVE-2022-1420-FP.c in line 6083.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Line	6320	6150
Object	tf	r

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Method echo_string_core(


```
.....
6320.                char_u *tf = NULL;
.....
6150.                r = *tfree;
```

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=1050090&projectid=50079&pathid=27
Status	New

The dangerous function, sprintf, was found in use at line 690 in vim@@vim-v9.0.2149-CVE-2021-4187-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	695	695
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method get_lambda_name(void)

```
.....
695.                sprintf((char*)name, "<lambda>%d", ++lambda_no);
```

Dangerous Functions\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=28
Status	New

The dangerous function, sprintf, was found in use at line 2046 in vim@@vim-v9.0.2149-CVE-2021-4187-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

Source	Destination
--------	-------------

File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	2070	2070
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method fname_trans_sid(

```
....  
2070.          sprintf((char *)fname_buf + 3, "%ld_",
```

Dangerous Functions\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=29
Status	New

The dangerous function, sprintf, was found in use at line 2788 in vim@@vim-v9.0.2149-CVE-2021-4187-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	2975	2975
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method call_user_func(

```
....  
2975.          sprintf((char *)numbuf, "%d", ai + 1);
```

Dangerous Functions\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=30
Status	New

The dangerous function, sprintf, was found in use at line 3348 in vim@@vim-v9.0.2149-CVE-2021-4187-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

Source	Destination
--------	-------------

File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	3359	3359
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method delete_script_functions(int sid)

```
....  
3359.          sprintf((char *)buf + 3, "%d_", sid);
```

Dangerous Functions\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=31
Status	New

The dangerous function, sprintf, was found in use at line 4182 in vim@@vim-v9.0.2149-CVE-2021-4187-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	4441	4441
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method trans_function_name_ext(

```
....  
4441.          sprintf((char *)sid_buf, "%ld_",  
(long)current_sctx.sc_sid);
```

Dangerous Functions\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=32
Status	New

The dangerous function, sprintf, was found in use at line 4666 in vim@@vim-v9.0.2149-CVE-2021-4187-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

Source	Destination
--------	-------------

File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	5137	5137
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method define_function(

```
.....  
5137.          sprintf(numbuf, "%d", ++func_nr);
```

Dangerous Functions\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=33
Status	New

The dangerous function, sprintf, was found in use at line 723 in vim@@vim-v9.0.2149-CVE-2022-2816-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2816-FP.c	vim@@vim-v9.0.2149-CVE-2022-2816-FP.c
Line	764	764
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2816-FP.c
Method ex_compiler(exarg_T *eap)

```
.....  
764.          sprintf((char *)buf, "compiler/%s.vim", eap->arg);
```

Dangerous Functions\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=34
Status	New

The dangerous function, sprintf, was found in use at line 2186 in vim@@vim-v9.0.2149-CVE-2022-2923-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

Source	Destination
--------	-------------

File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	2746	2746
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
....  
2746.                                sprintf((char *)buf, "^%s", items[4]);
```

Dangerous Functions\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=35
Status	New

The dangerous function, sprintf, was found in use at line 2186 in vim@@vim-v9.0.2149-CVE-2022-2923-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	2748	2748
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
....  
2748.                                sprintf((char *)buf, "%s$", items[4]);
```

Dangerous Functions\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=36
Status	New

The dangerous function, sprintf, was found in use at line 2186 in vim@@vim-v9.0.2149-CVE-2022-2923-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

Source	Destination
--------	-------------

File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	2807	2807
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
....
2807.                                sprintf((char *)buf, "%s",
```

Dangerous Functions\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=37
Status	New

The dangerous function, atoi, was found in use at line 2186 in vim@@vim-v9.0.2149-CVE-2022-2923-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	2473	2473
Object	atoi	atoi

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
....
2473.                                if (atoi((char *)items[1]) == 0)
```

Dangerous Functions\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=38
Status	New

The dangerous function, atoi, was found in use at line 2186 in vim@@vim-v9.0.2149-CVE-2022-2923-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

Source	Destination
--------	-------------

File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	2503	2503
Object	atoi	atoi

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
....  
2503.                compmax = atoi((char *)items[1]);
```

Dangerous Functions\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=39
Status	New

The dangerous function, atoi, was found in use at line 2186 in vim@@vim-v9.0.2149-CVE-2022-2923-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	2511	2511
Object	atoi	atoi

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
....  
2511.                compminlen = atoi((char *)items[1]);
```

Dangerous Functions\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=40
Status	New

The dangerous function, atoi, was found in use at line 2186 in vim@@vim-v9.0.2149-CVE-2022-2923-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

Source	Destination
--------	-------------

File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	2519	2519
Object	atoi	atoi

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
....  
2519.                compsymmax = atoi((char *)items[1]);
```

Dangerous Functions\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=41
Status	New

The dangerous function, atoi, was found in use at line 2186 in vim@@vim-v9.0.2149-CVE-2022-2923-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	2542	2542
Object	atoi	atoi

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
....  
2542.                if (atoi((char *)items[1]) == 0)
```

Dangerous Functions\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=42
Status	New

The dangerous function, atoi, was found in use at line 2186 in vim@@vim-v9.0.2149-CVE-2022-2923-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

Source	Destination
--------	-------------

File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	2688	2688
Object	atoi	atoi

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
....
2688.          aff_todo = atoi((char *)items[3]);
```

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=1050090&projectid=50079&pathid=340
Status	New

The variable declared in wp at vim@@vim-v9.0.2149-CVE-2022-2208-FP.c in line 137 is not initialized when it is used by w_p_diff at vim@@vim-v9.0.2149-CVE-2022-2208-FP.c in line 137.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Line	139	147
Object	wp	w_p_diff

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Method diff_buf_adjust(win_T *win)

```
....
139.      win_T      *wp;
....
147.      if (wp->w_buffer == win->w_buffer && wp->w_p_diff)
```

Use of Uninitialized Pointer\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=340

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=341
Status	New

The variable declared in wp at vim@@vim-v9.0.2149-CVE-2022-2208-FP.c in line 137 is not initialized when it is used by w_buffer at vim@@vim-v9.0.2149-CVE-2022-2208-FP.c in line 137.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Line	139	147
Object	wp	w_buffer

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Method diff_buf_adjust(win_T *win)

```
....
139.      win_T      *wp;
....
147.          if (wp->w_buffer == win->w_buffer && wp->w_p_diff)
```

Use of Uninitialized Pointer\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=342
Status	New

The variable declared in wp at vim@@vim-v9.0.2149-CVE-2022-2208-FP.c in line 137 is not initialized when it is used by wp at vim@@vim-v9.0.2149-CVE-2022-2208-FP.c in line 137.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Line	139	149
Object	wp	wp

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Method diff_buf_adjust(win_T *win)

```
....
139.      win_T      *wp;
....
149.          if (wp == NULL)
```

Use of Uninitialized Pointer\Path 4:

Severity	Medium
----------	--------

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=343
Status	New

The variable declared in tp at vim@@vim-v9.0.2149-CVE-2022-2208-FP.c in line 2256 is not initialized when it is used by tp_diff_invalid at vim@@vim-v9.0.2149-CVE-2022-2208-FP.c in line 2256.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Line	2264	2389
Object	tp	tp_diff_invalid

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Method diffopt_changed(void)

```
....  
2264.         tabpage_T      *tp;  
....  
2389.         tp->tp_diff_invalid = TRUE;
```

Use of Uninitialized Pointer\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=344
Status	New

The variable declared in wp at vim@@vim-v9.0.2149-CVE-2022-2210-FP.c in line 137 is not initialized when it is used by w_p_diff at vim@@vim-v9.0.2149-CVE-2022-2210-FP.c in line 137.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c
Line	139	147
Object	wp	w_p_diff

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2210-FP.c
Method diff_buf_adjust(win_T *win)

```
....  
139.         win_T      *wp;  
....  
147.         if (wp->w_buffer == win->w_buffer && wp->w_p_diff)
```

Use of Uninitialized Pointer\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=345
Status	New

The variable declared in wp at vim@@vim-v9.0.2149-CVE-2022-2210-FP.c in line 137 is not initialized when it is used by w_buffer at vim@@vim-v9.0.2149-CVE-2022-2210-FP.c in line 137.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c
Line	139	147
Object	wp	w_buffer

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2210-FP.c
Method diff_buf_adjust(win_T *win)

```
....  
139.      win_T      *wp;  
....  
147.      if (wp->w_buffer == win->w_buffer && wp->w_p_diff)
```

Use of Uninitialized Pointer\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=346
Status	New

The variable declared in wp at vim@@vim-v9.0.2149-CVE-2022-2210-FP.c in line 137 is not initialized when it is used by wp at vim@@vim-v9.0.2149-CVE-2022-2210-FP.c in line 137.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c
Line	139	149
Object	wp	wp

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2210-FP.c
Method diff_buf_adjust(win_T *win)

```

.....
139.      win_T      *wp;
.....
149.      if (wp == NULL)

```

Use of Uninitialized Pointer\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=347
Status	New

The variable declared in `tp` at `vim@@vim-v9.0.2149-CVE-2022-2210-FP.c` in line 2256 is not initialized when it is used by `tp_diff_invalid` at `vim@@vim-v9.0.2149-CVE-2022-2210-FP.c` in line 2256.

	Source	Destination
File	<code>vim@@vim-v9.0.2149-CVE-2022-2210-FP.c</code>	<code>vim@@vim-v9.0.2149-CVE-2022-2210-FP.c</code>
Line	2264	2389
Object	<code>tp</code>	<code>tp_diff_invalid</code>

Code Snippet

File Name `vim@@vim-v9.0.2149-CVE-2022-2210-FP.c`
Method `diffopt_changed(void)`

```

.....
2264.      tabpage_T      *tp;
.....
2389.      tp->tp_diff_invalid = TRUE;

```

Use of Uninitialized Pointer\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=348
Status	New

The variable declared in `wp` at `vim@@vim-v9.0.2149-CVE-2022-2289-FP.c` in line 137 is not initialized when it is used by `w_p_diff` at `vim@@vim-v9.0.2149-CVE-2022-2289-FP.c` in line 137.

	Source	Destination
File	<code>vim@@vim-v9.0.2149-CVE-2022-2289-FP.c</code>	<code>vim@@vim-v9.0.2149-CVE-2022-2289-FP.c</code>
Line	139	147
Object	<code>wp</code>	<code>w_p_diff</code>

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Method diff_buf_adjust(win_T *win)

```
....  
139.      win_T      *wp;  
....  
147.      if (wp->w_buffer == win->w_buffer && wp->w_p_diff)
```

Use of Uninitialized Pointer\Path 10:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=349>
Status New

The variable declared in wp at vim@@vim-v9.0.2149-CVE-2022-2289-FP.c in line 137 is not initialized when it is used by w_buffer at vim@@vim-v9.0.2149-CVE-2022-2289-FP.c in line 137.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Line	139	147
Object	wp	w_buffer

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Method diff_buf_adjust(win_T *win)

```
....  
139.      win_T      *wp;  
....  
147.      if (wp->w_buffer == win->w_buffer && wp->w_p_diff)
```

Use of Uninitialized Pointer\Path 11:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=350>
Status New

The variable declared in wp at vim@@vim-v9.0.2149-CVE-2022-2289-FP.c in line 137 is not initialized when it is used by wp at vim@@vim-v9.0.2149-CVE-2022-2289-FP.c in line 137.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Line	139	149
Object	wp	wp

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Method diff_buf_adjust(win_T *win)

```
....  
139.         win_T      *wp;  
....  
149.         if (wp == NULL)
```

Use of Uninitialized Pointer\Path 12:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=351>
Status New

The variable declared in tp at vim@@vim-v9.0.2149-CVE-2022-2289-FP.c in line 2256 is not initialized when it is used by tp_diff_invalid at vim@@vim-v9.0.2149-CVE-2022-2289-FP.c in line 2256.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Line	2264	2389
Object	tp	tp_diff_invalid

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Method diffopt_changed(void)

```
....  
2264.         tabpage_T      *tp;  
....  
2389.         tp->tp_diff_invalid = TRUE;
```

Use of Uninitialized Pointer\Path 13:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=352>
Status New

The variable declared in wp at vim@@vim-v9.0.2149-CVE-2022-2598-FP.c in line 137 is not initialized when it is used by w_p_diff at vim@@vim-v9.0.2149-CVE-2022-2598-FP.c in line 137.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c

Line	139	147
Object	wp	w_p_diff

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2598-FP.c

Method diff_buf_adjust(win_T *win)

```

.....
139.         win_T      *wp;
.....
147.         if (wp->w_buffer == win->w_buffer && wp->w_p_diff)

```

Use of Uninitialized Pointer\Path 14:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=353>

Status New

The variable declared in wp at vim@@vim-v9.0.2149-CVE-2022-2598-FP.c in line 137 is not initialized when it is used by w_buffer at vim@@vim-v9.0.2149-CVE-2022-2598-FP.c in line 137.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c
Line	139	147
Object	wp	w_buffer

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2598-FP.c

Method diff_buf_adjust(win_T *win)

```

.....
139.         win_T      *wp;
.....
147.         if (wp->w_buffer == win->w_buffer && wp->w_p_diff)

```

Use of Uninitialized Pointer\Path 15:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=354>

Status New

The variable declared in wp at vim@@vim-v9.0.2149-CVE-2022-2598-FP.c in line 137 is not initialized when it is used by wp at vim@@vim-v9.0.2149-CVE-2022-2598-FP.c in line 137.

Source	Destination
--------	-------------

File	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c
Line	139	149
Object	wp	wp

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2598-FP.c
Method diff_buf_adjust(win_T *win)

```
....
139.      win_T      *wp;
....
149.      if (wp == NULL)
```

Use of Uninitialized Pointer\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=355
Status	New

The variable declared in tp at vim@@vim-v9.0.2149-CVE-2022-2598-FP.c in line 2256 is not initialized when it is used by tp at vim@@vim-v9.0.2149-CVE-2022-2598-FP.c in line 2256.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c
Line	2264	2389
Object	tp	tp

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2598-FP.c
Method diffopt_changed(void)

```
....
2264.      tabpage_T      *tp;
....
2389.      tp->tp_diff_invalid = TRUE;
```

Integer Overflow

Query Path:

CPP\Cx\CPP Integer Overflow\Integer Overflow Version:0

Categories

PCI DSS v3.2: PCI DSS (3.2) - 6.5.2 - Buffer overflows
FISMA 2014: System And Information Integrity
NIST SP 800-53: SI-10 Information Input Validation (P1)

Description

Integer Overflow\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=14
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 291 of vim@@vim-v9.0.2149-CVE-2022-2208-FP.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Line	322	322
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Method diff_mark_adjust_tp(

```
....  
322.          inserted = amount;
```

Integer Overflow\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=15
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 291 of vim@@vim-v9.0.2149-CVE-2022-2208-FP.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Line	328	328
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Method diff_mark_adjust_tp(

```
....  
328.          inserted = amount_after;
```

Integer Overflow\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=16
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 291 of vim@@vim-v9.0.2149-CVE-2022-2208-FP.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Line	335	335
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Method diff_mark_adjust_tp(

```
....  
335.         deleted = -amount_after;
```

Integer Overflow\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=17
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 291 of vim@@vim-v9.0.2149-CVE-2022-2210-FP.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c
Line	322	322
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2210-FP.c
Method diff_mark_adjust_tp(

```
....  
322.         inserted = amount;
```

Integer Overflow\Path 5:

Severity	Medium
----------	--------

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=18
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 291 of vim@@vim-v9.0.2149-CVE-2022-2210-FP.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c
Line	328	328
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2210-FP.c
Method diff_mark_adjust_tp(

```
....  
328.          inserted = amount_after;
```

Integer Overflow\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=19
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 291 of vim@@vim-v9.0.2149-CVE-2022-2210-FP.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c
Line	335	335
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2210-FP.c
Method diff_mark_adjust_tp(

```
....  
335.          deleted = -amount_after;
```

Integer Overflow\Path 7:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=20
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 291 of vim@@vim-v9.0.2149-CVE-2022-2289-FP.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Line	322	322
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Method diff_mark_adjust_tp(

```
....  
322.         inserted = amount;
```

Integer Overflow\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=21
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 291 of vim@@vim-v9.0.2149-CVE-2022-2289-FP.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Line	328	328
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Method diff_mark_adjust_tp(

```
....  
328.         inserted = amount_after;
```

Integer Overflow\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=22

[PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=22](http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=22)

Status New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 291 of vim@@vim-v9.0.2149-CVE-2022-2289-FP.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Line	335	335
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Method diff_mark_adjust_tp(

```
....  
335.         deleted = -amount_after;
```

Integer Overflow\Path 10:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=23>
Status New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 291 of vim@@vim-v9.0.2149-CVE-2022-2598-FP.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c
Line	322	322
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2598-FP.c
Method diff_mark_adjust_tp(

```
....  
322.         inserted = amount;
```

Integer Overflow\Path 11:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=23>

[079&pathid=24](#)

Status New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 291 of vim@@vim-v9.0.2149-CVE-2022-2598-FP.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c
Line	328	328
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2598-FP.c

Method diff_mark_adjust_tp(

```
....  
328.         inserted = amount_after;
```

Integer Overflow\Path 12:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=25>

Status New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 291 of vim@@vim-v9.0.2149-CVE-2022-2598-FP.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c
Line	335	335
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2598-FP.c

Method diff_mark_adjust_tp(

```
....  
335.         deleted = -amount_after;
```

Integer Overflow\Path 13:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=26>

Status	New
--------	-----

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 3091 of vim@@vim-v9.0.2149-CVE-2022-2845-FP.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2845-FP.c	vim@@vim-v9.0.2149-CVE-2022-2845-FP.c
Line	3096	3096
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2845-FP.c
Method replace_join(

```
....
3096.      for (i = replace_stack_nr; --i >= 0; )
```

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=1050090&projectid=50079&pathid=5
Status	New

Calling free() (line 5399) on a variable that was not dynamically allocated (line 5399) in file vim@@vim-v9.0.1864-CVE-2023-0054-FP.c may result with a crash.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5433	5433
Object	tempitem	tempitem

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
5433.      free(tempitem);
```

MemoryFree on StackVariable\Path 2:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=6
Status	New

Calling free() (line 5479) on a variable that was not dynamically allocated (line 5479) in file vim@@vim-v9.0.1864-CVE-2023-0054-FP.c may result with a crash.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5504	5504
Object	tempitem	tempitem

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....  
5504.      free(tempitem);
```

MemoryFree on StackVariable\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=7
Status	New

Calling free() (line 5604) on a variable that was not dynamically allocated (line 5604) in file vim@@vim-v9.0.2149-CVE-2021-4136-FP.c may result with a crash.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	5638	5638
Object	tempitem	tempitem

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....  
5638.      free(tempitem);
```

MemoryFree on StackVariable\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=7

[PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=8](http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=8)

Status New

Calling free() (line 5684) on a variable that was not dynamically allocated (line 5684) in file vim@@vim-v9.0.2149-CVE-2021-4136-FP.c may result with a crash.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	5709	5709
Object	tempitem	tempitem

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c

Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....  
5709.      free(tempitem);
```

MemoryFree on StackVariable\Path 5:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=9>

Status New

Calling free() (line 5604) on a variable that was not dynamically allocated (line 5604) in file vim@@vim-v9.0.2149-CVE-2022-1420-FP.c may result with a crash.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Line	5638	5638
Object	tempitem	tempitem

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-1420-FP.c

Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....  
5638.      free(tempitem);
```

MemoryFree on StackVariable\Path 6:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=9>

[079&pathid=10](#)

Status New

Calling free() (line 5684) on a variable that was not dynamically allocated (line 5684) in file vim@@vim-v9.0.2149-CVE-2022-1420-FP.c may result with a crash.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Line	5709	5709
Object	tempitem	tempitem

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-1420-FP.c

Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....  
5709.      free(tempitem);
```

MemoryFree on StackVariable\Path 7:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=11>

Status New

Calling free() (line 5604) on a variable that was not dynamically allocated (line 5604) in file vim@@vim-v9.0.2149-CVE-2022-2889-FP.c may result with a crash.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2889-FP.c	vim@@vim-v9.0.2149-CVE-2022-2889-FP.c
Line	5638	5638
Object	tempitem	tempitem

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2889-FP.c

Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....  
5638.      free(tempitem);
```

MemoryFree on StackVariable\Path 8:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=12>

Status New

Calling free() (line 5684) on a variable that was not dynamically allocated (line 5684) in file vim@@vim-v9.0.2149-CVE-2022-2889-FP.c may result with a crash.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2889-FP.c	vim@@vim-v9.0.2149-CVE-2022-2889-FP.c
Line	5709	5709
Object	tempitem	tempitem

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2889-FP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
5709.      free(tempitem);
```

Divide By Zero

Query Path:

CPP\Cx\CPP Medium Threat\Divide By Zero Version:1

Description

Divide By Zero\Path 1:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=1>
Status New

The application performs an illegal operation in num_divide, in vim@@vim-v9.0.1864-CVE-2023-0054-FP.c. In line 50, the program attempts to divide by n2, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input n2 in num_divide of vim@@vim-v9.0.1864-CVE-2023-0054-FP.c, at line 50.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	76	76
Object	n2	n2

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method num_divide(varnumber_T n1, varnumber_T n2, int *failed)

```
....
76.      result = n1 / n2;
```

Divide By Zero\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=2
Status	New

The application performs an illegal operation in num_divide, in vim@@vim-v9.0.2149-CVE-2021-4136-FP.c. In line 50, the program attempts to divide by n2, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input n2 in num_divide of vim@@vim-v9.0.2149-CVE-2021-4136-FP.c, at line 50.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	76	76
Object	n2	n2

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method num_divide(varnumber_T n1, varnumber_T n2, int *failed)

```
....  
76.    result = n1 / n2;
```

Divide By Zero\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=3
Status	New

The application performs an illegal operation in num_divide, in vim@@vim-v9.0.2149-CVE-2022-1420-FP.c. In line 50, the program attempts to divide by n2, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input n2 in num_divide of vim@@vim-v9.0.2149-CVE-2022-1420-FP.c, at line 50.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Line	76	76
Object	n2	n2

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Method num_divide(varnumber_T n1, varnumber_T n2, int *failed)

```
....
76.    result = n1 / n2;
```

Divide By Zero\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=4
Status	New

The application performs an illegal operation in num_divide, in vim@@vim-v9.0.2149-CVE-2022-2889-FP.c. In line 50, the program attempts to divide by n2, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input n2 in num_divide of vim@@vim-v9.0.2149-CVE-2022-2889-FP.c, at line 50.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2889-FP.c	vim@@vim-v9.0.2149-CVE-2022-2889-FP.c
Line	76	76
Object	n2	n2

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2889-FP.c
Method num_divide(varnumber_T n1, varnumber_T n2, int *failed)

```
....
76.    result = n1 / n2;
```

Char Overflow

Query Path:

CPP\Cx\CPP Integer Overflow\Char Overflow Version:1

Categories

PCI DSS v3.2: PCI DSS (3.2) - 6.5.2 - Buffer overflows
NIST SP 800-53: SI-10 Information Input Validation (P1)

Description

Char Overflow\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=13
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 856 of vim@@vim-v9.0.2149-CVE-2021-4187-FP.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

Source	Destination
--------	-------------

File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	895	895
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method get_function_body(

```
....
895.      nesting_def[0] = vim9_function;
```

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=1050090&projectid=50079&pathid=105
Status	New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 260 is not initialized when it is used by skip at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 194.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	331	202
Object	null	skip

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method eval_expr_typval(

```
....
331.      if (eval1_emsg(&s, rettv, NULL) == FAIL)
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method eval1_emsg(char_u **arg, typval_T *rettv, exarg_T *eap)

```
....
202.      fill_evalarg_from_eap(&evalarg, eap, eap != NULL && eap-
>skip);
```

NULL Pointer Dereference\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=106
Status	New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 487 is not initialized when it is used by Pointer at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 487.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	544	544
Object	null	Pointer

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method skip_expr_concatenate(

```
....
544.      * ((char_u **) gap->ga_data) = NULL;
```

NULL Pointer Dereference\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=107
Status	New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 639 is not initialized when it is used by skip at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 610.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	644	621
Object	null	skip

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method eval_to_string(


```
....
644.         return eval_to_string_eap(arg, convert, NULL,
use_simple_function);
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method eval_to_string_eap(

```
....
621.         fill_evalarg_from_eap(&evalarg, eap, eap != NULL && eap-
>skip);
```

NULL Pointer Dereference\Path 4:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=108>

Status New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5531.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5211	5550
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method garbage_collect(int testing)

```
....
5211.                                     NULL, NULL);
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method set_ref_in_item_dict(

```
....
5550.         newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 5:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=108>

[079&pathid=109](#)

Status New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5531.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5216	5550
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method garbage_collect(int testing)

```
....
5216.                                     NULL, NULL);
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method set_ref_in_item_dict(

```
....
5550.     newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 6:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=110>

Status New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5531.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5221	5550
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method garbage_collect(int testing)

```
.....
5221.                &aucmd_win[i].auc_win->w_winvar.di_tv, copyID,
NULL, NULL);
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_dict(

```
.....
5550.                newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 7:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=111>
Status New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5531.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5225	5550
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method garbage_collect(int testing)

```
.....
5225.                NULL, NULL);
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_dict(

```
.....
5550.                newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 8:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=112>

Status New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5531.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5229	5550
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method garbage_collect(int testing)

```
....  
5229.                                     NULL, NULL);
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_dict(
↓

```
....  
5550.          newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=113
Status	New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5531.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5235	5550
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method garbage_collect(int testing)

```
....  
5235.                                     NULL, NULL);
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_dict(

```
....  
5550.          newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 10:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=114>
Status New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5514 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5531.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5523	5550
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_callback(callback_T *cb, int copyID)

```
....  
5523.          return set_ref_in_item(&tv, copyID, NULL, NULL);
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_dict(

```
....  
5550.          newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 11:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=115>
Status New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5514 is not initialized when it is used by prev at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5531.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5523	5550
Object	null	prev

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_callback(callback_T *cb, int copyID)

```
....
5523.         return set_ref_in_item(&tv, copyID, NULL, NULL);
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_dict(

```
....
5550.         newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=116
Status	New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by prev at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5531.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5235	5550
Object	null	prev

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method garbage_collect(int testing)

```
....
5235.                                     NULL, NULL);
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_dict(

```
....
5550.          newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=117
Status	New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by prev at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5531.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5229	5550
Object	null	prev

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method garbage_collect(int testing)

```
....
5229.                                     NULL, NULL) ;
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_dict(

```
....
5550.          newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=118
Status	New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by prev at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5531.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Line	5225	5550
Object	null	prev

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method garbage_collect(int testing)

```
....
5225.                                     NULL, NULL);
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method set_ref_in_item_dict(

```
....
5550.     newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 15:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=119>

Status New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by prev at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5531.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5221	5550
Object	null	prev

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method garbage_collect(int testing)

```
....
5221.     &aucmd_win[i].auc_win->w_winvar.di_tv, copyID,
NULL, NULL);
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method set_ref_in_item_dict(


```
....
5550.          newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=120
Status	New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by prev at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5531.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5216	5550
Object	null	prev

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method garbage_collect(int testing)

```
....
5216.                                     NULL, NULL) ;
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_dict(

```
....
5550.          newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=121
Status	New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by prev at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5531.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Line	5211	5550
Object	null	prev

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method garbage_collect(int testing)

```
....
5211.                                     NULL, NULL);
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_dict(

```
....
5550.      newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 18:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=122>
Status New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by list_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5561.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5211	5580
Object	null	list_stack

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method garbage_collect(int testing)

```
....
5211.                                     NULL, NULL);
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_list(

```
....
5580.      newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=123
Status	New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by list_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5561.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5216	5580
Object	null	list_stack

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method garbage_collect(int testing)

```
....  
5216.                                     NULL, NULL);
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_list(
↓

```
....  
5580.      newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=124
Status	New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by list_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5561.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5221	5580
Object	null	list_stack

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method garbage_collect(int testing)

```
....
5221.          &aucmd_win[i].auc_win->w_winvar.di_tv, copyID,
NULL, NULL);
```



File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method set_ref_in_item_list(

```
....
5580.          newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 21:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=125>

Status New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by list_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5561.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5225	5580
Object	null	list_stack

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method garbage_collect(int testing)

```
....
5225.          NULL, NULL);
```



File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method set_ref_in_item_list(

```
....
5580.          newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 22:

Severity Low

Result State To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=126
Status	New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by list_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5561.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5229	5580
Object	null	list_stack

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method garbage_collect(int testing)

```
....  
5229.                                     NULL, NULL);
```



File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_list(

```
....  
5580.      newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=127
Status	New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by list_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5561.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5235	5580
Object	null	list_stack

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method garbage_collect(int testing)

```
....
5235.                                     NULL, NULL);
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_list(

```
....
5580.          newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 24:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=128>
Status New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5514 is not initialized when it is used by list_stack at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5561.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5523	5580
Object	null	list_stack

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_callback(callback_T *cb, int copyID)

```
....
5523.          return set_ref_in_item(&tv, copyID, NULL, NULL);
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_list(

```
....
5580.          newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 25:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=129>
Status New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5514 is not initialized when it is used by prev at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5561.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5523	5580
Object	null	prev

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_callback(callback_T *cb, int copyID)

```
....
5523.         return set_ref_in_item(&tv, copyID, NULL, NULL);
```



File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_list(

```
....
5580.         newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=130
Status	New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by prev at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5561.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5221	5580
Object	null	prev

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method garbage_collect(int testing)

```
....
5221.         &aucmd_win[i].auc_win->w_winvar.di_tv, copyID,
NULL, NULL);
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_list(

```
....  
5580.          newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 27:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=131>
Status New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by prev at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5561.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5216	5580
Object	null	prev

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method garbage_collect(int testing)

```
....  
5216.          NULL, NULL);
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_list(

```
....  
5580.          newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 28:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=132>
Status New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by prev at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5561.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5225	5580
Object	null	prev

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method garbage_collect(int testing)

```
....
5225.                                     NULL, NULL) ;
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_list(

```
....
5580.      newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=133
Status	New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by prev at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5561.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5235	5580
Object	null	prev

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method garbage_collect(int testing)

```
....
5235.                                     NULL, NULL) ;
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_list(

```
....
5580.          newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=134
Status	New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by prev at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5561.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	5229	5580
Object	null	prev

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method garbage_collect(int testing)

```
....
5229.                                     NULL, NULL) ;
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Method set_ref_in_item_list(

```
....
5580.          newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=135
Status	New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5150 is not initialized when it is used by prev at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5561.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Line	5211	5580
Object	null	prev

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method garbage_collect(int testing)

```
....
5211.                                     NULL, NULL);
```

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method set_ref_in_item_list(

```
....
5580.          newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 32:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=136>

Status New

The variable declared in null at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5863 is not initialized when it is used by class_name at vim@@vim-v9.0.1864-CVE-2023-0054-FP.c in line 5863.

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	6087	6089
Object	null	class_name

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method echo_string_core(

```
....
6087.          class_T *cl = obj == NULL ? NULL : obj->obj_class;
....
6089.                                     : cl->class_name);
```

NULL Pointer Dereference\Path 33:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=137>

Status New

The variable declared in null at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 260 is not initialized when it is used by skip at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 194.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	331	202
Object	null	skip

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c

Method eval_expr_typval(

```
....
331.          if (eval1_emsg(&s, rettv, NULL) == FAIL)
```

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c

Method eval1_emsg(char_u **arg, typval_T *rettv, exarg_T *eap)

```
....
202.          fill_evalarg_from_eap(&evalarg, eap, eap != NULL && eap-
>skip);
```

NULL Pointer Dereference\Path 34:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=138>

Status New

The variable declared in null at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 487 is not initialized when it is used by Pointer at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 487.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	544	544
Object	null	Pointer

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c

Method skip_expr_concatenate(

```
.....
544.          * ((char_u **)gap->ga_data) = NULL;
```

NULL Pointer Dereference\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=139
Status	New

The variable declared in null at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 639 is not initialized when it is used by skip at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 610.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	644	621
Object	null	skip

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method eval_to_string(

```
.....
644.          return eval_to_string_eap(arg, convert, NULL,
use_simple_function);
```



File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method eval_to_string_eap(

```
.....
621.          fill_evalarg_from_eap(&evalarg, eap, eap != NULL && eap-
>skip);
```

NULL Pointer Dereference\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=140
Status	New

The variable declared in null at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 1822 is not initialized when it is used by di_tv at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 1822.

Source	Destination
--------	-------------

File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	1882	1889
Object	null	di_tv

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method set_var_lval(

```

....
1882.          di = NULL;
....
1889.          di->di_tv.vval.v_typealias-
>ta_name);

```

NULL Pointer Dereference\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=141
Status	New

The variable declared in null at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 1822 is not initialized when it is used by di_tv at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 1822.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	1882	1886
Object	null	di_tv

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method set_var_lval(

```

....
1882.          di = NULL;
....
1886.          if (di != NULL && di->di_tv.v_type == VAR_TYPEALIAS)

```

NULL Pointer Dereference\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=142
Status	New

The variable declared in null at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5354 is not initialized when it is used by ht_stack at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5736.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	5415	5755
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method garbage_collect(int testing)

```
....  
5415.                                     NULL, NULL);
```

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method set_ref_in_item_dict(

```
....  
5755.      newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 39:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=143>
Status New

The variable declared in null at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5354 is not initialized when it is used by ht_stack at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5736.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	5420	5755
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method garbage_collect(int testing)

```
....  
5420.                                     NULL, NULL);
```

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c

Method set_ref_in_item_dict(

```
....
5755.          newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 40:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=144>

Status New

The variable declared in null at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5354 is not initialized when it is used by ht_stack at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5736.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	5425	5755
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c

Method garbage_collect(int testing)

```
....
5425.          &aucmd_win[i].auc_win->w_winvar.di_tv, copyID,
NULL, NULL);
```

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c

Method set_ref_in_item_dict(

```
....
5755.          newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 41:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=145>

Status New

The variable declared in null at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5354 is not initialized when it is used by ht_stack at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5736.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	5429	5755
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method garbage_collect(int testing)

```
....
5429.                                     NULL, NULL) ;
```

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method set_ref_in_item_dict(

```
....
5755.     newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=146
Status	New

The variable declared in null at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5354 is not initialized when it is used by ht_stack at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5736.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	5433	5755
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method garbage_collect(int testing)

```
....
5433.                                     NULL, NULL) ;
```

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method set_ref_in_item_dict(

```
....
5755.          newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=147
Status	New

The variable declared in null at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5354 is not initialized when it is used by ht_stack at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5736.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	5439	5755
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method garbage_collect(int testing)

```
....
5439.                                     NULL, NULL) ;
```

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method set_ref_in_item_dict(

```
....
5755.          newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=148
Status	New

The variable declared in null at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5719 is not initialized when it is used by ht_stack at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5736.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c

Line	5728	5755
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method set_ref_in_callback(callback_T *cb, int copyID)

```
....
5728.         return set_ref_in_item(&tv, copyID, NULL, NULL);
```

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method set_ref_in_item_dict(

```
....
5755.         newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 45:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=149>
Status New

The variable declared in null at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5354 is not initialized when it is used by prev at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5736.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	5420	5755
Object	null	prev

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method garbage_collect(int testing)

```
....
5420.                                     NULL, NULL);
```

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method set_ref_in_item_dict(

```
....
5755.         newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=150
Status	New

The variable declared in null at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5354 is not initialized when it is used by prev at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5736.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	5425	5755
Object	null	prev

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method garbage_collect(int testing)

```
....
5425.                &aucmd_win[i].auc_win->w_winvar.di_tv, copyID,
NULL, NULL);
```

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method set_ref_in_item_dict(

```
....
5755.        newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=151
Status	New

The variable declared in null at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5354 is not initialized when it is used by prev at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5736.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	5415	5755
Object	null	prev

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method garbage_collect(int testing)

```
....  
5415.                                     NULL, NULL);
```

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method set_ref_in_item_dict(

```
....  
5755.         newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 48:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=152>
Status New

The variable declared in null at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5719 is not initialized when it is used by prev at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5736.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	5728	5755
Object	null	prev

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method set_ref_in_callback(callback_T *cb, int copyID)

```
....  
5728.         return set_ref_in_item(&tv, copyID, NULL, NULL);
```

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method set_ref_in_item_dict(

```
....  
5755.         newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 49:

Severity Low
Result State To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=153
Status	New

The variable declared in null at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5354 is not initialized when it is used by prev at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5736.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	5439	5755
Object	null	prev

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method garbage_collect(int testing)

```
....
5439.                                     NULL, NULL);
```

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method set_ref_in_item_dict(

```
....
5755.     newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=154
Status	New

The variable declared in null at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5354 is not initialized when it is used by prev at vim@@vim-v9.0.2149-CVE-2021-4136-FP.c in line 5736.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	5429	5755
Object	null	prev

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method garbage_collect(int testing)

```
....
5429.
```

```
NULL, NULL);
```

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c

Method set_ref_in_item_dict(

```
....
5755.      newitem->prev = *ht_stack;
```

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=1050090&projectid=50079&pathid=53>

Status New

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	402	402
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method init_evalarg(evalarg_T *evalarg)

```
....
402.      ga_init2(&evalarg->eval_tofree_ga, sizeof(char_u *), 20);
```

Use of Sizeof On a Pointer Type\Path 2:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=54>

Status New

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	505	505

Object	sizeof	sizeof
--------	--------	--------

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method skip_expr_concatenate(

```
....  
505.          ga_init2(gap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 3:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=55>

Status New

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	509	509
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method skip_expr_concatenate(

```
....  
509.          ga_init2(freegap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 4:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=56>

Status New

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c	vim@@vim-v9.0.1864-CVE-2023-0054-FP.c
Line	3816	3816
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-0054-FP.c

Method eval8(


```
.....  
3816.          ga_init2(&type_list, sizeof(type_T *), 10);
```

Use of Sizeof On a Pointer Type\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=57
Status	New

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c
Line	1398	1398
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-2610-FP.c
Method cleanup_subexpr(void)

```
.....  
1398.          vim_memset(rex.reg_startp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=58
Status	New

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c
Line	1399	1399
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-2610-FP.c
Method cleanup_subexpr(void)

```
.....  
1399.          vim_memset(rex.reg_endp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 7:

Severity	Low
----------	-----

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=59
Status	New

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c
Line	1419	1419
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-2610-FP.c
Method cleanup_zsubexpr(void)

```
....  
1419.      vim_memset(reg_startzp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=60
Status	New

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c
Line	1420	1420
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-2610-FP.c
Method cleanup_zsubexpr(void)

```
....  
1420.      vim_memset(reg_endzp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=61
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	402	402
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method init_evalarg(evalarg_T *evalarg)

```
....  
402.          ga_init2(&evalarg->eval_tofree_ga, sizeof(char_u *), 20);
```

Use of Sizeof On a Pointer Type\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=62
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	505	505
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Method skip_expr_concatenate(

```
....  
505.          ga_init2(gap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=63
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	509	509

Object	sizeof	sizeof
--------	--------	--------

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c

Method skip_expr_concatenate(

```
....
509.          ga_init2(freegap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 12:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=64>

Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c	vim@@vim-v9.0.2149-CVE-2021-4136-FP.c
Line	4016	4016
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4136-FP.c

Method eval8(

```
....
4016.          ga_init2(&type_list, sizeof(type_T *), 10);
```

Use of Sizeof On a Pointer Type\Path 13:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=65>

Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	248	248
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c

Method get_function_args(

```
.....
248.          ga_init2(newargs, sizeof(char_u *), 3);
```

Use of Sizeof On a Pointer Type\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=66
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	250	250
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method get_function_args(

```
.....
250.          ga_init2(argtypes, sizeof(char_u *), 3);
```

Use of Sizeof On a Pointer Type\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=67
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	254	254
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method get_function_args(

```
.....
254.          ga_init2(default_args, sizeof(char_u *), 3);
```

Use of Sizeof On a Pointer Type\Path 16:

Severity	Low
----------	-----

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=68
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	524	524
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method parse_argument_types(

```
....  
524.      ga_init2(&fp->uf_type_list, sizeof(type_T *), 10);
```

Use of Sizeof On a Pointer Type\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=69
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	1349	1349
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method lambda_function_body(

```
....  
1349.      ga_init2(&newlines, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=70
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	1641	1641
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method get_lambda_tv(

```
.....  
1641.          ga_init2(&newlines, sizeof(char_u *), 1);
```

Use of Sizeof On a Pointer Type\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=71
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	1993	1993
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method get_func_tv(

```
.....  
1993.          ga_init2(&funcargs, sizeof(typval_T *), 50);
```

Use of Sizeof On a Pointer Type\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=72
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	2661	2661

Object	sizeof	sizeof
--------	--------	--------

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c

Method copy_lambda_to_global_func(

```
.....  
2661.                                sizeof(type_T *) * fp-  
>uf_args.ga_len);
```

Use of Sizeof On a Pointer Type\Path 21:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=73>

Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	2840	2840
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c

Method call_user_func(

```
.....  
2840.      ga_init2(&fc->fc_ufuncs, sizeof(ufunc_T *), 1);
```

Use of Sizeof On a Pointer Type\Path 22:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=74>

Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	4936	4936
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c

Method define_function(


```
.....
4936.          ga_init2(&newlines, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=75
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	5455	5455
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method ex_function(exarg_T *eap)

```
.....
5455.          ga_init2(&lines_to_free, sizeof(char_u *), 50);
```

Use of Sizeof On a Pointer Type\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=76
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	5748	5748
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method copy_function(ufunc_T *fp)

```
.....
5748.          sizeof(type_T *) * ufunc-
>uf_args.ga_len);
```

Use of Sizeof On a Pointer Type\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=77
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c
Line	1398	1398
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4192-FP.c
Method cleanup_subexpr(void)

```
....  
1398.      vim_memset(rex.reg_startp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=78
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c
Line	1399	1399
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4192-FP.c
Method cleanup_subexpr(void)

```
....  
1399.      vim_memset(rex.reg_endp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=79
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c
Line	1419	1419
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4192-FP.c
Method cleanup_zsubexpr(void)

```
....  
1419.      vim_memset(reg_startzp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=80
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c
Line	1420	1420
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4192-FP.c
Method cleanup_zsubexpr(void)

```
....  
1420.      vim_memset(reg_endzp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=81
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Line	402	402

Object	sizeof	sizeof
--------	--------	--------

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-1420-FP.c

Method init_evalarg(evalarg_T *evalarg)

```
....  
402.          ga_init2(&evalarg->eval_tofree_ga, sizeof(char_u *), 20);
```

Use of Sizeof On a Pointer Type\Path 30:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=82>

Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Line	505	505
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-1420-FP.c

Method skip_expr_concatenate(

```
....  
505.          ga_init2(gap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 31:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=83>

Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Line	509	509
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-1420-FP.c

Method skip_expr_concatenate(

```
.....  
509.          ga_init2(freegap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=84
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c	vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Line	4016	4016
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-1420-FP.c
Method eval8(

```
.....  
4016.          ga_init2(&type_list, sizeof(type_T *), 10);
```

Use of Sizeof On a Pointer Type\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=85
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Line	861	861
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Method diff_try_update(

```
.....  
861.          ga_init2(&dio->dio_diff.dout_ga, sizeof(char *), 1000);
```

Use of Sizeof On a Pointer Type\Path 34:

Severity	Low
----------	-----

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=86
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c
Line	861	861
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2210-FP.c
Method diff_try_update(

```
....  
861.          ga_init2(&dio->dio_diff.dout_ga, sizeof(char *), 1000);
```

Use of Sizeof On a Pointer Type\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=87
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Line	861	861
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Method diff_try_update(

```
....  
861.          ga_init2(&dio->dio_diff.dout_ga, sizeof(char *), 1000);
```

Use of Sizeof On a Pointer Type\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=88
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c
Line	1398	1398
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2345-FP.c
Method cleanup_subexpr(void)

```
....  
1398.      vim_memset(rex.reg_startp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 37:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=89>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c
Line	1399	1399
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2345-FP.c
Method cleanup_subexpr(void)

```
....  
1399.      vim_memset(rex.reg_endp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 38:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=90>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c
Line	1419	1419

Object	sizeof	sizeof
--------	--------	--------

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2345-FP.c
Method cleanup_zsubexpr(void)

```
....  
1419.          vim_memset(reg_startzp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 39:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=91>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c
Line	1420	1420
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2345-FP.c
Method cleanup_zsubexpr(void)

```
....  
1420.          vim_memset(reg_endzp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 40:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=92>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c
Line	861	861
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2598-FP.c
Method diff_try_update(


```
....  
861.          ga_init2(&dio->dio_diff.dout_ga, sizeof(char *), 1000);
```

Use of Sizeof On a Pointer Type\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=93
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2889-FP.c	vim@@vim-v9.0.2149-CVE-2022-2889-FP.c
Line	402	402
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2889-FP.c
Method init_evalarg(evalarg_T *evalarg)

```
....  
402.          ga_init2(&evalarg->eval_tofree_ga, sizeof(char_u *), 20);
```

Use of Sizeof On a Pointer Type\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=94
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2889-FP.c	vim@@vim-v9.0.2149-CVE-2022-2889-FP.c
Line	505	505
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2889-FP.c
Method skip_expr_concatenate(

```
....  
505.          ga_init2(gap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 43:

Severity	Low
----------	-----

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=95
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2889-FP.c	vim@@vim-v9.0.2149-CVE-2022-2889-FP.c
Line	509	509
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2889-FP.c
Method skip_expr_concatenate(

```
....  
509.          ga_init2(freegap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=96
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2889-FP.c	vim@@vim-v9.0.2149-CVE-2022-2889-FP.c
Line	4016	4016
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2889-FP.c
Method eval8(

```
....  
4016.          ga_init2(&type_list, sizeof(type_T *), 10);
```

Use of Sizeof On a Pointer Type\Path 45:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=97
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	1265	1265
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method read_compound(FILE *fd, slang_T *slang, int len)

```
....  
1265.          ga_init2(gap, sizeof(char_u *), c);
```

Use of Sizeof On a Pointer Type\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=98
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	1427	1427
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method set_sofo(slang_T *lp, char_u *from, char_u *to)

```
....  
1427.          ga_init2(gap, sizeof(int *), 1);
```

Use of Sizeof On a Pointer Type\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=99
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	1430	1430

Object	sizeof	sizeof
--------	--------	--------

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method set_sofo(slang_T *lp, char_u *from, char_u *to)

```
....  
1430.          vim_memset(gap->ga_data, 0, sizeof(int *) * 256);
```

Use of Sizeof On a Pointer Type\Path 48:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=100>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	4299	4299
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method getroom(

```
....  
4299.          bl->sb_used = (bl->sb_used + sizeof(char *) - 1)
```

Use of Sizeof On a Pointer Type\Path 49:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=101>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	4300	4300
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method getroom(

```
.....
4300.                                & ~(sizeof(char *) - 1);
```

Use of Sizeof On a Pointer Type\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=102
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	5938	5938
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method mkspell(

```
.....
5938.          ga_init2(&spin.si_comppat, sizeof(char_u *), 20);
```

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=1050090&projectid=50079&pathid=533
Status	New

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c
Line	2802	2802
Object	fprintf	fprintf

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-2610-FP.c
Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....  
2802.                fprintf(f, "Syntax error in \"%s\"\\n", expr);
```

Improper Resource Access Authorization\Path 2:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=534>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c
Line	2778	2778
Object	fprintf	fprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4192-FP.c
Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....  
2778.                fprintf(f, "Syntax error in \"%s\"\\n", expr);
```

Improper Resource Access Authorization\Path 3:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=535>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c
Line	2778	2778
Object	fprintf	fprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2345-FP.c
Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....  
2778.                fprintf(f, "Syntax error in \"%s\"\\n", expr);
```

Improper Resource Access Authorization\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=536
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	6354	6354
Object	fprintf	fprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method spell_add_word(

```
....  
6354.                fprintf(fd, "%.*s/!\n", len, word);
```

Improper Resource Access Authorization\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=537
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	6356	6356
Object	fprintf	fprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method spell_add_word(

```
....  
6356.                fprintf(fd, "%.*s/?\n", len, word);
```

Improper Resource Access Authorization\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=538

Status	New
--------	-----

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	6358	6358
Object	fprintf	fprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method spell_add_word(

```
....  
6358.          fprintf(fd, "%.*s\n", len, word);
```

Improper Resource Access Authorization\Path 7:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=539>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	4968	4968
Object	fputc	fputc

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
4968.          fputc(128, fd);                                // <charflagslen>
```

Improper Resource Access Authorization\Path 8:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=540>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c

Line	4976	4976
Object	fputc	fputc

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
4976.                fputc(flags, fd);                // <charflags>
```

Improper Resource Access Authorization\Path 9:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=541>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	6305	6305
Object	fputc	fputc

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method spell_add_word(

```
....  
6305.                fputc('#', fd);
```

Improper Resource Access Authorization\Path 10:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=542>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	6626	6626
Object	fputc	fputc

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c

Method write_spell_prefcond(FILE *fd, garray_T *gap, size_t *fwv)

```
....  
6626.          fputc(len, fd);
```

Improper Resource Access Authorization\Path 11:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=543>

Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	6632	6632
Object	fputc	fputc

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c

Method write_spell_prefcond(FILE *fd, garray_T *gap, size_t *fwv)

```
....  
6632.          fputc(0, fd);
```

Improper Resource Access Authorization\Path 12:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=544>

Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Line	1035	1035
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2208-FP.c

Method check_external_diff(diffio_T *diffio)

```
....  
1035.          if (fwrite("line1\n", (size_t)6, (size_t)1, fd) != 1)
```

Improper Resource Access Authorization\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=545
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Line	1043	1043
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Method check_external_diff(diffio_T *diffio)

```
....  
1043.          if (fwrite("line2\n", (size_t)6, (size_t)1, fd) != 1)
```

Improper Resource Access Authorization\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=546
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c
Line	1035	1035
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2210-FP.c
Method check_external_diff(diffio_T *diffio)

```
....  
1035.          if (fwrite("line1\n", (size_t)6, (size_t)1, fd) != 1)
```

Improper Resource Access Authorization\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=547
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c
Line	1043	1043
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2210-FP.c
Method check_external_diff(diffio_T *diffio)

```
....  
1043.                if (fwrite("line2\n", (size_t)6, (size_t)1, fd) != 1)
```

Improper Resource Access Authorization\Path 16:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=548>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Line	1035	1035
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Method check_external_diff(diffio_T *diffio)

```
....  
1035.                if (fwrite("line1\n", (size_t)6, (size_t)1, fd) != 1)
```

Improper Resource Access Authorization\Path 17:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=549>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Line	1043	1043

Object	fwrite	fwrite
--------	--------	--------

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Method check_external_diff(diffio_T *diffio)

```
....  
1043.                if (fwrite("line2\n", (size_t)6, (size_t)1, fd) != 1)
```

Improper Resource Access Authorization\Path 18:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=550>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c
Line	1035	1035
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2598-FP.c
Method check_external_diff(diffio_T *diffio)

```
....  
1035.                if (fwrite("line1\n", (size_t)6, (size_t)1, fd) != 1)
```

Improper Resource Access Authorization\Path 19:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=551>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c
Line	1043	1043
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2598-FP.c
Method check_external_diff(diffio_T *diffio)

```
.....  
1043.                if (fwrite("line2\n", (size_t)6, (size_t)1, fd) != 1)
```

Improper Resource Access Authorization\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=552
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	4903	4903
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
.....  
4903.                fvw &= fwrite(VIMSPELLMAGIC, VIMSPELLMAGICL, (size_t)1, fd);
```

Improper Resource Access Authorization\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=553
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	4922	4922
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
.....  
4922.                fvw &= fwrite(spin->si_info, (size_t)i, (size_t)1, fd); //  
<infotext>
```

Improper Resource Access Authorization\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=554
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	4933	4933
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
4933.          fwv &= fwrite(spin->si_region_name, (size_t)1, (size_t)1,  
fd);
```

Improper Resource Access Authorization\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=555
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	4980	4980
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
4980.          fwv &= fwrite(folchars, (size_t)1, (size_t)1, fd); //  
<folchars>
```

Improper Resource Access Authorization\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=556

Status	New
--------	-----

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	4991	4991
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
4991.          fvw &= fwrite(spin->si_midword, (size_t)i, (size_t)1, fd);
```

Improper Resource Access Authorization\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=557
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	5079	5079
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
5079.          fvw &= fwrite(p, 1, (size_t)1, fd);
```

Improper Resource Access Authorization\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=558
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c

Line	5097	5097
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
5097.          fwv &= fwrite(spin->si_sofofr, 1, (size_t)1, fd); //  
<sofofrom>
```

Improper Resource Access Authorization\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=559
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	5101	5101
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
5101.          fwv &= fwrite(spin->si_sofoto, 1, (size_t)1, fd); //  
<sofoto>
```

Improper Resource Access Authorization\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=560
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	5126	5126
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
5126.                fvw &= fwrite(hi->hi_key, (size_t)1, (size_t)1,  
fd);
```

Improper Resource Access Authorization\Path 29:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=561>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	5142	5142
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
5142.                fvw &= fwrite(spin->si_map.ga_data, (size_t)1, (size_t)1,  
fd);
```

Improper Resource Access Authorization\Path 30:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=562>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	5208	5208
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
5208.          fwv &= fwrite(p, (size_t)STRLEN(p), (size_t)1, fd);
```

Improper Resource Access Authorization\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=563
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	5212	5212
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
5212.          fwv &= fwrite(spin->si_compflags, (size_t)STRLEN(spin->si_compflags),
```

Improper Resource Access Authorization\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=564
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	5236	5236
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
5236.          fwv &= fwrite(spin->si_syllable, (size_t)1, (size_t)1, fd);
```

Improper Resource Access Authorization\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=565
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	5839	5839
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method sug_write(spellinfo_T *spin, char_u *fname)

```
....  
5839.      if (fwrite(VIMSUGMAGIC, VIMSUGMAGICL, (size_t)1, fd) != 1) //  
<fileID>
```

Improper Resource Access Authorization\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=566
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	5881	5881
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method sug_write(spellinfo_T *spin, char_u *fname)

```
....  
5881.      if (fwrite(line, (size_t)len, (size_t)1, fd) == 0)
```

Improper Resource Access Authorization\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=567
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	6627	6627
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method write_spell_prefcond(FILE *fd, garray_T *gap, size_t *fwv)

```
....
6627.                *fwv &= fwrite(p, (size_t)len, (size_t)1, fd);
```

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=1050090&projectid=50079&pathid=317
Status	New

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c
Line	2561	2561
Object	len	len

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-2610-FP.c
Method reg_submatch(int no)

```
....
2561.                retval[len] = '\n';
```

Unchecked Array Index\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=318

Status	New
--------	-----

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c
Line	2572	2572
Object	len	len

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-2610-FP.c
Method reg_submatch(int no)

```
....  
2572.                retval[len] = '\n';
```

Unchecked Array Index\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=319
Status	New

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c
Line	2580	2580
Object	len	len

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-2610-FP.c
Method reg_submatch(int no)

```
....  
2580.                retval[len] = NUL;
```

Unchecked Array Index\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=320
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c

Line	1817	1817
Object	len	len

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c

Method deref_func_name(

```
....  
1817.          name[len] = NUL;
```

Unchecked Array Index\Path 5:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=321>

Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	1822	1822
Object	len	len

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c

Method deref_func_name(

```
....  
1822.          name[len] = cc;
```

Unchecked Array Index\Path 6:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=322>

Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c
Line	2537	2537
Object	len	len

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4192-FP.c

Method reg_submatch(int no)

```
....  
2537.                retval[len] = '\n';
```

Unchecked Array Index\Path 7:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=323>

Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c
Line	2548	2548
Object	len	len

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4192-FP.c

Method reg_submatch(int no)

```
....  
2548.                retval[len] = '\n';
```

Unchecked Array Index\Path 8:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=324>

Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c
Line	2556	2556
Object	len	len

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4192-FP.c

Method reg_submatch(int no)

```
....  
2556.                retval[len] = NUL;
```

Unchecked Array Index\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=325
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Line	1884	1884
Object	idx_new	idx_new

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Method diff_copy_entry(

```
....  
1884.      dp->df_lnum[idx_new] = dp->df_lnum[idx_orig] - off;
```

Unchecked Array Index\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=326
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c	vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Line	1885	1885
Object	idx_new	idx_new

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2208-FP.c
Method diff_copy_entry(

```
....  
1885.      dp->df_count[idx_new] = dp->df_count[idx_orig];
```

Unchecked Array Index\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=327
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c
Line	1884	1884
Object	idx_new	idx_new

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2210-FP.c

Method diff_copy_entry(

```
.....
1884.         dp->df_lnum[idx_new] = dp->df_lnum[idx_orig] - off;
```

Unchecked Array Index\Path 12:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=328>

Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c	vim@@vim-v9.0.2149-CVE-2022-2210-FP.c
Line	1885	1885
Object	idx_new	idx_new

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2210-FP.c

Method diff_copy_entry(

```
.....
1885.         dp->df_count[idx_new] = dp->df_count[idx_orig];
```

Unchecked Array Index\Path 13:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=329>

Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Line	1884	1884

Object	idx_new	idx_new
--------	---------	---------

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2289-FP.c

Method diff_copy_entry(

```
.....
1884.         dp->df_lnum[idx_new] = dp->df_lnum[idx_orig] - off;
```

Unchecked Array Index\Path 14:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=330>

Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c	vim@@vim-v9.0.2149-CVE-2022-2289-FP.c
Line	1885	1885
Object	idx_new	idx_new

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2289-FP.c

Method diff_copy_entry(

```
.....
1885.         dp->df_count[idx_new] = dp->df_count[idx_orig];
```

Unchecked Array Index\Path 15:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=331>

Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c
Line	2537	2537
Object	len	len

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2345-FP.c

Method reg_submatch(int no)

```
.....
2537.                retval[len] = '\n';
```

Unchecked Array Index\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=332
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c
Line	2548	2548
Object	len	len

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2345-FP.c
Method reg_submatch(int no)

```
.....
2548.                retval[len] = '\n';
```

Unchecked Array Index\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=333
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c
Line	2556	2556
Object	len	len

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2345-FP.c
Method reg_submatch(int no)

```
.....
2556.                retval[len] = NUL;
```

Unchecked Array Index\Path 18:

Severity	Low
----------	-----

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=334
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c
Line	1884	1884
Object	idx_new	idx_new

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2598-FP.c
Method diff_copy_entry(

```
....  
1884.      dp->df_lnum[idx_new] = dp->df_lnum[idx_orig] - off;
```

Unchecked Array Index\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=335
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c	vim@@vim-v9.0.2149-CVE-2022-2598-FP.c
Line	1885	1885
Object	idx_new	idx_new

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2598-FP.c
Method diff_copy_entry(

```
....  
1885.      dp->df_count[idx_new] = dp->df_count[idx_orig];
```

Unchecked Array Index\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=336
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	6691	6691
Object	cl	cl

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method set_map_str(slang_T *lp, char_u *map)

```
....  
6691.          b[cl] = NUL;
```

Unchecked Array Index\Path 21:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=337>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	1569	1569
Object	i	i

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method mb_str2wide(char_u *s)

```
....  
1569.          res[i] = NUL;
```

Unchecked Array Index\Path 22:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=338>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	3774	3774

Object	cnt	cnt
--------	-----	-----

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method get_pfxlist(

```
....  
3774.      store_afflist[cnt] = NUL;
```

Unchecked Array Index\Path 23:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=339>
Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c	vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Line	3810	3810
Object	cnt	cnt

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2923-FP.c
Method get_compflags(

```
....  
3810.      store_afflist[cnt] = NUL;
```

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=1050090&projectid=50079&pathid=43>
Status New

The get_lambda_name method calls the sprintf function, at line 690 of vim@@vim-v9.0.2149-CVE-2021-4187-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	695	695
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method get_lambda_name(void)

```
....  
695.          sprintf((char*)name, "<lambda>%d", ++lambda_no);
```

Unchecked Return Value\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=44
Status	New

The fname_trans_sid method calls the sprintf function, at line 2046 of vim@@vim-v9.0.2149-CVE-2021-4187-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	2070	2070
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method fname_trans_sid(

```
....  
2070.          sprintf((char *)fname_buf + 3, "%ld_",
```

Unchecked Return Value\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=45
Status	New

The call_user_func method calls the sprintf function, at line 2788 of vim@@vim-v9.0.2149-CVE-2021-4187-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	2975	2975
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method call_user_func(

```
....  
2975.          sprintf((char *)numbuf, "%d", ai + 1);
```

Unchecked Return Value\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=46
Status	New

The delete_script_functions method calls the sprintf function, at line 3348 of vim@@vim-v9.0.2149-CVE-2021-4187-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	3359	3359
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method delete_script_functions(int sid)

```
....  
3359.          sprintf((char *)buf + 3, "%d_", sid);
```

Unchecked Return Value\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=47
Status	New

The trans_function_name_ext method calls the sprintf function, at line 4182 of vim@@vim-v9.0.2149-CVE-2021-4187-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	4441	4441
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method trans_function_name_ext(

```
....  
4441.          sprintf((char *)sid_buf, "%ld_",  
(long)current_sctx.sc_sid);
```

Unchecked Return Value\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=48
Status	New

The define_function method calls the sprintf function, at line 4666 of vim@@vim-v9.0.2149-CVE-2021-4187-FP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c	vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Line	5137	5137
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4187-FP.c
Method define_function(

```
....  
5137.          sprintf(numbuf, "%d", ++func_nr);
```

Unchecked Return Value\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=49
Status	New

The `ex_compiler` method calls the `sprintf` function, at line 723 of `vim@@vim-v9.0.2149-CVE-2022-2816-FP.c`. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	<code>vim@@vim-v9.0.2149-CVE-2022-2816-FP.c</code>	<code>vim@@vim-v9.0.2149-CVE-2022-2816-FP.c</code>
Line	764	764
Object	<code>sprintf</code>	<code>sprintf</code>

Code Snippet

File Name `vim@@vim-v9.0.2149-CVE-2022-2816-FP.c`

Method `ex_compiler(exarg_T *eap)`

```
....  
764.      sprintf((char *)buf, "compiler/%s.vim", eap->arg);
```

Unchecked Return Value\Path 8:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=50>

Status New

The `spell_read_aff` method calls the `sprintf` function, at line 2186 of `vim@@vim-v9.0.2149-CVE-2022-2923-FP.c`. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	<code>vim@@vim-v9.0.2149-CVE-2022-2923-FP.c</code>	<code>vim@@vim-v9.0.2149-CVE-2022-2923-FP.c</code>
Line	2746	2746
Object	<code>sprintf</code>	<code>sprintf</code>

Code Snippet

File Name `vim@@vim-v9.0.2149-CVE-2022-2923-FP.c`

Method `spell_read_aff(spellinfo_T *spin, char_u *fname)`

```
....  
2746.      sprintf((char *)buf, "^%s", items[4]);
```

Unchecked Return Value\Path 9:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=51>

Status New

The `spell_read_aff` method calls the `sprintf` function, at line 2186 of `vim@@vim-v9.0.2149-CVE-2022-2923-FP.c`. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	<code>vim@@vim-v9.0.2149-CVE-2022-2923-FP.c</code>	<code>vim@@vim-v9.0.2149-CVE-2022-2923-FP.c</code>
Line	2748	2748
Object	<code>sprintf</code>	<code>sprintf</code>

Code Snippet

File Name `vim@@vim-v9.0.2149-CVE-2022-2923-FP.c`
Method `spell_read_aff(spellinfo_T *spin, char_u *fname)`

```
....  
2748.                                sprintf((char *)buf, "%s$", items[4]);
```

Unchecked Return Value\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=52
Status	New

The `spell_read_aff` method calls the `sprintf` function, at line 2186 of `vim@@vim-v9.0.2149-CVE-2022-2923-FP.c`. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	<code>vim@@vim-v9.0.2149-CVE-2022-2923-FP.c</code>	<code>vim@@vim-v9.0.2149-CVE-2022-2923-FP.c</code>
Line	2807	2807
Object	<code>sprintf</code>	<code>sprintf</code>

Code Snippet

File Name `vim@@vim-v9.0.2149-CVE-2022-2923-FP.c`
Method `spell_read_aff(spellinfo_T *spin, char_u *fname)`

```
....  
2807.                                sprintf((char *)buf, "^%s",
```

Potential Precision Problem

Query Path:

CPP\Cx\CPP Buffer Overflow\Potential Precision Problem Version:0

Categories

NIST SP 800-53: SI-10 Information Input Validation (P1)

OWASP Top 10 2017: A1-Injection

Description

Potential Precision Problem\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=310
Status	New

The size of the buffer used by `ex_compiler` in `"compiler/%s.vim"`, at line 723 of `vim@@vim-v9.0.2149-CVE-2022-2816-FP.c`, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that `ex_compiler` passes to `"compiler/%s.vim"`, at line 723 of `vim@@vim-v9.0.2149-CVE-2022-2816-FP.c`, to overwrite the target buffer.

	Source	Destination
File	<code>vim@@vim-v9.0.2149-CVE-2022-2816-FP.c</code>	<code>vim@@vim-v9.0.2149-CVE-2022-2816-FP.c</code>
Line	764	764
Object	<code>"compiler/%s.vim"</code>	<code>"compiler/%s.vim"</code>

Code Snippet

File Name `vim@@vim-v9.0.2149-CVE-2022-2816-FP.c`
 Method `ex_compiler(exarg_T *eap)`

```
....
764.      sprintf((char *)buf, "compiler/%s.vim", eap->arg);
```

Potential Precision Problem\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=311
Status	New

The size of the buffer used by `spell_read_aff` in `"^%s"`, at line 2186 of `vim@@vim-v9.0.2149-CVE-2022-2923-FP.c`, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that `spell_read_aff` passes to `"^%s"`, at line 2186 of `vim@@vim-v9.0.2149-CVE-2022-2923-FP.c`, to overwrite the target buffer.

	Source	Destination
File	<code>vim@@vim-v9.0.2149-CVE-2022-2923-FP.c</code>	<code>vim@@vim-v9.0.2149-CVE-2022-2923-FP.c</code>
Line	2746	2746
Object	<code>"^%s"</code>	<code>"^%s"</code>

Code Snippet

File Name `vim@@vim-v9.0.2149-CVE-2022-2923-FP.c`
 Method `spell_read_aff(spellinfo_T *spin, char_u *fname)`

```
.....
2746.                                sprintf((char *)buf, "%s", items[4]);
```

Potential Precision Problem\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=312
Status	New

The size of the buffer used by `spell_read_aff` in `"%s$"`, at line 2186 of `vim@@vim-v9.0.2149-CVE-2022-2923-FP.c`, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that `spell_read_aff` passes to `"%s$"`, at line 2186 of `vim@@vim-v9.0.2149-CVE-2022-2923-FP.c`, to overwrite the target buffer.

	Source	Destination
File	<code>vim@@vim-v9.0.2149-CVE-2022-2923-FP.c</code>	<code>vim@@vim-v9.0.2149-CVE-2022-2923-FP.c</code>
Line	2748	2748
Object	<code>"%s\$"</code>	<code>"%s\$"</code>

Code Snippet

File Name `vim@@vim-v9.0.2149-CVE-2022-2923-FP.c`
 Method `spell_read_aff(spellinfo_T *spin, char_u *fname)`

```
.....
2748.                                sprintf((char *)buf, "%s$", items[4]);
```

Potential Precision Problem\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=313
Status	New

The size of the buffer used by `spell_read_aff` in `"^%s"`, at line 2186 of `vim@@vim-v9.0.2149-CVE-2022-2923-FP.c`, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that `spell_read_aff` passes to `"^%s"`, at line 2186 of `vim@@vim-v9.0.2149-CVE-2022-2923-FP.c`, to overwrite the target buffer.

	Source	Destination
File	<code>vim@@vim-v9.0.2149-CVE-2022-2923-FP.c</code>	<code>vim@@vim-v9.0.2149-CVE-2022-2923-FP.c</code>
Line	2807	2807
Object	<code>"^%s"</code>	<code>"^%s"</code>

Code Snippet

File Name `vim@@vim-v9.0.2149-CVE-2022-2923-FP.c`

Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
....  
2807.                                sprintf((char *)buf, "^%s",
```

Arithmenic Operation On Boolean

Query Path:

CPP\Cx\CPP Low Visibility\Arithmenic Operation On Boolean Version:1

Categories

FISMA 2014: Audit And Accountability

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

Description

Arithmenic Operation On Boolean\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=314
Status	New

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c
Line	1351	1351
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-2610-FP.c

Method reg_match_visual(void)

```
....  
1351.            if (cols < start || cols > end - (*p_sel == 'e'))
```

Arithmenic Operation On Boolean\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=315
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c
Line	1351	1351
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4192-FP.c

Method reg_match_visual(void)

```
....
1351.          if (cols < start || cols > end - (*p_sel == 'e'))
```

Arithmenic Operation On Boolean\Path 3:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=316>

Status New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c
Line	1351	1351
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2345-FP.c

Method reg_match_visual(void)

```
....
1351.          if (cols < start || cols > end - (*p_sel == 'e'))
```

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=1050090&projectid=50079&pathid=568>

Status New

	Source	Destination
File	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c

Line	2799	2799
Object	f	f

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-2610-FP.c
Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....  
2799.          f = fopen(BT_REGEX_DEBUG_LOG_NAME, "a");
```

Incorrect Permission Assignment For Critical Resources\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=569
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c
Line	2775	2775
Object	f	f

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4192-FP.c
Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....  
2775.          f = fopen(BT_REGEX_DEBUG_LOG_NAME, "a");
```

Incorrect Permission Assignment For Critical Resources\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=570
Status	New

	Source	Destination
File	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c
Line	2775	2775
Object	f	f

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2345-FP.c

Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....  
2775.          f = fopen(BT_REGEX_DEBUG_LOG_NAME, "a");
```

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=1050090&projectid=50079&pathid=571
Status	New

The vim_regcomp method in vim@@vim-v9.0.1864-CVE-2023-2610-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	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c	vim@@vim-v9.0.1864-CVE-2023-2610-FP.c
Line	2799	2799
Object	fopen	fopen

Code Snippet

File Name vim@@vim-v9.0.1864-CVE-2023-2610-FP.c
Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....  
2799.          f = fopen(BT_REGEX_DEBUG_LOG_NAME, "a");
```

TOCTOU\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=572
Status	New

The vim_regcomp method in vim@@vim-v9.0.2149-CVE-2021-4192-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	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c	vim@@vim-v9.0.2149-CVE-2021-4192-FP.c
Line	2775	2775

Object	fopen	fopen
--------	-------	-------

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2021-4192-FP.c
Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....
2775.          f = fopen(BT_REGEX_DEBUG_LOG_NAME, "a");
```

TOCTOU\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=573
Status	New

The vim_regcomp method in vim@@vim-v9.0.2149-CVE-2022-2345-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	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c	vim@@vim-v9.0.2149-CVE-2022-2345-FP.c
Line	2775	2775
Object	fopen	fopen

Code Snippet

File Name vim@@vim-v9.0.2149-CVE-2022-2345-FP.c
Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....
2775.          f = fopen(BT_REGEX_DEBUG_LOG_NAME, "a");
```

Potential Off by One Error in Loops

Query Path:

CPP\Cx\CPP Heuristic\Potential Off by One Error in Loops Version:1

Categories

PCI DSS v3.2: PCI DSS (3.2) - 6.5.1 - Injection flaws - particularly SQL injection
NIST SP 800-53: SI-16 Memory Protection (P1)
OWASP Top 10 2017: A1-Injection

Description

Potential Off by One Error in Loops\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050090&projectid=50079&pathid=104
Status	New

The buffer allocated by `<=` in `vim@@vim-v9.0.2149-CVE-2021-4187-FP.c` at line 826 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	<code>vim@@vim-v9.0.2149-CVE-2021-4187-FP.c</code>	<code>vim@@vim-v9.0.2149-CVE-2021-4187-FP.c</code>
Line	845	845
Object	<code><=</code>	<code><=</code>

Code Snippet

File Name `vim@@vim-v9.0.2149-CVE-2021-4187-FP.c`

Method `function_using_block_scopes(ufunc_T *fp, cstack_T *cstack)`

```
....  
845.      for (i = 0; i <= cstack->cs_idx; ++i)
```

Divide By Zero

Risk

What might happen

When a program divides a number by zero, an exception will be raised. If this exception is not handled by the application, unexpected results may occur, including crashing the application. This can be considered a DoS (Denial of Service) attack, if an external user has control of the value of the denominator or can cause this error to occur.

Cause

How does it happen

The program receives an unexpected value, and uses it for division without filtering, validation, or verifying that the value is not zero. The application does not explicitly handle this error or prevent division by zero from occurring.

General Recommendations

How to avoid it

- Before dividing by an unknown value, validate the number and explicitly ensure it does not evaluate to zero.
- Validate all untrusted input from all sources, in particular verifying that it is not zero before dividing with it.
- Verify output of methods, calculations, dictionary lookups, and so on, and ensure it is not zero before dividing with the result.
- Ensure divide-by-zero errors are caught and handled appropriately.

Source Code Examples

Java

Divide by Zero

```
public float getAverage(HttpServletRequest req) {  
    int total = Integer.parseInt(req.getParameter("total"));  
    int count = Integer.parseInt(req.getParameter("count"));  
  
    return total / count;  
}
```

Checked Division

```
public float getAverage(HttpServletRequest req) {  
    int total = Integer.parseInt(req.getParameter("total"));  
    int count = Integer.parseInt(req.getParameter("count"));  
  
    if (count > 0)  
        return total / count;  
    else  
        return 0;  
}
```

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;  
}
```

Char Overflow

Risk

What might happen

Assigning large data types into smaller data types, without proper checks and explicit casting, will lead to undefined behavior and unintentional effects, such as data corruption (e.g. value wraparound, wherein maximum values become minimum values); system crashes; infinite loops; logic errors, such as bypassing of security mechanisms; or even buffer overflows leading to arbitrary code execution.

Cause

How does it happen

This flaw can occur when implicitly casting numerical data types of a larger size, into a variable with a data type of a smaller size. This forces the program to discard some bits of information from the number. Depending on how the numerical data types are stored in memory, this is often the bits with the highest value, causing substantial corruption of the stored number. Alternatively, the sign bit of a signed integer could be lost, completely reversing the intention of the number.

General Recommendations

How to avoid it

- Avoid casting larger data types to smaller types.
 - Prefer promoting the target variable to a large enough data type.
 - If downcasting is necessary, always check that values are valid and in range of the target type, before casting
-

Source Code Examples

CPP

Unsafe Downsize Casting

```
int unsafe_addition(short op1, int op2) {  
    // op2 gets forced from int into a short  
    short total = op1 + op2;  
    return total;  
}
```

Safer Use of Proper Data Types

```
int safe_addition(short op1, int op2) {  
    // total variable is of type int, the largest type that is needed  
    int total = 0;  
    // check if total will overflow available integer size  
    if (INT_MAX - abs(op2) > op1)
```

```
{
    total = op1 + op2;
}
else
{
    // instead of overflow, saturate (but this is not always a good thing)
    total = INT_MAX
}

return total;
}
```


Integer Overflow

Risk

What might happen

Assigning large data types into smaller data types, without proper checks and explicit casting, will lead to undefined behavior and unintentional effects, such as data corruption (e.g. value wraparound, wherein maximum values become minimum values); system crashes; infinite loops; logic errors, such as bypassing of security mechanisms; or even buffer overflows leading to arbitrary code execution.

Cause

How does it happen

This flaw can occur when implicitly casting numerical data types of a larger size, into a variable with a data type of a smaller size. This forces the program to discard some bits of information from the number. Depending on how the numerical data types are stored in memory, this is often the bits with the highest value, causing substantial corruption of the stored number. Alternatively, the sign bit of a signed integer could be lost, completely reversing the intention of the number.

General Recommendations

How to avoid it

- Avoid casting larger data types to smaller types.
 - Prefer promoting the target variable to a large enough data type.
 - If downcasting is necessary, always check that values are valid and in range of the target type, before casting
-

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;
}
```

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());
```

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 function. These functions often return error codes in the result, or share other status codes with its 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 (strcmp(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 (! strcmp(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 `strcmp()` 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](#)

Potential Off by One Error in Loops

Risk

What might happen

An off by one error may result in overwriting or over-reading of unintended memory; in most cases, this can result in unexpected behavior and even application crashes. In other cases, where allocation can be controlled by an attacker, a combination of variable assignment and an off by one error can result in execution of malicious code.

Cause

How does it happen

Often when designating variables to memory, a calculation error may occur when determining size or length that is off by one.

For example in loops, when allocating an array of size 2, its cells are counted as 0,1 - therefore, if a For loop iterator on the array is incorrectly set with the start condition `i=0` and the continuation condition `i<=2`, three cells will be accessed instead of 2, and an attempt will be made to write or read cell [2], which was not originally allocated, resulting in potential corruption of memory outside the bounds of the originally assigned array.

Another example occurs when a null-byte terminated string, in the form of a character array, is copied without its terminating null-byte. Without the null-byte, the string representation is unterminated, resulting in certain functions to over-read memory as they expect the missing null terminator.

General Recommendations

How to avoid it

- Always ensure that a given iteration boundary is correct:
 - With array iterations, consider that arrays begin with cell 0 and end with cell `n-1`, for a size `n` array.
 - With character arrays and null-byte terminated string representations, consider that the null byte is required and should not be overwritten or ignored; ensure functions in use are not vulnerable to off-by-one, specifically for instances where null-bytes are automatically appended after the buffer, instead of in place of its last character.
 - Where possible, use safe functions that manage memory and are not prone to off-by-one errors.
-

Source Code Examples

CPP

Off-By-One in For Loop

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
for (int i = 0; i <= 5; i++)
{
    ptr[i] = i * 2 + 1; // ptr[5] will be set, but is out of bounds
}
```

```
}
```

Proper Iteration in For Loop

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
for (int i = 0; i < 5; i++)
{
    ptr[i] = i * 2 + 1; // ptr[0-4] are well defined
}
```

Off-By-One in strncat

```
strncat(buf, input, sizeof(buf) - strlen(buf)); // actual value should be sizeof(buf) -  
strlen(buf)-1 - this form will overwrite the terminating nullbyte
```

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

Potential Precision Problem

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 its 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 (strlen(inputString, MAX_INPUT_SIZE) < sizeof(buffer))  
    {  
        strncpy(buffer, inputString, sizeof(buffer));  
    }  
}
```


Indicator of Poor Code Quality

Weakness ID: 398 (*Weakness Class*)

Status: Draft

Description

Description Summary

The code has features that do not directly introduce a weakness or vulnerability, but indicate that the product has not been carefully developed or maintained.

Extended Description

Programs are more likely to be secure when good development practices are followed. If a program is complex, difficult to maintain, not portable, or shows evidence of neglect, then there is a higher likelihood that weaknesses are buried in the code.

Time of Introduction

- Architecture and Design
- Implementation

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	18	Source Code	Development Concepts (primary)699
ChildOf	Weakness Class	710	Coding Standards Violation	Research Concepts (primary)1000
ParentOf	Weakness Variant	107	Struts: Unused Validation Form	Research Concepts (primary)1000
ParentOf	Weakness Variant	110	Struts: Validator Without Form Field	Research Concepts (primary)1000
ParentOf	Category	399	Resource Management Errors	Development Concepts (primary)699
ParentOf	Weakness Base	401	Failure to Release Memory Before Removing Last Reference ('Memory Leak')	Seven Pernicious Kingdoms (primary)700
ParentOf	Weakness Base	404	Improper Resource Shutdown or Release	Development Concepts699 Seven Pernicious Kingdoms (primary)700
ParentOf	Weakness Variant	415	Double Free	Seven Pernicious Kingdoms (primary)700
ParentOf	Weakness Base	416	Use After Free	Seven Pernicious Kingdoms (primary)700
ParentOf	Weakness Variant	457	Use of Uninitialized Variable	Seven Pernicious Kingdoms (primary)700
ParentOf	Weakness Base	474	Use of Function with Inconsistent Implementations	Development Concepts (primary)699 Seven Pernicious Kingdoms (primary)700 Research Concepts (primary)1000
ParentOf	Weakness Base	475	Undefined Behavior for Input to API	Development Concepts (primary)699 Seven Pernicious Kingdoms (primary)700
ParentOf	Weakness Base	476	NULL Pointer Dereference	Development Concepts

				(primary)699 Seven Pernicious Kingdoms (primary)700 Research Concepts (primary)1000
ParentOf	Weakness Base	477	Use of Obsolete Functions	Development Concepts (primary)699 Seven Pernicious Kingdoms (primary)700 Research Concepts (primary)1000
ParentOf	Weakness Variant	478	Missing Default Case in Switch Statement	Development Concepts (primary)699
ParentOf	Weakness Variant	479	Unsafe Function Call from a Signal Handler	Development Concepts (primary)699
ParentOf	Weakness Variant	483	Incorrect Block Delimitation	Development Concepts (primary)699
ParentOf	Weakness Base	484	Omitted Break Statement in Switch	Development Concepts (primary)699 Research Concepts1000
ParentOf	Weakness Variant	546	Suspicious Comment	Development Concepts (primary)699 Research Concepts (primary)1000
ParentOf	Weakness Variant	547	Use of Hard-coded, Security-relevant Constants	Development Concepts (primary)699 Research Concepts (primary)1000
ParentOf	Weakness Variant	561	Dead Code	Development Concepts (primary)699 Research Concepts (primary)1000
ParentOf	Weakness Base	562	Return of Stack Variable Address	Development Concepts (primary)699 Research Concepts1000
ParentOf	Weakness Variant	563	Unused Variable	Development Concepts (primary)699 Research Concepts (primary)1000
ParentOf	Category	569	Expression Issues	Development Concepts (primary)699
ParentOf	Weakness Variant	585	Empty Synchronized Block	Development Concepts (primary)699 Research Concepts (primary)1000
ParentOf	Weakness Variant	586	Explicit Call to Finalize()	Development Concepts (primary)699
ParentOf	Weakness Variant	617	Reachable Assertion	Development Concepts (primary)699
ParentOf	Weakness Base	676	Use of Potentially Dangerous Function	Development Concepts (primary)699 Research Concepts (primary)1000
MemberOf	View	700	Seven Pernicious Kingdoms	Seven Pernicious Kingdoms (primary)700

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
----------------------	---------	-----	------------------

7 Pernicious Kingdoms			Code Quality
-----------------------	--	--	--------------

Content History

Submissions

Submission Date	Submitter	Organization	Source
	7 Pernicious Kingdoms		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci updated Time of Introduction	Cigital	External
2008-09-08	CWE Content Team updated Description, Relationships, Taxonomy Mappings	MITRE	Internal
2009-10-29	CWE Content Team updated Relationships	MITRE	Internal

Previous Entry Names

Change Date	Previous Entry Name
2008-04-11	Code Quality

[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 `ArrayIndexOutOfBoundsException` 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 cause 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 defaults 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, overridden 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 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