

vul_files_63 Scan Report

Project Name	vul_files_63
Scan Start	Wednesday, January 8, 2025 11:52:51 PM
Preset	Checkmarx Default
Scan Time	00h:24m:35s
Lines Of Code Scanned	299370
Files Scanned	60
Report Creation Time	Thursday, January 9, 2025 9:30:06 AM
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065
Team	CxServer
Checkmarx Version	8.7.0
Scan Type	Full
Source Origin	LocalPath
Density	1/1000 (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

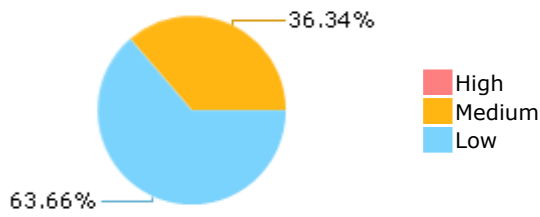
Results Limit

Results limit per query was set to 50

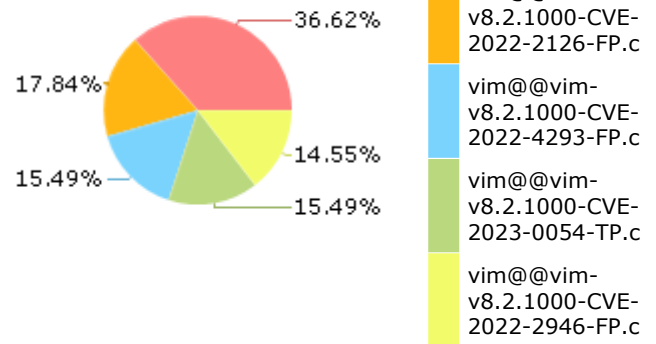
Selected Queries

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

Result Summary

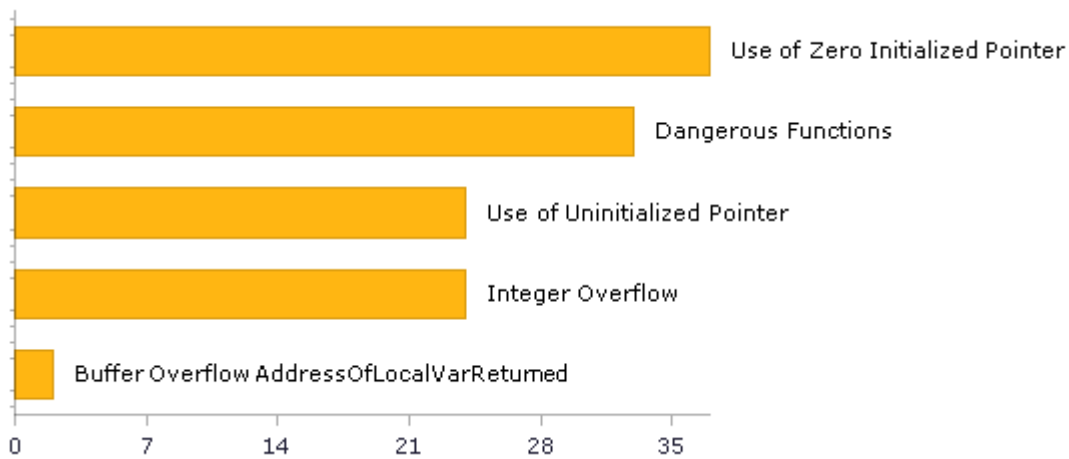


Most Vulnerable Files



- vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
- vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
- vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
- vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
- vim@@vim-v8.2.1000-CVE-2022-2946-FP.c

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	102	33
A2-Broken Authentication	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	36	36
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	33	33
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	33	33
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	4	4
PCI DSS (3.2) - 6.5.2 - Buffer overflows	26	26
PCI DSS (3.2) - 6.5.3 - Insecure cryptographic storage	0	0
PCI DSS (3.2) - 6.5.4 - Insecure communications	0	0
PCI DSS (3.2) - 6.5.5 - Improper error handling*	0	0
PCI DSS (3.2) - 6.5.7 - Cross-site scripting (XSS)	0	0
PCI DSS (3.2) - 6.5.8 - Improper access control	0	0
PCI DSS (3.2) - 6.5.9 - Cross-site request forgery	0	0
PCI DSS (3.2) - 6.5.10 - Broken authentication and session management	0	0

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

Scan Summary - FISMA 2014

Category	Description	Issues Found	Best Fix Locations
Access Control	Organizations must limit information system access to authorized users, processes acting on behalf of authorized users, or devices (including other information systems) and to the types of transactions and functions that authorized users are permitted to exercise.	2	2
Audit And Accountability*	Organizations must: (i) create, protect, and retain information system audit records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful, unauthorized, or inappropriate information system activity; and (ii) ensure that the actions of individual information system users can be uniquely traced to those users so they can be held accountable for their actions.	8	8
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.	34	34
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.	24	24

* 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)	36	36
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)*	152	53
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*	66	66
SI-11 Error Handling (P2)*	20	20
SI-15 Information Output Filtering (P0)	0	0
SI-16 Memory Protection (P1)	4	4

* 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

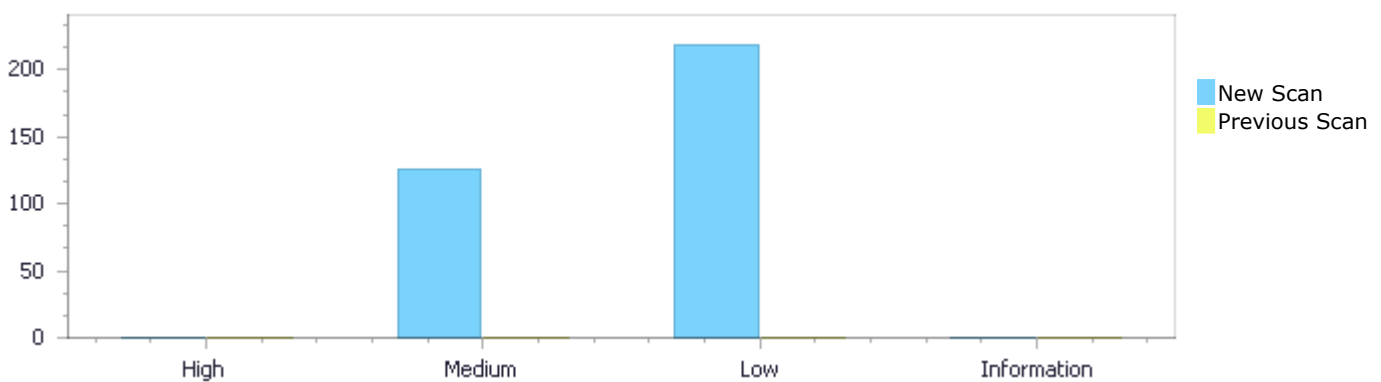
Scan Summary - Custom

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	125	219	0	344
Recurrent Issues	0	0	0	0	0
Total	0	125	219	0	344

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	125	219	0	344
Urgent	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0
Total	0	125	219	0	344

Result Summary

Vulnerability Type	Occurrences	Severity
Use of Zero Initialized Pointer	37	Medium
Dangerous Functions	33	Medium
Integer Overflow	24	Medium
Use of Uninitialized Pointer	24	Medium
Buffer Overflow AddressOfLocalVarReturned	2	Medium

Divide By Zero	2	Medium
Use of Uninitialized Variable	2	Medium
Memory Leak	1	Medium
NULL Pointer Dereference	78	Low
Improper Resource Access Authorization	34	Low
Use of Sizeof On a Pointer Type	29	Low
Unchecked Array Index	24	Low
Unchecked Return Value	20	Low
Potential Precision Problem	18	Low
Arithmetic Operation On Boolean	8	Low
Potential Off by One Error in Loops	4	Low
Incorrect Permission Assignment For Critical Resources	2	Low
TOCTOU	2	Low

10 Most Vulnerable Files

High and Medium Vulnerabilities

File Name	Issues Found
vim@@vim-v8.2.1000-CVE-2022-2946-FP.c	19
vim@@vim-v8.2.1000-CVE-2022-2126-FP.c	12
vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	10
vim@@vim-v8.2.1000-CVE-2022-1851-FP.c	9
vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	8
vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	8
vim@@vim-v8.2.1000-CVE-2022-2125-FP.c	7
vim@@vim-v8.2.1000-CVE-2022-2183-FP.c	7
vim@@vim-v8.2.1000-CVE-2022-2208-FP.c	7
vim@@vim-v8.2.1000-CVE-2022-2210-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=1030076&projectid=30065&pathid=107
Status	New

The variable declared in next_leader_flags at vim@@vim-v8.2.1000-CVE-2022-1851-FP.c in line 931 is not initialized when it is used by flags at vim@@vim-v8.2.1000-CVE-2022-1851-FP.c in line 454.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-1851-FP.c	vim@@vim-v8.2.1000-CVE-2022-1851-FP.c
Line	999	472
Object	next_leader_flags	flags

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-1851-FP.c
Method format_lines(

```
....  
999.         next_leader_flags = NULL;
```



File Name vim@@vim-v8.2.1000-CVE-2022-1851-FP.c
Method fmt_check_par(

```
....  
472.         flags = *leader_flags;
```

Use of Zero Initialized Pointer\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=108
Status	New

The variable declared in `leader_flags` at `vim@@vim-v8.2.1000-CVE-2022-1851-FP.c` in line 931 is not initialized when it is used by `flags` at `vim@@vim-v8.2.1000-CVE-2022-1851-FP.c` in line 454.

	Source	Destination
File	<code>vim@@vim-v8.2.1000-CVE-2022-1851-FP.c</code>	<code>vim@@vim-v8.2.1000-CVE-2022-1851-FP.c</code>
Line	943	472
Object	<code>leader_flags</code>	<code>flags</code>

Code Snippet

File Name `vim@@vim-v8.2.1000-CVE-2022-1851-FP.c`

Method `format_lines(`

```
.....
943.      char_u  *leader_flags = NULL;    // flags for leader of current
line
```

File Name `vim@@vim-v8.2.1000-CVE-2022-1851-FP.c`

Method `fmt_check_par(`

```
.....
472.      flags = *leader_flags;
```

Use of Zero Initialized Pointer\Path 3:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=109>

Status New

The variable declared in `new_vts_array` at `vim@@vim-v8.2.1000-CVE-2022-2125-FP.c` in line 1528 is not initialized when it is used by `new_line` at `vim@@vim-v8.2.1000-CVE-2022-2125-FP.c` in line 1528.

	Source	Destination
File	<code>vim@@vim-v8.2.1000-CVE-2022-2125-FP.c</code>	<code>vim@@vim-v8.2.1000-CVE-2022-2125-FP.c</code>
Line	1544	1651
Object	<code>new_vts_array</code>	<code>new_line</code>

Code Snippet

File Name `vim@@vim-v8.2.1000-CVE-2022-2125-FP.c`

Method `ex_retab(exarg_T *eap)`

```

.....
1544.      int          *new_vts_array = NULL;
.....
1651.          new_line = alloc(old_len - col + start_col + len
+ 1);

```

Use of Zero Initialized Pointer\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=110
Status	New

The variable declared in Pointer at vim@@vim-v8.2.1000-CVE-2022-2125-FP.c in line 23 is not initialized when it is used by new_line at vim@@vim-v8.2.1000-CVE-2022-2125-FP.c in line 1528.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2125-FP.c	vim@@vim-v8.2.1000-CVE-2022-2125-FP.c
Line	31	1651
Object	Pointer	new_line

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2125-FP.c
Method tabstop_set(char_u *var, int **array)

```

.....
31.      *array = NULL;

```

File Name vim@@vim-v8.2.1000-CVE-2022-2125-FP.c
Method ex_retab(exarg_T *eap)

```

.....
1651.          new_line = alloc(old_len - col + start_col + len
+ 1);

```

Use of Zero Initialized Pointer\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=111
Status	New

The variable declared in new_vts_array at vim@@vim-v8.2.1000-CVE-2022-2125-FP.c in line 1528 is not initialized when it is used by new_line at vim@@vim-v8.2.1000-CVE-2022-2125-FP.c in line 1528.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2125-FP.c	vim@@vim-v8.2.1000-CVE-2022-2125-FP.c
Line	1544	1658
Object	new_vts_array	new_line

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2125-FP.c
Method ex_retab(exarg_T *eap)

```
....
1544.      int      *new_vts_array = NULL;
....
1658.      ptr = new_line + start_col;
```

Use of Zero Initialized Pointer\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=112
Status	New

The variable declared in Pointer at vim@@vim-v8.2.1000-CVE-2022-2125-FP.c in line 23 is not initialized when it is used by new_line at vim@@vim-v8.2.1000-CVE-2022-2125-FP.c in line 1528.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2125-FP.c	vim@@vim-v8.2.1000-CVE-2022-2125-FP.c
Line	31	1658
Object	Pointer	new_line

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2125-FP.c
Method tabstop_set(char_u *var, int **array)

```
....
31.      *array = NULL;
```



File Name vim@@vim-v8.2.1000-CVE-2022-2125-FP.c
Method ex_retab(exarg_T *eap)

```
....
1658.      ptr = new_line + start_col;
```

Use of Zero Initialized Pointer\Path 7:

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

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=113
Status	New

The variable declared in new_vts_array at vim@@vim-v8.2.1000-CVE-2022-2183-FP.c in line 1528 is not initialized when it is used by new_line at vim@@vim-v8.2.1000-CVE-2022-2183-FP.c in line 1528.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2183-FP.c	vim@@vim-v8.2.1000-CVE-2022-2183-FP.c
Line	1544	1651
Object	new_vts_array	new_line

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2183-FP.c
Method ex_retab(exarg_T *eap)

```
....  
1544.      int      *new_vts_array = NULL;  
....  
1651.      new_line = alloc(old_len - col + start_col + len  
+ 1);
```

Use of Zero Initialized Pointer\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=114
Status	New

The variable declared in Pointer at vim@@vim-v8.2.1000-CVE-2022-2183-FP.c in line 23 is not initialized when it is used by new_line at vim@@vim-v8.2.1000-CVE-2022-2183-FP.c in line 1528.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2183-FP.c	vim@@vim-v8.2.1000-CVE-2022-2183-FP.c
Line	31	1651
Object	Pointer	new_line

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2183-FP.c
Method tabstop_set(char_u *var, int **array)

```
....  
31.      *array = NULL;
```

File Name vim@@vim-v8.2.1000-CVE-2022-2183-FP.c

Method ex_retab(exarg_T *eap)

```
....  
1651.                new_line = alloc(old_len - col + start_col + len  
+ 1);
```

Use of Zero Initialized Pointer\Path 9:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=115>

Status New

The variable declared in new_vts_array at vim@@vim-v8.2.1000-CVE-2022-2183-FP.c in line 1528 is not initialized when it is used by new_line at vim@@vim-v8.2.1000-CVE-2022-2183-FP.c in line 1528.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2183-FP.c	vim@@vim-v8.2.1000-CVE-2022-2183-FP.c
Line	1544	1658
Object	new_vts_array	new_line

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2183-FP.c

Method ex_retab(exarg_T *eap)

```
....  
1544.      int      *new_vts_array = NULL;  
....  
1658.      ptr = new_line + start_col;
```

Use of Zero Initialized Pointer\Path 10:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=116>

Status New

The variable declared in Pointer at vim@@vim-v8.2.1000-CVE-2022-2183-FP.c in line 23 is not initialized when it is used by new_line at vim@@vim-v8.2.1000-CVE-2022-2183-FP.c in line 1528.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2183-FP.c	vim@@vim-v8.2.1000-CVE-2022-2183-FP.c
Line	31	1658
Object	Pointer	new_line

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2183-FP.c
Method tabstop_set(char_u *var, int **array)

```
....
31.      *array = NULL;
```

File Name vim@@vim-v8.2.1000-CVE-2022-2183-FP.c
Method ex_retab(exarg_T *eap)

```
....
1658.                                ptr = new_line + start_col;
```

Use of Zero Initialized Pointer\Path 11:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=117>
Status New

The variable declared in saved_line at vim@@vim-v8.2.1000-CVE-2022-2207-FP.c in line 4838 is not initialized when it is used by saved_line at vim@@vim-v8.2.1000-CVE-2022-2207-FP.c in line 4838.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2207-FP.c	vim@@vim-v8.2.1000-CVE-2022-2207-FP.c
Line	4939	5091
Object	saved_line	saved_line

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2207-FP.c
Method ins_tab(void)

```
....
4939.      char_u      *saved_line = NULL;      // init for GCC
....
5091.      ins_bytes_len(saved_line + change_col,
```

Use of Zero Initialized Pointer\Path 12:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=118>
Status New

The variable declared in saved_line at vim@@vim-v8.2.1000-CVE-2022-2845-FP.c in line 4838 is not initialized when it is used by saved_line at vim@@vim-v8.2.1000-CVE-2022-2845-FP.c in line 4838.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2845-FP.c	vim@@vim-v8.2.1000-CVE-2022-2845-FP.c
Line	4939	5091
Object	saved_line	saved_line

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2845-FP.c

Method ins_tab(void)

```
....
4939.      char_u      *saved_line = NULL;      // init for GCC
....
5091.      ins_bytes_len(saved_line + change_col,
```

Use of Zero Initialized Pointer\Path 13:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=119>

Status New

The variable declared in matches at vim@@vim-v8.2.1000-CVE-2022-2946-FP.c in line 132 is not initialized when it is used by matches at vim@@vim-v8.2.1000-CVE-2022-2946-FP.c in line 132.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	168	580
Object	matches	matches

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c

Method do_tag(

```
....
168.      static char_u **matches = NULL;
....
580.      parse_match(matches[j], &tagp);
```

Use of Zero Initialized Pointer\Path 14:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=120>

Status New

The variable declared in `curwin_save` at `vim@@vim-v8.2.1000-CVE-2022-2946-FP.c` in line 3308 is not initialized when it is used by `curwin_save` at `vim@@vim-v8.2.1000-CVE-2022-2946-FP.c` in line 3308.

	Source	Destination
File	<code>vim@@vim-v8.2.1000-CVE-2022-2946-FP.c</code>	<code>vim@@vim-v8.2.1000-CVE-2022-2946-FP.c</code>
Line	3330	3484
Object	<code>curwin_save</code>	<code>curwin_save</code>

Code Snippet

File Name `vim@@vim-v8.2.1000-CVE-2022-2946-FP.c`

Method `jumpto_tag(`

```
....  
3330.      win_T  *curwin_save = NULL;  
....  
3484.      keep_help_flag = bt_help(curwin_save->w_buffer);
```

Use of Zero Initialized Pointer\Path 15:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=121>

Status New

The variable declared in `ht_stack` at `vim@@vim-v8.2.1000-CVE-2022-4293-FP.c` in line 3622 is not initialized when it is used by `ht_stack` at `vim@@vim-v8.2.1000-CVE-2022-4293-FP.c` in line 3622.

	Source	Destination
File	<code>vim@@vim-v8.2.1000-CVE-2022-4293-FP.c</code>	<code>vim@@vim-v8.2.1000-CVE-2022-4293-FP.c</code>
Line	3628	3655
Object	<code>ht_stack</code>	<code>ht_stack</code>

Code Snippet

File Name `vim@@vim-v8.2.1000-CVE-2022-4293-FP.c`

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

```
....  
3628.      ht_stack_T  *ht_stack = NULL;  
....  
3655.      ht_stack = ht_stack->prev;
```

Use of Zero Initialized Pointer\Path 16:

Severity Medium

Result State To Verify

Online Results <http://WIN->

PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=122

Status New

The variable declared in ht_stack at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3622 is not initialized when it is used by ht_stack at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3622.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3628	3653
Object	ht_stack	ht_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c

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

```
....  
3628.      ht_stack_T      *ht_stack = NULL;  
....  
3653.      cur_ht = ht_stack->ht;
```

Use of Zero Initialized Pointer\Path 17:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=123>

Status New

The variable declared in list_stack at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3699 is not initialized when it is used by list_stack at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3699.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3704	3723
Object	list_stack	list_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c

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

```
....  
3704.      list_stack_T *list_stack = NULL;  
....  
3723.      list_stack = list_stack->prev;
```

Use of Zero Initialized Pointer\Path 18:

Severity Medium

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=124
Status	New

The variable declared in list_stack at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3699 is not initialized when it is used by list_stack at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3699.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3704	3721
Object	list_stack	list_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....  
3704.      list_stack_T *list_stack = NULL;  
....  
3721.      cur_l = list_stack->list;
```

Use of Zero Initialized Pointer\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=125
Status	New

The variable declared in ht_stack at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3622 is not initialized when it is used by ht_stack at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3622.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3628	3655
Object	ht_stack	ht_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....  
3628.      ht_stack_T *ht_stack = NULL;  
....  
3655.      ht_stack = ht_stack->prev;
```


Use of Zero Initialized Pointer\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=126
Status	New

The variable declared in ht_stack at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3622 is not initialized when it is used by ht_stack at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3622.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3628	3653
Object	ht_stack	ht_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....  
3628.      ht_stack_T    *ht_stack = NULL;  
....  
3653.      cur_ht = ht_stack->ht;
```

Use of Zero Initialized Pointer\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=127
Status	New

The variable declared in list_stack at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3699 is not initialized when it is used by list_stack at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3699.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3704	3723
Object	list_stack	list_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....  
3704.      list_stack_T *list_stack = NULL;  
....  
3723.      list_stack = list_stack->prev;
```

Use of Zero Initialized Pointer\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=128
Status	New

The variable declared in list_stack at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3699 is not initialized when it is used by list_stack at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3699.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3704	3721
Object	list_stack	list_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....  
3704.      list_stack_T *list_stack = NULL;  
....  
3721.      cur_l = list_stack->list;
```

Use of Zero Initialized Pointer\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=129
Status	New

The variable declared in next_leader_flags at vim@@vim-v8.2.1000-CVE-2022-1851-FP.c in line 931 is not initialized when it is used by leader_flags at vim@@vim-v8.2.1000-CVE-2022-1851-FP.c in line 931.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-1851-FP.c	vim@@vim-v8.2.1000-CVE-2022-1851-FP.c
Line	999	991
Object	next_leader_flags	leader_flags

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-1851-FP.c
Method format_lines(

```
....  
999.          next_leader_flags = NULL;  
....  
991.          leader_flags = next_leader_flags;
```

Use of Zero Initialized Pointer\Path 24:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=130>
Status New

The variable declared in bl at vim@@vim-v8.2.1000-CVE-2022-2923-FP.c in line 4278 is not initialized when it is used by bl at vim@@vim-v8.2.1000-CVE-2022-2923-FP.c in line 4278.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	4295	4314
Object	bl	bl

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method getroom(

```
....  
4295.          bl = NULL;  
....  
4314.          p = bl->sb_data + bl->sb_used;
```

Use of Zero Initialized Pointer\Path 25:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=131>
Status New

The variable declared in matches at vim@@vim-v8.2.1000-CVE-2022-2946-FP.c in line 1546 is not initialized when it is used by matches at vim@@vim-v8.2.1000-CVE-2022-2946-FP.c in line 132.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	2781	597
Object	matches	matches

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Method find_tags(

```
....  
2781.         matches = NULL;
```

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Method do_tag(

```
....  
597.         matches = new_matches;
```

Use of Zero Initialized Pointer\Path 26:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=132>
Status New

The variable declared in matches at vim@@vim-v8.2.1000-CVE-2022-2946-FP.c in line 1546 is not initialized when it is used by matches at vim@@vim-v8.2.1000-CVE-2022-2946-FP.c in line 3966.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	2781	3981
Object	matches	matches

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Method find_tags(

```
....  
2781.         matches = NULL;
```

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Method get_tags(list_T *list, char_u *pat, char_u *buf_fname)

```
....  
3981.         parse_match(matches[i], &tp);
```

Use of Zero Initialized Pointer\Path 27:

Severity Medium
Result State To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=133
Status	New

The variable declared in `tn_search_ctx` at `vim@@vim-v8.2.1000-CVE-2022-2946-FP.c` in line 2866 is not initialized when it is used by `fname` at `vim@@vim-v8.2.1000-CVE-2022-2946-FP.c` in line 2866.

	Source	Destination
File	<code>vim@@vim-v8.2.1000-CVE-2022-2946-FP.c</code>	<code>vim@@vim-v8.2.1000-CVE-2022-2946-FP.c</code>
Line	2965	2951
Object	<code>tn_search_ctx</code>	<code>fname</code>

Code Snippet

File Name `vim@@vim-v8.2.1000-CVE-2022-2946-FP.c`
Method `get_tagfname(`

```
....  
2965.          tnp->tn_search_ctx = NULL;  
....  
2951.          fname = vim_findfile(tnp->tn_search_ctx);
```

Use of Zero Initialized Pointer\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=134
Status	New

The variable declared in `tn_search_ctx` at `vim@@vim-v8.2.1000-CVE-2022-2946-FP.c` in line 3005 is not initialized when it is used by `fname` at `vim@@vim-v8.2.1000-CVE-2022-2946-FP.c` in line 2866.

	Source	Destination
File	<code>vim@@vim-v8.2.1000-CVE-2022-2946-FP.c</code>	<code>vim@@vim-v8.2.1000-CVE-2022-2946-FP.c</code>
Line	3009	2951
Object	<code>tn_search_ctx</code>	<code>fname</code>

Code Snippet

File Name `vim@@vim-v8.2.1000-CVE-2022-2946-FP.c`
Method `tagname_free(tagname_T *tnp)`

```
....  
3009.          tnp->tn_search_ctx = NULL;
```

File Name `vim@@vim-v8.2.1000-CVE-2022-2946-FP.c`

Method get_tagfname(

```
....  
2951.          fname = vim_findfile(tnp->tn_search_ctx);
```

Use of Zero Initialized Pointer\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=135
Status	New

The variable declared in tn_search_ctx at vim@@vim-v8.2.1000-CVE-2022-2946-FP.c in line 2866 is not initialized when it is used by tn_search_ctx at vim@@vim-v8.2.1000-CVE-2022-2946-FP.c in line 2866.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	2965	2986
Object	tn_search_ctx	tn_search_ctx

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Method get_tagfname(

```
....  
2965.          tnp->tn_search_ctx = NULL;  
....  
2986.          tnp->tn_search_ctx = vim_findfile_init(buf, filename,
```

Use of Zero Initialized Pointer\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=136
Status	New

The variable declared in tn_search_ctx at vim@@vim-v8.2.1000-CVE-2022-2946-FP.c in line 3005 is not initialized when it is used by tn_search_ctx at vim@@vim-v8.2.1000-CVE-2022-2946-FP.c in line 2866.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	3009	2986
Object	tn_search_ctx	tn_search_ctx

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Method tagname_free(tagname_T *tnp)

```
....  
3009.         tnp->tn_search_ctx = NULL;
```

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Method get_tagfname(
↓

```
....  
2986.         tnp->tn_search_ctx = vim_findfile_init(buf, filename,
```

Use of Zero Initialized Pointer\Path 31:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=137>
Status New

The variable declared in fname_end at vim@@vim-v8.2.1000-CVE-2022-2946-FP.c in line 3308 is not initialized when it is used by fname_end at vim@@vim-v8.2.1000-CVE-2022-2946-FP.c in line 3308.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	3351	3356
Object	fname_end	fname_end

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Method jumpto_tag(
↓

```
....  
3351.         tagp.fname_end = NULL;  
....  
3356.         *tagp.fname_end = NUL;
```

Use of Zero Initialized Pointer\Path 32:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=138>
Status New

The variable declared in retval at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 361 is not initialized when it is used by retval at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 4628.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	374	4649
Object	retval	retval

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method eval_to_string(

```
....  
374.         retval = NULL;
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method make_expanded_name(

```
....  
4649.         retval = alloc(strlen(temp_result) + (expr_start - in_start)
```

Use of Zero Initialized Pointer\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=139
Status	New

The variable declared in Pointer at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 4529 is not initialized when it is used by retval at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 4628.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	4543	4649
Object	Pointer	retval

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method find_name_end(

```
....  
4543.         *expr_start = NULL;
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method make_expanded_name(


```
.....
4649.          retval = alloc(strlen(temp_result) + (expr_start - in_start))
```

Use of Zero Initialized Pointer\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=140
Status	New

The variable declared in Pointer at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3924 is not initialized when it is used by r at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3924.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	4091	4096
Object	Pointer	r

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method echo_string_core(

```
.....
4091.          *torelease = NULL;
.....
4096.          r = *torelease;
```

Use of Zero Initialized Pointer\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=141
Status	New

The variable declared in retval at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 361 is not initialized when it is used by retval at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 4628.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	374	4649
Object	retval	retval

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method eval_to_string(

```
....
374.         retval = NULL;
```

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method make_expanded_name(

```
....
4649.         retval = alloc(strlen(temp_result) + (expr_start - in_start))
```

Use of Zero Initialized Pointer\Path 36:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=142>

Status New

The variable declared in Pointer at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 4529 is not initialized when it is used by retval at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 4628.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	4543	4649
Object	Pointer	retval

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method find_name_end(

```
....
4543.         *expr_start = NULL;
```

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method make_expanded_name(

```
....
4649.         retval = alloc(strlen(temp_result) + (expr_start - in_start))
```

Use of Zero Initialized Pointer\Path 37:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=143>

Status New

The variable declared in Pointer at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3924 is not initialized when it is used by r at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3924.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	4091	4096
Object	Pointer	r

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method echo_string_core(

```

....
4091.          *torelease = NULL;
....
4096.          r = *torelease;

```

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=1030076&projectid=30065&pathid=47
Status	New

The dangerous function, sprintf, was found in use at line 1234 in vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	1362	1362
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
.....
1362.                                sprintf(changename[depth], "prefix");
```

Dangerous Functions\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=48
Status	New

The dangerous function, sprintf, was found in use at line 1234 in vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	1760	1760
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
.....
1760.                                sprintf(changename[depth], "%.s-%s: split",
```

Dangerous Functions\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=49
Status	New

The dangerous function, sprintf, was found in use at line 1234 in vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	1763	1763
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
....
1763.                                sprintf(changename[depth], "%. *s-%s:
compound",
```

Dangerous Functions\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=50
Status	New

The dangerous function, sprintf, was found in use at line 1234 in vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	1909	1909
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
....
1909.                                sprintf(changename[depth], "%. *s-%s: subst %c to
%c",
```

Dangerous Functions\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=51
Status	New

The dangerous function, sprintf, was found in use at line 1234 in vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	1913	1913
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c

Method suggest_trie_walk(

```
....  
1913.                sprintf(changename[depth], "%.s-%s: accept %c",
```

Dangerous Functions\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=52
Status	New

The dangerous function, sprintf, was found in use at line 1234 in vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2048	2048
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
....  
2048.                sprintf(changename[depth], "%.s-%s: delete %c",
```

Dangerous Functions\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=53
Status	New

The dangerous function, sprintf, was found in use at line 1234 in vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2145	2145
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
.....  
2145.          sprintf(changename[depth], "%.s-%s: insert %c",
```

Dangerous Functions\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=54
Status	New

The dangerous function, sprintf, was found in use at line 1234 in vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2245	2245
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
.....  
2245.          sprintf(changename[depth], "%.s-%s: swap %c and %c",
```

Dangerous Functions\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=55
Status	New

The dangerous function, sprintf, was found in use at line 1234 in vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2333	2333
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
.....
2333.                sprintf(changename[depth], "%.s-%s: swap3 %c and %c",
```

Dangerous Functions\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=56
Status	New

The dangerous function, sprintf, was found in use at line 1234 in vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2401	2401
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
.....
2401.                sprintf(changename[depth], "%.s-%s: rotate left
%c%c%c",
```

Dangerous Functions\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=57
Status	New

The dangerous function, sprintf, was found in use at line 1234 in vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2462	2462
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(


```
.....
2462.                sprintf(changename[depth], "%.s-%s: rotate right
%c%c%c",
```

Dangerous Functions\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=58
Status	New

The dangerous function, sprintf, was found in use at line 1234 in vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2574	2574
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
.....
2574.                sprintf(changename[depth], "%.s-%s: replace %s
with %s",
```

Dangerous Functions\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=59
Status	New

The dangerous function, sprintf, was found in use at line 722 in vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2816-FP.c	vim@@vim-v8.2.1000-CVE-2022-2816-FP.c
Line	762	762
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2816-FP.c

Method ex_compiler(exarg_T *eap)

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

Dangerous Functions\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=60
Status	New

The dangerous function, sprintf, was found in use at line 2175 in vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	2735	2735
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
....  
2735.          sprintf((char *)buf, "^%s", items[4]);
```

Dangerous Functions\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=61
Status	New

The dangerous function, sprintf, was found in use at line 2175 in vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	2737	2737
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
.....
2737.                                sprintf((char *)buf, "%s$", items[4]);
```

Dangerous Functions\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=62
Status	New

The dangerous function, sprintf, was found in use at line 2175 in vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	2796	2796
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
.....
2796.                                sprintf((char *)buf, "^%s",
```

Dangerous Functions\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=63
Status	New

The dangerous function, sprintf, was found in use at line 132 in vim@@vim-v8.2.1000-CVE-2022-2946-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-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	720	720
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Method do_tag(

```
.....
720.                sprintf((char *)IObuff, _("tag %d of %d%s"),
```

Dangerous Functions\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=64
Status	New

The dangerous function, sprintf, was found in use at line 1546 in vim@@vim-v8.2.1000-CVE-2022-2946-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-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	2545	2545
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Method find_tags(

```
.....
2545.                sprintf((char *)p + len + 1 + ML_EXTRA, "%06d",
```

Dangerous Functions\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=65
Status	New

The dangerous function, sprintf, was found in use at line 3308 in vim@@vim-v8.2.1000-CVE-2022-2946-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-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	3571	3571
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Method jumpto_tag(

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=66
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	3576	3576
Object	sprintf	sprintf

```

3576.          sprintf((char *)pbuf, "%s\\s\\*",
Z ]\\.\\*\\<%s\\s\\*(",

```

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=67
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2816-FP.c	vim@@vim-v8.2.1000-CVE-2022-2816-FP.c
Line	1395	1395
Object	strtok	strtok

PAGE 45 OF 230

```
.....  
1395.         loc = (char_u *)strtok((char *)locale_list, "\n");
```

Dangerous Functions\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=68
Status	New

The dangerous function, strtok, was found in use at line 1345 in vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2816-FP.c	vim@@vim-v8.2.1000-CVE-2022-2816-FP.c
Line	1419	1419
Object	strtok	strtok

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2816-FP.c
Method find_locales(void)

```
.....  
1419.         loc = (char_u *)strtok(NULL, "\n");
```

Dangerous Functions\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=69
Status	New

The dangerous function, atoi, was found in use at line 23 in vim@@vim-v8.2.1000-CVE-2022-2125-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-v8.2.1000-CVE-2022-2125-FP.c	vim@@vim-v8.2.1000-CVE-2022-2125-FP.c
Line	70	70
Object	atoi	atoi

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2125-FP.c
Method tabstop_set(char_u *var, int **array)

```
....  
70.    (*array)[t++] = atoi((char *)cp);
```

Dangerous Functions\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=70
Status	New

The dangerous function, atoi, was found in use at line 23 in vim@@vim-v8.2.1000-CVE-2022-2183-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-v8.2.1000-CVE-2022-2183-FP.c	vim@@vim-v8.2.1000-CVE-2022-2183-FP.c
Line	70	70
Object	atoi	atoi

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2183-FP.c
Method tabstop_set(char_u *var, int **array)

```
....  
70.    (*array)[t++] = atoi((char *)cp);
```

Dangerous Functions\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=71
Status	New

The dangerous function, atoi, was found in use at line 2695 in vim@@vim-v8.2.1000-CVE-2022-2264-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-v8.2.1000-CVE-2022-2264-FP.c	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Line	2724	2724
Object	atoi	atoi

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Method write_reg_contents_ex(

```
.....  
2724.                int      num = atoi((char *)str);
```

Dangerous Functions\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=72
Status	New

The dangerous function, atoi, was found in use at line 2175 in vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	2462	2462
Object	atoi	atoi

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
.....  
2462.                if (atoi((char *)items[1]) == 0)
```

Dangerous Functions\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=73
Status	New

The dangerous function, atoi, was found in use at line 2175 in vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	2492	2492
Object	atoi	atoi

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)


```
.....
2492.                                compmax = atoi((char *)items[1]);
```

Dangerous Functions\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=74
Status	New

The dangerous function, atoi, was found in use at line 2175 in vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	2500	2500
Object	atoi	atoi

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
.....
2500.                                compminlen = atoi((char *)items[1]);
```

Dangerous Functions\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=75
Status	New

The dangerous function, atoi, was found in use at line 2175 in vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	2508	2508
Object	atoi	atoi

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
.....  
2508.                compsymmax = atoi((char *)items[1]);
```

Dangerous Functions\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=76
Status	New

The dangerous function, atoi, was found in use at line 2175 in vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	2531	2531
Object	atoi	atoi

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
.....  
2531.                if (atoi((char *)items[1]) == 0)
```

Dangerous Functions\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=77
Status	New

The dangerous function, atoi, was found in use at line 2175 in vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	2677	2677
Object	atoi	atoi

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
.....  
2677.                aff_todo = atoi((char *)items[3]);
```

Dangerous Functions\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=78
Status	New

The dangerous function, atoi, was found in use at line 3183 in vim@@vim-v8.2.1000-CVE-2022-2946-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-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	3241	3241
Object	atoi	atoi

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Method parse_match(

```
.....  
3241.                tagp->tagline = atoi((char *)p + 5);
```

Dangerous Functions\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=79
Status	New

The dangerous function, atoi, was found in use at line 2695 in vim@@vim-v8.2.1000-CVE-2022-3520-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-v8.2.1000-CVE-2022-3520-FP.c	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c
Line	2724	2724
Object	atoi	atoi

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-3520-FP.c
Method write_reg_contents_ex(

```
....  
2724.          int      num = atoi((char *)str);
```

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=1030076&projectid=30065&pathid=23
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 3086 of vim@@vim-v8.2.1000-CVE-2022-2207-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-v8.2.1000-CVE-2022-2207-FP.c	vim@@vim-v8.2.1000-CVE-2022-2207-FP.c
Line	3091	3091
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2207-FP.c
Method replace_join(

```
....  
3091.          for (i = replace_stack_nr; --i >= 0; )
```

Integer Overflow\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=24
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 277 of vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2208-FP.c	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Line	308	308
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Method diff_mark_adjust_tp(

```
....  
308.          inserted = amount;
```

Integer Overflow\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=25
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 277 of vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2208-FP.c	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Line	314	314
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Method diff_mark_adjust_tp(

```
....  
314.          inserted = amount_after;
```

Integer Overflow\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=26
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 277 of vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2208-FP.c	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Line	321	321
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Method diff_mark_adjust_tp(

```
....  
321.         deleted = -amount_after;
```

Integer Overflow\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=27
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 277 of vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2210-FP.c	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Line	308	308
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Method diff_mark_adjust_tp(

```
....  
308.         inserted = amount;
```

Integer Overflow\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=28
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 277 of vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2210-FP.c	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Line	314	314
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Method diff_mark_adjust_tp(

```
....  
314.          inserted = amount_after;
```

Integer Overflow\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=29
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 277 of vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2210-FP.c	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Line	321	321
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Method diff_mark_adjust_tp(

```
....  
321.          deleted = -amount_after;
```

Integer Overflow\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=30
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 2775 of vim@@vim-v8.2.1000-CVE-2022-2264-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-v8.2.1000-CVE-2022-2264-FP.c	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Line	2885	2885
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Method str_to_reg(

```
....  
2885.          extra += i;
```

Integer Overflow\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=31
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1488 of vim@@vim-v8.2.1000-CVE-2022-2264-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-v8.2.1000-CVE-2022-2264-FP.c	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Line	1840	1840
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Method do_put(

```
....  
1840.          totlen = count * (yanklen + spaces) + bd.startspaces +  
bd.endspaces;
```

Integer Overflow\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=32
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1488 of vim@@vim-v8.2.1000-CVE-2022-2264-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-v8.2.1000-CVE-2022-2264-FP.c	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Line	1953	1953
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Method do_put(

```
....  
1953.                totlen = count * yanklen;
```

Integer Overflow\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=33
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1488 of vim@@vim-v8.2.1000-CVE-2022-2264-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-v8.2.1000-CVE-2022-2264-FP.c	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Line	1833	1833
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Method do_put(

```
....  
1833.                spaces = y_width + 1;
```

Integer Overflow\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=34
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 277 of vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2289-FP.c	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Line	308	308
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Method diff_mark_adjust_tp(

```
....  
308.          inserted = amount;
```

Integer Overflow\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=35
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 277 of vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2289-FP.c	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Line	314	314
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Method diff_mark_adjust_tp(

```
....  
314.          inserted = amount_after;
```

Integer Overflow\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=36
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 277 of vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2289-FP.c	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Line	321	321
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Method diff_mark_adjust_tp(

```
....  
321.         deleted = -amount_after;
```

Integer Overflow\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=37
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 277 of vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2598-FP.c	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c
Line	308	308
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2598-FP.c
Method diff_mark_adjust_tp(

```
....  
308.         inserted = amount;
```

Integer Overflow\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=38
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 277 of vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2598-FP.c	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c
Line	314	314
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2598-FP.c
Method diff_mark_adjust_tp(

```
....  
314.          inserted = amount_after;
```

Integer Overflow\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=39
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 277 of vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2598-FP.c	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c
Line	321	321
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2598-FP.c
Method diff_mark_adjust_tp(

```
....  
321.          deleted = -amount_after;
```

Integer Overflow\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=40
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 3086 of vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2845-FP.c	vim@@vim-v8.2.1000-CVE-2022-2845-FP.c
Line	3091	3091
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2845-FP.c
Method replace_join(

```
....  
3091.         for (i = replace_stack_nr; --i >= 0; )
```

Integer Overflow\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=41
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1546 of vim@@vim-v8.2.1000-CVE-2022-2946-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-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	2600	2600
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Method find_tags(

```
....  
2600.         len = (int)tag_fname_len + (int)STRLEN(lbuf) + 3;
```

Integer Overflow\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=42
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1546 of vim@@vim-v8.2.1000-CVE-2022-2946-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-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	2605	2605
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Method find_tags(

```
....  
2605. len += (int)ebuf_len + 1;
```

Integer Overflow\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=43
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 2775 of vim@@vim-v8.2.1000-CVE-2022-3520-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-v8.2.1000-CVE-2022-3520-FP.c	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c
Line	2885	2885
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-3520-FP.c
Method str_to_reg(

```
....  
2885. extra += i;
```

Integer Overflow\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=44
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1488 of vim@@vim-v8.2.1000-CVE-2022-3520-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-v8.2.1000-CVE-2022-3520-FP.c	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c
Line	1840	1840
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-3520-FP.c
Method do_put(

```
....  
1840.          totlen = count * (yanklen + spaces) + bd.startspaces +  
bd.endspaces;
```

Integer Overflow\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=45
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1488 of vim@@vim-v8.2.1000-CVE-2022-3520-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-v8.2.1000-CVE-2022-3520-FP.c	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c
Line	1953	1953
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-3520-FP.c
Method do_put(

```
....  
1953.          totlen = count * yanklen;
```

Integer Overflow\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=46
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1488 of vim@@vim-v8.2.1000-CVE-2022-3520-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-v8.2.1000-CVE-2022-3520-FP.c	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c
Line	1833	1833
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-3520-FP.c
Method do_put(

```
....
1833.          spaces = y_width + 1;
```

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=1030076&projectid=30065&pathid=81
Status	New

The variable declared in wp at vim@@vim-v8.2.1000-CVE-2022-1851-FP.c in line 803 is not initialized when it is used by w_old_cursor_lnum at vim@@vim-v8.2.1000-CVE-2022-1851-FP.c in line 803.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-1851-FP.c	vim@@vim-v8.2.1000-CVE-2022-1851-FP.c
Line	854	862
Object	wp	w_old_cursor_lnum

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-1851-FP.c
Method op_format(

```
....
854.          win_T *wp;
....
862.          if (wp->w_old_cursor_lnum > wp->w_old_visual_lnum)
```

Use of Uninitialized Pointer\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=81

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=82
Status	New

The variable declared in wp at vim@@vim-v8.2.1000-CVE-2022-1851-FP.c in line 803 is not initialized when it is used by w_old_cursor_lnum at vim@@vim-v8.2.1000-CVE-2022-1851-FP.c in line 803.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-1851-FP.c	vim@@vim-v8.2.1000-CVE-2022-1851-FP.c
Line	854	858
Object	wp	w_old_cursor_lnum

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-1851-FP.c
Method op_format(

```
....  
854.         win_T *wp;  
....  
858.         if (wp->w_old_cursor_lnum != 0)
```

Use of Uninitialized Pointer\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=83
Status	New

The variable declared in wp at vim@@vim-v8.2.1000-CVE-2022-1851-FP.c in line 803 is not initialized when it is used by w_old_visual_lnum at vim@@vim-v8.2.1000-CVE-2022-1851-FP.c in line 803.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-1851-FP.c	vim@@vim-v8.2.1000-CVE-2022-1851-FP.c
Line	854	862
Object	wp	w_old_visual_lnum

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-1851-FP.c
Method op_format(

```
....  
854.         win_T *wp;  
....  
862.         if (wp->w_old_cursor_lnum > wp->w_old_visual_lnum)
```

Use of Uninitialized Pointer\Path 4:

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

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=84
Status	New

The variable declared in wp at vim@@vim-v8.2.1000-CVE-2022-1851-FP.c in line 803 is not initialized when it is used by w_old_cursor_lnum at vim@@vim-v8.2.1000-CVE-2022-1851-FP.c in line 803.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-1851-FP.c	vim@@vim-v8.2.1000-CVE-2022-1851-FP.c
Line	854	863
Object	wp	w_old_cursor_lnum

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-1851-FP.c
Method op_format(

```
....  
854.          win_T *wp;  
....  
863.          wp->w_old_cursor_lnum += old_line_count;
```

Use of Uninitialized Pointer\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=85
Status	New

The variable declared in wp at vim@@vim-v8.2.1000-CVE-2022-1851-FP.c in line 803 is not initialized when it is used by w_old_visual_lnum at vim@@vim-v8.2.1000-CVE-2022-1851-FP.c in line 803.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-1851-FP.c	vim@@vim-v8.2.1000-CVE-2022-1851-FP.c
Line	854	865
Object	wp	w_old_visual_lnum

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-1851-FP.c
Method op_format(

```
....  
854.          win_T *wp;  
....  
865.          wp->w_old_visual_lnum += old_line_count;
```

Use of Uninitialized Pointer\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=86
Status	New

The variable declared in wp at vim@@vim-v8.2.1000-CVE-2022-2208-FP.c in line 123 is not initialized when it is used by w_p_diff at vim@@vim-v8.2.1000-CVE-2022-2208-FP.c in line 123.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Line	125	133
Object	wp	w_p_diff

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Method diff_buf_adjust(win_T *win)

```
....  
125.      win_T      *wp;  
....  
133.      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=1030076&projectid=30065&pathid=87
Status	New

The variable declared in wp at vim@@vim-v8.2.1000-CVE-2022-2208-FP.c in line 123 is not initialized when it is used by w_buffer at vim@@vim-v8.2.1000-CVE-2022-2208-FP.c in line 123.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Line	125	133
Object	wp	w_buffer

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Method diff_buf_adjust(win_T *win)

```

.....
125.      win_T    *wp;
.....
133.      if (wp->w_buffer == win->w_buffer && wp->w_p_diff)

```

Use of Uninitialized Pointer\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=88
Status	New

The variable declared in wp at vim@@vim-v8.2.1000-CVE-2022-2208-FP.c in line 123 is not initialized when it is used by wp at vim@@vim-v8.2.1000-CVE-2022-2208-FP.c in line 123.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Line	125	135
Object	wp	wp

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Method diff_buf_adjust(win_T *win)

```

.....
125.      win_T    *wp;
.....
135.      if (wp == NULL)

```

Use of Uninitialized Pointer\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=89
Status	New

The variable declared in tp at vim@@vim-v8.2.1000-CVE-2022-2208-FP.c in line 2175 is not initialized when it is used by tp_diff_invalid at vim@@vim-v8.2.1000-CVE-2022-2208-FP.c in line 2175.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Line	2183	2301
Object	tp	tp_diff_invalid

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Method diffopt_changed(void)

```
....  
2183.         tabpage_T      *tp;  
....  
2301.         tp->tp_diff_invalid = TRUE;
```

Use of Uninitialized Pointer\Path 10:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=90>
Status New

The variable declared in wp at vim@@vim-v8.2.1000-CVE-2022-2210-FP.c in line 123 is not initialized when it is used by w_p_diff at vim@@vim-v8.2.1000-CVE-2022-2210-FP.c in line 133.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Line	125	133
Object	wp	w_p_diff

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Method diff_buf_adjust(win_T *win)

```
....  
125.         win_T      *wp;  
....  
133.         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=1030076&projectid=30065&pathid=91>
Status New

The variable declared in wp at vim@@vim-v8.2.1000-CVE-2022-2210-FP.c in line 123 is not initialized when it is used by w_buffer at vim@@vim-v8.2.1000-CVE-2022-2210-FP.c in line 133.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Line	125	133
Object	wp	w_buffer

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Method diff_buf_adjust(win_T *win)

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

Use of Uninitialized Pointer\Path 12:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=92>
Status New

The variable declared in wp at vim@@vim-v8.2.1000-CVE-2022-2210-FP.c in line 123 is not initialized when it is used by wp at vim@@vim-v8.2.1000-CVE-2022-2210-FP.c in line 123.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Line	125	135
Object	wp	wp

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Method diff_buf_adjust(win_T *win)

```
....  
125.      win_T      *wp;  
....  
135.      if (wp == NULL)
```

Use of Uninitialized Pointer\Path 13:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=93>
Status New

The variable declared in tp at vim@@vim-v8.2.1000-CVE-2022-2210-FP.c in line 2175 is not initialized when it is used by tp_diff_invalid at vim@@vim-v8.2.1000-CVE-2022-2210-FP.c in line 2175.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c

Line	2183	2301
Object	tp	tp_diff_invalid

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Method diffopt_changed(void)

```
....
2183.         tabpage_T      *tp;
....
2301.         tp->tp_diff_invalid = TRUE;
```

Use of Uninitialized Pointer\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=94
Status	New

The variable declared in wp at vim@@vim-v8.2.1000-CVE-2022-2289-FP.c in line 123 is not initialized when it is used by w_p_diff at vim@@vim-v8.2.1000-CVE-2022-2289-FP.c in line 133.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Line	125	133
Object	wp	w_p_diff

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Method diff_buf_adjust(win_T *win)

```
....
125.         win_T      *wp;
....
133.         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=1030076&projectid=30065&pathid=95
Status	New

The variable declared in wp at vim@@vim-v8.2.1000-CVE-2022-2289-FP.c in line 123 is not initialized when it is used by w_buffer at vim@@vim-v8.2.1000-CVE-2022-2289-FP.c in line 133.

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

File	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Line	125	133
Object	wp	w_buffer

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Method diff_buf_adjust(win_T *win)

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

Use of Uninitialized Pointer\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=96
Status	New

The variable declared in wp at vim@@vim-v8.2.1000-CVE-2022-2289-FP.c in line 123 is not initialized when it is used by wp at vim@@vim-v8.2.1000-CVE-2022-2289-FP.c in line 123.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Line	125	135
Object	wp	wp

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Method diff_buf_adjust(win_T *win)

```
....
125.      win_T      *wp;
....
135.          if (wp == NULL)
```

Use of Uninitialized Pointer\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=97
Status	New

The variable declared in tp at vim@@vim-v8.2.1000-CVE-2022-2289-FP.c in line 2175 is not initialized when it is used by tp_diff_invalid at vim@@vim-v8.2.1000-CVE-2022-2289-FP.c in line 2175.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Line	2183	2301
Object	tp	tp_diff_invalid

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Method diffopt_changed(void)

```
....
2183.      tabpage_T      *tp;
....
2301.      tp->tp_diff_invalid = TRUE;
```

Use of Uninitialized Pointer\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=98
Status	New

The variable declared in wp at vim@@vim-v8.2.1000-CVE-2022-2598-FP.c in line 123 is not initialized when it is used by w_p_diff at vim@@vim-v8.2.1000-CVE-2022-2598-FP.c in line 133.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c
Line	125	133
Object	wp	w_p_diff

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2598-FP.c
Method diff_buf_adjust(win_T *win)

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

Use of Uninitialized Pointer\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=99
Status	New

The variable declared in wp at vim@@vim-v8.2.1000-CVE-2022-2598-FP.c in line 123 is not initialized when it is used by w_buffer at vim@@vim-v8.2.1000-CVE-2022-2598-FP.c in line 123.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c
Line	125	133
Object	wp	w_buffer

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2598-FP.c

Method diff_buf_adjust(win_T *win)

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

Use of Uninitialized Pointer\Path 20:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=100>

Status New

The variable declared in wp at vim@@vim-v8.2.1000-CVE-2022-2598-FP.c in line 123 is not initialized when it is used by wp at vim@@vim-v8.2.1000-CVE-2022-2598-FP.c in line 123.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c
Line	125	135
Object	wp	wp

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2598-FP.c

Method diff_buf_adjust(win_T *win)

```
....  
125.      win_T      *wp;  
....  
135.      if (wp == NULL)
```

Use of Uninitialized Pointer\Path 21:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=101>

Status New

The variable declared in tp at vim@@vim-v8.2.1000-CVE-2022-2598-FP.c in line 2175 is not initialized when it is used by tp_diff_invalid at vim@@vim-v8.2.1000-CVE-2022-2598-FP.c in line 2301.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c
Line	2183	2301
Object	tp	tp_diff_invalid

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2598-FP.c

Method diffopt_changed(void)

```
....  
2183.      tabpage_T      *tp;  
....  
2301.      tp->tp_diff_invalid = TRUE;
```

Use of Uninitialized Pointer\Path 22:

Severity Medium

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=102>

Status New

The variable declared in li at vim@@vim-v8.2.1000-CVE-2022-2946-FP.c in line 4184 is not initialized when it is used by li_tv at vim@@vim-v8.2.1000-CVE-2022-2946-FP.c in line 4184.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	4186	4196
Object	li	li_tv

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c

Method tagstack_push_items(win_T *wp, list_T *l)

```
....  
4186.      listitem_T      *li;  
....  
4196.      if (li->li_tv.v_type != VAR_DICT || li->li_tv.vval.v_dict ==  
NULL)
```

Use of Uninitialized Pointer\Path 23:

Severity Medium

Result State To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=103
Status	New

The variable declared in li at vim@@vim-v8.2.1000-CVE-2022-2946-FP.c in line 4184 is not initialized when it is used by li_tv at vim@@vim-v8.2.1000-CVE-2022-2946-FP.c in line 4184.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	4186	4196
Object	li	li_tv

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Method tagstack_push_items(win_T *wp, list_T *l)

```

....
4186.      listitem_T    *li;
....
4196.      if (li->li_tv.v_type != VAR_DICT || li->li_tv.vval.v_dict ==
NULL)

```

Use of Uninitialized Pointer\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=104
Status	New

The variable declared in li at vim@@vim-v8.2.1000-CVE-2022-2946-FP.c in line 4184 is not initialized when it is used by li at vim@@vim-v8.2.1000-CVE-2022-2946-FP.c in line 4184.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	4186	4198
Object	li	li

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Method tagstack_push_items(win_T *wp, list_T *l)

```

....
4186.      listitem_T    *li;
....
4198.      itemdict = li->li_tv.vval.v_dict;

```

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=1030076&projectid=30065&pathid=1
Status	New

The application performs an illegal operation in `tabstop_padding`, in `vim@@vim-v8.2.1000-CVE-2022-2125-FP.c`. In line 86, the program attempts to divide by `ts`, 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 `ts` in `tabstop_padding` of `vim@@vim-v8.2.1000-CVE-2022-2125-FP.c`, at line 86.

	Source	Destination
File	<code>vim@@vim-v8.2.1000-CVE-2022-2125-FP.c</code>	<code>vim@@vim-v8.2.1000-CVE-2022-2125-FP.c</code>
Line	95	95
Object	<code>ts</code>	<code>ts</code>

Code Snippet

File Name `vim@@vim-v8.2.1000-CVE-2022-2125-FP.c`
Method `tabstop_padding(colnr_T col, int ts_arg, int *vts)`

```
....  
95.    return ts - (col % ts);
```

Divide By Zero\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=2
Status	New

The application performs an illegal operation in `tabstop_padding`, in `vim@@vim-v8.2.1000-CVE-2022-2183-FP.c`. In line 86, the program attempts to divide by `ts`, 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 `ts` in `tabstop_padding` of `vim@@vim-v8.2.1000-CVE-2022-2183-FP.c`, at line 86.

	Source	Destination
File	<code>vim@@vim-v8.2.1000-CVE-2022-2183-FP.c</code>	<code>vim@@vim-v8.2.1000-CVE-2022-2183-FP.c</code>
Line	95	95
Object	<code>ts</code>	<code>ts</code>

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2183-FP.c
Method tabstop_padding(colnr_T col, int ts_arg, int *vts)

```
....  
95.     return ts - (col % ts);
```

Buffer Overflow AddressOfLocalVarReturned

Query Path:

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

Categories

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

Description

Buffer Overflow AddressOfLocalVarReturned\Path 1:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=3>
Status New

The pointer VIsual at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 4224 is being used after it has been freed.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	4284	4284
Object	VIsual	VIsual

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method var2fpos(

```
....  
4284.         return &VIsual;
```

Buffer Overflow AddressOfLocalVarReturned\Path 2:

Severity Medium
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=4>
Status New

The pointer VIsual at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 4224 is being used after it has been freed.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	4284	4284
Object	VIsual	VIsual

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method var2fpos(

```
....  
4284.          return &VIsual;
```

Use of Uninitialized Variable

Query Path:

CPP\Cx\CPP Medium Threat\Use of Uninitialized Variable Version:0

Categories

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

Description

Use of Uninitialized Variable\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=105
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2125-FP.c	vim@@vim-v8.2.1000-CVE-2022-2125-FP.c
Line	1391	1520
Object	ind_len	ind_len

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2125-FP.c
Method copy_indent(int size, char_u *src)

```
....  
1391.      int      ind_len;  
....  
1520.      curwin->w_cursor.col = ind_len;
```

Use of Uninitialized Variable\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=105

Status	065&pathid=106 New
--------	---

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2183-FP.c	vim@@vim-v8.2.1000-CVE-2022-2183-FP.c
Line	1391	1520
Object	ind_len	ind_len

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2183-FP.c
Method copy_indent(int size, char_u *src)

```
....
1391.      int          ind_len;
....
1520.      curwin->w_cursor.col = ind_len;
```

Memory Leak

Query Path:

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

Categories

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

Description

Memory Leak\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=80
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-1851-FP.c	vim@@vim-v8.2.1000-CVE-2022-1851-FP.c
Line	627	627
Object	neW	neW

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-1851-FP.c
Method auto_format(

```
....
627.      char_u  *new, *pnew;
```

NULL Pointer Dereference

Query Path:

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=1030076&projectid=30065&pathid=197
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-2923-FP.c in line 5449 is not initialized when it is used by sl_map_hash at vim@@vim-v8.2.1000-CVE-2022-2923-FP.c in line 6599.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	5469	6654
Object	null	sl_map_hash

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method spell_make_sugfile(spellinfo_T *spin, char_u *wfname)

```
....
5469.         slang = spell_load_file(wfname, NULL, NULL, FALSE);
```



File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method set_map_str(slang_T *lp, char_u *map)

```
....
6654.         hash_add_item(&lp->sl_map_hash, hi, b, hash);
```

NULL Pointer Dereference\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=198
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-2923-FP.c in line 5449 is not initialized when it is used by sl_map_hash at vim@@vim-v8.2.1000-CVE-2022-2923-FP.c in line 6599.

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

File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	5469	6616
Object	null	sl_map_hash

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method spell_make_sugfile(spellinfo_T *spin, char_u *wfname)

```
....
5469.          slang = spell_load_file(wfname, NULL, NULL, FALSE);
```

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method set_map_str(slang_T *lp, char_u *map)

```
....
6616.          hash_init(&lp->sl_map_hash);
```

NULL Pointer Dereference\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=199
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by ht_stack at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3462	3766
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
3462.                                     NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
.....
3766.                                newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=200
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by ht_stack at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3467	3766
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
.....
3467.                                NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
.....
3766.                                newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=201
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by ht_stack at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c

Line	3470	3766
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
3470.                                     NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
....
3766.                                newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 6:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=202>
Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by ht_stack at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3474	3766
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
3474.                                     NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
....
3766.                                newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=203
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by ht_stack at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3478	3766
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
3478.                                     NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
....
3766.                                     newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=204
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by ht_stack at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3484	3766
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c

Method garbage_collect(int testing)

```
.....  
3484.                                     NULL, NULL);
```



File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
.....  
3766.                               newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 9:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=205>

Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3467	3766
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c

Method garbage_collect(int testing)

```
.....  
3467.                                     NULL, NULL);
```



File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
.....  
3766.                               newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 10:

Severity Low

Result State To Verify

Online Results <http://WIN->

PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=206

Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3462	3766
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
3462.                                     NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
....
3766.                                newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 11:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=207>
Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3470	3766
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
.....
3470.                                     NULL, NULL) ;
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
.....
3766.                                newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 12:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=208>
Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3484	3766
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
.....
3484.                                     NULL, NULL) ;
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
.....
3766.                                newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 13:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=209>
Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3474	3766
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
3474.                                     NULL, NULL) ;
```



File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
....
3766.                               newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=210
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3478	3766
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
3478.                                     NULL, NULL) ;
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
....
3766.                                newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 15:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=211>
Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by list_stack at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3462	3793
Object	null	list_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
3462.                                NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
....
3793.                                newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 16:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=212>
Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by list_stack at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3467	3793
Object	null	list_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
3467.                                     NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
....
3793.                                newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=213
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by list_stack at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3470	3793
Object	null	list_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
3470.                                     NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
.....
3793.                                newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=214
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by list_stack at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3474	3793
Object	null	list_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
.....
3474.                                NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
.....
3793.                                newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=215
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by list_stack at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c

Line	3478	3793
Object	null	list_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c

Method garbage_collect(int testing)

```
....
3478.                                     NULL, NULL);
```



File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
....
3793.                                newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 20:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=216>

Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by list_stack at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3484	3793
Object	null	list_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c

Method garbage_collect(int testing)

```
....
3484.                                     NULL, NULL);
```



File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
....
3793.                                newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=217
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3470	3793
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
3470.                                     NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
....
3793.                               newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=218
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3467	3793
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c

Method garbage_collect(int testing)

```
....
3467.                                     NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
....
3793.                                newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 23:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=219>

Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3474	3793
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c

Method garbage_collect(int testing)

```
....
3474.                                     NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
....
3793.                                newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 24:

Severity Low

Result State To Verify

Online Results <http://WIN->

PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=220

Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3484	3793
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....  
3484.                                     NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(
↓

```
....  
3793.                               newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 25:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=221>
Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3462	3793
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)


```
.....
3462.                                     NULL, NULL) ;
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
.....
3793.                                     newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 26:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=222>

Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2022-4293-FP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c	vim@@vim-v8.2.1000-CVE-2022-4293-FP.c
Line	3478	3793
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c

Method garbage_collect(int testing)

```
.....
3478.                                     NULL, NULL) ;
```

File Name vim@@vim-v8.2.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
.....
3793.                                     newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 27:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=223>

Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by ht_stack at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3462	3766
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
3462.                                     NULL, NULL) ;
```



File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
....
3766.                                     newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=224
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by ht_stack at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3467	3766
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
3467.                                     NULL, NULL) ;
```

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
....  
3766. newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 29:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=225>
Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by ht_stack at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3470	3766
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....  
3470. NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
....  
3766. newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 30:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=226>
Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by ht_stack at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3474	3766
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
3474.                                     NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
....
3766.                                newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=227
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by ht_stack at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3478	3766
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
3478.                                     NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
.....
3766.                                newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=228
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by ht_stack at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3484	3766
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
.....
3484.                                NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
.....
3766.                                newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=229
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Line	3484	3766
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method garbage_collect(int testing)

```
....
3484.                                     NULL, NULL);
```



File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method set_ref_in_item(

```
....
3766.                                newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 34:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=230>

Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3478	3766
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method garbage_collect(int testing)

```
....
3478.                                     NULL, NULL);
```



File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method set_ref_in_item(

```
....
3766.                                newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=231
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3470	3766
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
3470.                                     NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
....
3766.                                newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=232
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3467	3766
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method garbage_collect(int testing)

```
....
3467.                                     NULL, NULL);
```



File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method set_ref_in_item(

```
....
3766.                                     newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 37:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=233>

Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3462	3766
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method garbage_collect(int testing)

```
....
3462.                                     NULL, NULL);
```



File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method set_ref_in_item(

```
....
3766.                                     newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 38:

Severity Low

Result State To Verify

Online Results <http://WIN->

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=234
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3474	3766
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
3474.                                     NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
....
3766.                                newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=235
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by list_stack at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3462	3793
Object	null	list_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
.....
3462.                                     NULL, NULL) ;
```

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method set_ref_in_item(

```
.....
3793.                                     newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 40:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=236>

Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by list_stack at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3467	3793
Object	null	list_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method garbage_collect(int testing)

```
.....
3467.                                     NULL, NULL) ;
```

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method set_ref_in_item(

```
.....
3793.                                     newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 41:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=237>

Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by list_stack at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3470	3793
Object	null	list_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
3470.                                     NULL, NULL) ;
```



File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
....
3793.                                     newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=238
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by list_stack at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3474	3793
Object	null	list_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
3474.                                     NULL, NULL) ;
```

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
....  
3793. newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 43:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=239>
Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by list_stack at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3478	3793
Object	null	list_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....  
3478. NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
....  
3793. newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 44:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=240>
Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by list_stack at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3484	3793
Object	null	list_stack

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
3484.                                     NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
....
3793.                                newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 45:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=241
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3474	3793
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
3474.                                     NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
.....
3793.                                newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=242
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3462	3793
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
.....
3462.                                NULL, NULL) ;
```

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
.....
3793.                                newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=243
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Line	3478	3793
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method garbage_collect(int testing)

```
....
3478.                                     NULL, NULL);
```



File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method set_ref_in_item(

```
....
3793.                                newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 48:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=244>

Status New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3467	3793
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method garbage_collect(int testing)

```
....
3467.                                     NULL, NULL);
```



File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method set_ref_in_item(

```
....
3793.                                newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 49:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=245
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3484	3793
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
3484.                                     NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
....
3793.                                newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=246
Status	New

The variable declared in null at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3401 is not initialized when it is used by prev at vim@@vim-v8.2.1000-CVE-2023-0054-TP.c in line 3738.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c	vim@@vim-v8.2.1000-CVE-2023-0054-TP.c
Line	3470	3793
Object	null	prev

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method garbage_collect(int testing)

```
....  
3470.                                     NULL, NULL);
```

File Name vim@@vim-v8.2.1000-CVE-2023-0054-TP.c

Method set_ref_in_item(
↓

```
....  
3793.                                     newitem->prev = *list_stack;
```

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=1030076&projectid=30065&pathid=307>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c
Line	2670	2670
Object	fprintf	fprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2345-FP.c

Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....  
2670.                                     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=1030076&projectid=30065&pathid=307>

Status	065&pathid=308 New
--------	---

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c
Line	2670	2670
Object	fprintf	fprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2581-FP.c
Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....  
2670.                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=1030076&projectid=30065&pathid=309
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	6310	6310
Object	fprintf	fprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method spell_add_word(

```
....  
6310.                fprintf(fd, "%.s/!\n", len, word);
```

Improper Resource Access Authorization\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=310
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c

Line	6312	6312
Object	fprintf	fprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c

Method spell_add_word(

```
....  
6312.                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=1030076&projectid=30065&pathid=311>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	6314	6314
Object	fprintf	fprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c

Method spell_add_word(

```
....  
6314.                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=1030076&projectid=30065&pathid=312>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	4937	4937
Object	fputc	fputc

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c

Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
.....  
4937.          fputc(128, fd);          // <charflagslen>
```

Improper Resource Access Authorization\Path 7:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=313>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	4945	4945
Object	fputc	fputc

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c

Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
.....  
4945.          fputc(flags, fd);          // <charflags>
```

Improper Resource Access Authorization\Path 8:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=314>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	6265	6265
Object	fputc	fputc

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c

Method spell_add_word(

```
.....  
6265.          fputc('#', fd);
```

Improper Resource Access Authorization\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=315
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	6582	6582
Object	fputc	fputc

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method write_spell_prefcond(FILE *fd, garray_T *gap)

```
....  
6582.                fputc(len, fd);
```

Improper Resource Access Authorization\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=316
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	6588	6588
Object	fputc	fputc

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method write_spell_prefcond(FILE *fd, garray_T *gap)

```
....  
6588.                fputc(0, fd);
```

Improper Resource Access Authorization\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=317
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Line	987	987
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Method check_external_diff(diffio_T *diffio)

```
....  
987.                if (fwrite("line1\n", (size_t)6, (size_t)1, fd) != 1)
```

Improper Resource Access Authorization\Path 12:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=318>
Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Line	995	995
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Method check_external_diff(diffio_T *diffio)

```
....  
995.                if (fwrite("line2\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=1030076&projectid=30065&pathid=319>
Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Line	987	987

Object	fwrite	fwrite
--------	--------	--------

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Method check_external_diff(diffio_T *diffio)

```
....  
987.                if (fwrite("line1\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=1030076&projectid=30065&pathid=320>
Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Line	995	995
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Method check_external_diff(diffio_T *diffio)

```
....  
995.                if (fwrite("line2\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=1030076&projectid=30065&pathid=321>
Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Line	987	987
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Method check_external_diff(diffio_T *diffio)

```
....  
987.          if (fwrite("line1\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=1030076&projectid=30065&pathid=322
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Line	995	995
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Method check_external_diff(diffio_T *diffio)

```
....  
995.          if (fwrite("line2\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=1030076&projectid=30065&pathid=323
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c
Line	987	987
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2598-FP.c
Method check_external_diff(diffio_T *diffio)

```
....  
987.          if (fwrite("line1\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=1030076&projectid=30065&pathid=324
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c
Line	995	995
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2598-FP.c
Method check_external_diff(diffio_T *diffio)

```
....  
995.          if (fwrite("line2\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=1030076&projectid=30065&pathid=325
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	4872	4872
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
4872.          fvw &= fwrite(VIMSPELLMAGIC, VIMSPELLMAGICL, (size_t)1, fd);
```

Improper Resource Access Authorization\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=326
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	4891	4891
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
4891.          fvw &= fwrite(spin->si_info, (size_t)i, (size_t)1, fd); //  
<infotext>
```

Improper Resource Access Authorization\Path 21:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=327>
Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	4902	4902
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
4902.          fvw &= fwrite(spin->si_region_name, (size_t)1, (size_t)1,  
fd);
```

Improper Resource Access Authorization\Path 22:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=328>
Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c

Line	4949	4949
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
4949.          fvw &= fwrite(folchars, (size_t)1, (size_t)1, fd); //  
<folchars>
```

Improper Resource Access Authorization\Path 23:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=329>
Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	4960	4960
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
4960.          fvw &= fwrite(spin->si_midword, (size_t)i, (size_t)1, fd);
```

Improper Resource Access Authorization\Path 24:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=330>
Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	5048	5048
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c

Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
5048.                fvw &= fwrite(p, 1, (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=1030076&projectid=30065&pathid=331>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	5066	5066
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c

Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
5066.                fvw &= fwrite(spin->si_sofofr, 1, (size_t)1, fd); //  
<sofofrom>
```

Improper Resource Access Authorization\Path 26:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=332>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	5070	5070
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c

Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
5070.                fvw &= fwrite(spin->si_sofoto, 1, (size_t)1, fd); //  
<sofoto>
```

Improper Resource Access Authorization\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=333
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	5095	5095
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
5095.          fvw &= fwrite(hi->hi_key, (size_t)1, (size_t)1,  
fd);
```

Improper Resource Access Authorization\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=334
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	5111	5111
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
5111.          fvw &= fwrite(spin->si_map.ga_data, (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=1030076&projectid=30065&pathid=335
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	5177	5177
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
5177.          fwv &= fwrite(p, (size_t)STRLEN(p), (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=1030076&projectid=30065&pathid=336
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	5181	5181
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
5181.          fwv &= fwrite(spin->si_compflags, (size_t)STRLEN(spin->si_compflags),
```

Improper Resource Access Authorization\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=337
Status	New

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

File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	5205	5205
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method write_vim_spell(spellinfo_T *spin, char_u *fname)

```
....  
5205.          fwv &= fwrite(spin->si_syllable, (size_t)1, (size_t)1, fd);
```

Improper Resource Access Authorization\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=338
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	5806	5806
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method sug_write(spellinfo_T *spin, char_u *fname)

```
....  
5806.          if (fwrite(VIMSUGMAGIC, VIMSUGMAGICL, (size_t)1, fd) != 1) //  
<fileID>
```

Improper Resource Access Authorization\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=339
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	5848	5848
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method sug_write(spellinfo_T *spin, char_u *fname)

```
....  
5848.          if (fwrite(line, (size_t)len, (size_t)1, fd) == 0)
```

Improper Resource Access Authorization\Path 34:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=340>
Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	6583	6583
Object	fwrite	fwrite

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method write_spell_prefcond(FILE *fd, garray_T *gap)

```
....  
6583.          x &= fwrite(p, (size_t)len, (size_t)1, fd);
```

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=1030076&projectid=30065&pathid=164>
Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	722	722
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c

Method spell_suggest_list(

```
....  
722.          ga_init2(gap, sizeof(char_u *), sug.su_ga.ga_len + 1);
```

Use of Sizeof On a Pointer Type\Path 2:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=165>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Line	813	813
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2208-FP.c

Method diff_try_update(

```
....  
813.          ga_init2(&dio->dio_diff.dout_ga, sizeof(char *), 1000);
```

Use of Sizeof On a Pointer Type\Path 3:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=166>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Line	813	813
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2210-FP.c

Method diff_try_update(

```
....  
813.          ga_init2(&dio->dio_diff.dout_ga, sizeof(char *), 1000);
```

Use of Sizeof On a Pointer Type\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=167
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Line	1172	1172
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Method op_yank(oparg_T *oap, int deleting, int mess)

```
....  
1172.      y_current->y_array = lalloc_clear(sizeof(char_u *) *  
yanklines, TRUE);
```

Use of Sizeof On a Pointer Type\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=168
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Line	1465	1465
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Method copy_yank_reg(yankreg_T *reg)

```
....  
1465.      sizeof(char_u *) * y_current->y_size,  
TRUE);
```

Use of Sizeof On a Pointer Type\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=169

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

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Line	2839	2839
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Method str_to_reg(

```
.....  
2839.      pp = lalloc_clear((y_ptr->y_size + newlines) * sizeof(char_u  
*), TRUE);
```

Use of Sizeof On a Pointer Type\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=170
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Line	813	813
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Method diff_try_update(

```
.....  
813.      ga_init2(&dio->dio_diff.dout_ga, sizeof(char *), 1000);
```

Use of Sizeof On a Pointer Type\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=171
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c

Line	1379	1379
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2345-FP.c

Method cleanup_subexpr(void)

```
....  
1379.          vim_memset(rex.reg_startp, 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=1030076&projectid=30065&pathid=172>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c
Line	1380	1380
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2345-FP.c

Method cleanup_subexpr(void)

```
....  
1380.          vim_memset(rex.reg_endp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 10:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=173>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c
Line	1400	1400
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2345-FP.c

Method cleanup_zsubexpr(void)

```
....  
1400.          vim_memset(reg_startzp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 11:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=174>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c
Line	1401	1401
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2345-FP.c

Method cleanup_zsubexpr(void)

```
....  
1401.          vim_memset(reg_endzp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 12:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=175>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c
Line	1379	1379
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2581-FP.c

Method cleanup_subexpr(void)

```
....  
1379.          vim_memset(rex.reg_startp, 0, sizeof(char_u *) *  
NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=176
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c
Line	1380	1380
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2581-FP.c
Method cleanup_subexpr(void)

```
....  
1380.          vim_memset(rex.reg_endp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=177
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c
Line	1400	1400
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2581-FP.c
Method cleanup_zsubexpr(void)

```
....  
1400.          vim_memset(reg_startzp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=178
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c
Line	1401	1401
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2581-FP.c
Method cleanup_zsubexpr(void)

```
....  
1401.          vim_memset(reg_endzp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 16:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=179>
Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c
Line	813	813
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2598-FP.c
Method diff_try_update(

```
....  
813.          ga_init2(&dio->dio_diff.dout_ga, sizeof(char *), 1000);
```

Use of Sizeof On a Pointer Type\Path 17:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=180>
Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2816-FP.c	vim@@vim-v8.2.1000-CVE-2022-2816-FP.c
Line	1391	1391

Object	sizeof	sizeof
--------	--------	--------

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2816-FP.c

Method find_locales(void)

```
....  
1391.      ga_init2(&locales_ga, sizeof(char_u *), 20);
```

Use of Sizeof On a Pointer Type\Path 18:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=181>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	1256	1256
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c

Method read_compound(FILE *fd, slang_T *slang, int len)

```
....  
1256.      ga_init2(gap, sizeof(char_u *), c);
```

Use of Sizeof On a Pointer Type\Path 19:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=182>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	1418	1418
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c

Method set_sofo(slang_T *lp, char_u *from, char_u *to)


```
.....  
1418.          ga_init2(gap, sizeof(int *), 1);
```

Use of Sizeof On a Pointer Type\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=183
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	1421	1421
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method set_sofo(slang_T *lp, char_u *from, char_u *to)

```
.....  
1421.          vim_memset(gap->ga_data, 0, sizeof(int *) * 256);
```

Use of Sizeof On a Pointer Type\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=184
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	4289	4289
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method getroom(

```
.....  
4289.          bl->sb_used = (bl->sb_used + sizeof(char *) - 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=1030076&projectid=30065&pathid=185
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	4290	4290
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method getroom(

```
....  
4290.                                & ~(sizeof(char *) - 1);
```

Use of Sizeof On a Pointer Type\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=186
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	5905	5905
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method mkspell(

```
....  
5905.          ga_init2(&spin.si_comppat, (int)sizeof(char_u *), 20);
```

Use of Sizeof On a Pointer Type\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=187
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	5906	5906
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method mkspell(

```
.....  
5906.          ga_init2(&spin.si_prefcond, (int)sizeof(char_u *), 50);
```

Use of Sizeof On a Pointer Type\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=188
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	1693	1693
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Method find_tags(

```
.....  
1693.          ga_init2(&ga_match[mtt], (int)sizeof(char_u *), 100);
```

Use of Sizeof On a Pointer Type\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=189
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	2888	2888

Object	sizeof	sizeof
--------	--------	--------

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c

Method get_tagfname(

```
....  
2888.          ga_init2(&tag_fnames, (int)sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 27:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=190>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c
Line	1172	1172
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-3520-FP.c

Method op_yank(oparg_T *oap, int deleting, int mess)

```
....  
1172.      y_current->y_array = lalloc_clear(sizeof(char_u *) *  
yanklines, TRUE);
```

Use of Sizeof On a Pointer Type\Path 28:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=191>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c
Line	1465	1465
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-3520-FP.c

Method copy_yank_reg(yankreg_T *reg)

```
.....
1465.                                sizeof(char_u *) * y_current->y_size,
TRUE);
```

Use of Sizeof On a Pointer Type\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=192
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c
Line	2839	2839
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-3520-FP.c
Method str_to_reg(

```
.....
2839.      pp = lalloc_clear((y_ptr->y_size + newlines) * sizeof(char_u
*), TRUE);
```

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=1030076&projectid=30065&pathid=283
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Line	1803	1803
Object	idx_new	idx_new

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2208-FP.c

Method diff_copy_entry(

```
....  
1803.          dp->df_lnum[idx_new] = dp->df_lnum[idx_orig] - off;
```

Unchecked Array Index\Path 2:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=284>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c	vim@@vim-v8.2.1000-CVE-2022-2208-FP.c
Line	1804	1804
Object	idx_new	idx_new

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2208-FP.c

Method diff_copy_entry(

```
....  
1804.          dp->df_count[idx_new] = dp->df_count[idx_orig];
```

Unchecked Array Index\Path 3:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=285>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Line	1803	1803
Object	idx_new	idx_new

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2210-FP.c

Method diff_copy_entry(

```
....  
1803.          dp->df_lnum[idx_new] = dp->df_lnum[idx_orig] - off;
```

Unchecked Array Index\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=286
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c	vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Line	1804	1804
Object	idx_new	idx_new

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2210-FP.c
Method diff_copy_entry(

```
....  
1804.         dp->df_count[idx_new] = dp->df_count[idx_orig];
```

Unchecked Array Index\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=287
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Line	1440	1440
Object	y_idx	y_idx

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Method yank_copy_line(struct block_def *bd, long y_idx)

```
....  
1440.         y_current->y_array[y_idx] = pnew;
```

Unchecked Array Index\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=288

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

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Line	2587	2587
Object	len	len

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Method get_reg_contents(int regname, int flags)

```
....  
2587.         retval[len] = NUL;
```

Unchecked Array Index\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=289
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Line	1803	1803
Object	idx_new	idx_new

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2289-FP.c
Method diff_copy_entry(

```
....  
1803.         dp->df_lnum[idx_new] = dp->df_lnum[idx_orig] - off;
```

Unchecked Array Index\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=290
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c	vim@@vim-v8.2.1000-CVE-2022-2289-FP.c

Line	1804	1804
Object	idx_new	idx_new

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2289-FP.c

Method diff_copy_entry(

```
.....  
1804.         dp->df_count[idx_new] = dp->df_count[idx_orig];
```

Unchecked Array Index\Path 9:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=291>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c
Line	2428	2428
Object	len	len

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2345-FP.c

Method reg_submatch(int no)

```
.....  
2428.         retval[len] = '\n';
```

Unchecked Array Index\Path 10:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=292>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c
Line	2439	2439
Object	len	len

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2345-FP.c

Method reg_submatch(int no)

```
....  
2439.                retval[len] = '\n';
```

Unchecked Array Index\Path 11:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=293>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c
Line	2447	2447
Object	len	len

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2345-FP.c

Method reg_submatch(int no)

```
....  
2447.                retval[len] = NUL;
```

Unchecked Array Index\Path 12:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=294>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c
Line	2428	2428
Object	len	len

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2581-FP.c

Method reg_submatch(int no)

```
....  
2428.                retval[len] = '\n';
```

Unchecked Array Index\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=295
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c
Line	2439	2439
Object	len	len

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2581-FP.c
Method reg_submatch(int no)

```
....  
2439.                retval[len] = '\n';
```

Unchecked Array Index\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=296
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c
Line	2447	2447
Object	len	len

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2581-FP.c
Method reg_submatch(int no)

```
....  
2447.                retval[len] = NUL;
```

Unchecked Array Index\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=297
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c
Line	1803	1803
Object	idx_new	idx_new

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2598-FP.c

Method diff_copy_entry(

```
....  
1803.      dp->df_lnum[idx_new] = dp->df_lnum[idx_orig] - off;
```

Unchecked Array Index\Path 16:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=298>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c	vim@@vim-v8.2.1000-CVE-2022-2598-FP.c
Line	1804	1804
Object	idx_new	idx_new

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2598-FP.c

Method diff_copy_entry(

```
....  
1804.      dp->df_count[idx_new] = dp->df_count[idx_orig];
```

Unchecked Array Index\Path 17:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=299>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	6648	6648

Object	cl	cl
--------	----	----

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method set_map_str(slang_T *lp, char_u *map)

```
....  
6648.          b[cl] = NUL;
```

Unchecked Array Index\Path 18:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=300>
Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	1559	1559
Object	i	i

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method mb_str2wide(char_u *s)

```
....  
1559.          res[i] = NUL;
```

Unchecked Array Index\Path 19:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=301>
Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	3764	3764
Object	cnt	cnt

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method get_pfxlist(

```
.....
3764.          store_afflist[cnt] = NUL;
```

Unchecked Array Index\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=302
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	3800	3800
Object	cnt	cnt

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method get_compflags(

```
.....
3800.          store_afflist[cnt] = NUL;
```

Unchecked Array Index\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=303
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	2536	2536
Object	len	len

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Method find_tags(

```
.....
2536.          p[len] = '@';
```

Unchecked Array Index\Path 22:

Severity	Low
----------	-----

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=304
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	2627	2627
Object	ebuf_len	ebuf_len

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c

Method find_tags(

```
....  
2627.                s[ebuf_len] = TAG_SEP;
```

Unchecked Array Index\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=305
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c
Line	1440	1440
Object	y_idx	y_idx

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-3520-FP.c

Method yank_copy_line(struct block_def *bd, long y_idx)

```
....  
1440.        y_current->y_array[y_idx] = pnew;
```

Unchecked Array Index\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=306
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c
Line	2587	2587
Object	len	len

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-3520-FP.c
Method get_reg_contents(int regname, int flags)

```
....  
2587.         retval[len] = 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=1030076&projectid=30065&pathid=144
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	1362	1362
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
....  
1362.         sprintf(changename[depth], "prefix");
```

Unchecked Return Value\Path 2:

Severity	Low
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=145
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	1760	1760
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
....  
1760.                                sprintf(changename[depth], "%. *s-%s: split",
```

Unchecked Return Value\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=146
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	1763	1763
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
....  
1763.                                sprintf(changename[depth], "%. *s-%s:  
compound",
```

Unchecked Return Value\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=147
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	1909	1909
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
....  
1909.             sprintf(changename[depth], "%.s-%s: subst %c to  
%c",
```

Unchecked Return Value\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=148
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	1913	1913
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
....  
1913.             sprintf(changename[depth], "%.s-%s: accept %c",
```

Unchecked Return Value\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=149
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2048	2048
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
....  
2048.          sprintf(changename[depth], "%.s-%s: delete %c",
```

Unchecked Return Value\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=150
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2145	2145
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
....  
2145.          sprintf(changename[depth], "%.s-%s: insert %c",
```

Unchecked Return Value\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=151
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2245	2245
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
....  
2245.          sprintf(changename[depth], "%.s-%s: swap %c and %c",
```

Unchecked Return Value\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=152
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2333	2333
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c

Method suggest_trie_walk(

```
....  
2333.          sprintf(changename[depth], "%.s-%s: swap3 %c and %c",
```

Unchecked Return Value\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=153
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2401	2401
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
....  
2401.          sprintf(changename[depth], "%.s-%s: rotate left  
%c%c%c",
```

Unchecked Return Value\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=154
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2462	2462
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c

Method suggest_trie_walk(

```
....  
2462.                                sprintf(changename[depth], "%.s-%s: rotate right  
%c%c%c",
```

Unchecked Return Value\Path 12:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=155>

Status New

The suggest_trie_walk method calls the sprintf function, at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-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-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2574	2574
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c

Method suggest_trie_walk(

```
....  
2574.                                sprintf(changename[depth], "%.s-%s: replace %s  
with %s",
```

Unchecked Return Value\Path 13:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=156>

Status New

The ex_compiler method calls the sprintf function, at line 722 of vim@@vim-v8.2.1000-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	vim@@vim-v8.2.1000-CVE-2022-2816-FP.c	vim@@vim-v8.2.1000-CVE-2022-2816-FP.c
Line	762	762

Object	sprintf	sprintf
--------	---------	---------

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2816-FP.c
Method ex_compiler(exarg_T *eap)

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

Unchecked Return Value\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=157
Status	New

The spell_read_aff method calls the sprintf function, at line 2175 of vim@@vim-v8.2.1000-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	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	2735	2735
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
....  
2735.          sprintf((char *)buf, "^%s", items[4]);
```

Unchecked Return Value\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=158
Status	New

The spell_read_aff method calls the sprintf function, at line 2175 of vim@@vim-v8.2.1000-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	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c

Line	2737	2737
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
....
2737.                                sprintf((char *)buf, "%s$", items[4]);
```

Unchecked Return Value\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=159
Status	New

The spell_read_aff method calls the sprintf function, at line 2175 of vim@@vim-v8.2.1000-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	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	2796	2796
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
....
2796.                                sprintf((char *)buf, "^%s",
```

Unchecked Return Value\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=160
Status	New

The do_tag method calls the sprintf function, at line 132 of vim@@vim-v8.2.1000-CVE-2022-2946-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-v8.2.1000-CVE-2022-2946-	vim@@vim-v8.2.1000-CVE-2022-2946-

	FP.c	FP.c
Line	720	720
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c

Method do_tag(

```
....  
720.                sprintf((char *)IObuff, _("tag %d of %d%s"),
```

Unchecked Return Value\Path 18:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=161>

Status New

The find_tags method calls the sprintf function, at line 1546 of vim@@vim-v8.2.1000-CVE-2022-2946-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-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	2545	2545
Object	sprintf	sprintf

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c

Method find_tags(

```
....  
2545.                sprintf((char *)p + len + 1 + ML_EXTRA, "%06d",
```

Unchecked Return Value\Path 19:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=162>

Status New

The jumpto_tag method calls the sprintf function, at line 3308 of vim@@vim-v8.2.1000-CVE-2022-2946-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
--------	-------------

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=5
Status	New

The size of the buffer used by `do_tag` in `_`, at line 132 of `vim@@vim-v8.2.1000-CVE-2022-2946-FP.c`, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that `do_tag` passes to `"tag %d of %d%s"`, at line 132 of `vim@@vim-v8.2.1000-CVE-2022-2946-FP.c`, to overwrite the target buffer.

	Source	Destination
File	<code>vim@@vim-v8.2.1000-CVE-2022-2946-FP.c</code>	<code>vim@@vim-v8.2.1000-CVE-2022-2946-FP.c</code>
Line	720	720
Object	<code>"tag %d of %d%s"</code>	—

Code Snippet

File Name `vim@@vim-v8.2.1000-CVE-2022-2946-FP.c`
Method `do_tag(`

```
....  
720.             sprintf((char *)IObuff, _("tag %d of %d%s"),
```

Potential Precision Problem\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=6
Status	New

The size of the buffer used by `suggest_trie_walk` in `"%.*s-%s: split"`, at line 1234 of `vim@@vim-v8.2.1000-CVE-2022-2126-FP.c`, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that `suggest_trie_walk` passes to `"%.*s-%s: split"`, at line 1234 of `vim@@vim-v8.2.1000-CVE-2022-2126-FP.c`, to overwrite the target buffer.

	Source	Destination
File	<code>vim@@vim-v8.2.1000-CVE-2022-2126-FP.c</code>	<code>vim@@vim-v8.2.1000-CVE-2022-2126-FP.c</code>
Line	1760	1760
Object	<code>"%.*s-%s: split"</code>	<code>"%.*s-%s: split"</code>

Code Snippet

File Name `vim@@vim-v8.2.1000-CVE-2022-2126-FP.c`
Method `suggest_trie_walk(`

```
....  
1760.             sprintf(changename[depth], "%.*s-%s: split",
```

Potential Precision Problem\Path 3:

Severity	Low
----------	-----

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=7
Status	New

The size of the buffer used by suggest_trie_walk in "%.s-%s: compound", at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that suggest_trie_walk passes to "%.s-%s: compound", at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-FP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	1763	1763
Object	"%.s-%s: compound"	"%.s-%s: compound"

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
....  
1763.                                sprintf(changename[depth], "%.s-%s:  
compound",
```

Potential Precision Problem\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=8
Status	New

The size of the buffer used by suggest_trie_walk in "%.s-%s: subst %c to %c", at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that suggest_trie_walk passes to "%.s-%s: subst %c to %c", at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-FP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	1909	1909
Object	"%.s-%s: subst %c to %c"	"%.s-%s: subst %c to %c"

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
....  
1909.                                sprintf(changename[depth], "%.s-%s: subst %c to  
%c",
```

Potential Precision Problem\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=9
Status	New

The size of the buffer used by suggest_trie_walk in "%.s-%s: accept %c", at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that suggest_trie_walk passes to "%.s-%s: accept %c", at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-FP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	1913	1913
Object	"%.s-%s: accept %c"	"%.s-%s: accept %c"

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
....  
1913.          sprintf(changename[depth], "%.s-%s: accept %c",
```

Potential Precision Problem\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=10
Status	New

The size of the buffer used by suggest_trie_walk in "%.s-%s: delete %c", at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that suggest_trie_walk passes to "%.s-%s: delete %c", at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-FP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2048	2048
Object	"%.s-%s: delete %c"	"%.s-%s: delete %c"

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
.....
2048.          sprintf(changename[depth], "%.s-%s: delete %c",
```

Potential Precision Problem\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=11
Status	New

The size of the buffer used by suggest_trie_walk in "%.s-%s: insert %c", at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that suggest_trie_walk passes to "%.s-%s: insert %c", at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-FP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2145	2145
Object	"%.s-%s: insert %c"	"%.s-%s: insert %c"

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
.....
2145.          sprintf(changename[depth], "%.s-%s: insert %c",
```

Potential Precision Problem\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=12
Status	New

The size of the buffer used by suggest_trie_walk in "%.s-%s: swap %c and %c", at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that suggest_trie_walk passes to "%.s-%s: swap %c and %c", at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-FP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2245	2245
Object	"%.s-%s: swap %c and %c"	"%.s-%s: swap %c and %c"

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c

Method suggest_trie_walk(

```
....  
2245.          sprintf(changename[depth], "%.s-%s: swap %c and %c",
```

Potential Precision Problem\Path 9:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=13>

Status New

The size of the buffer used by suggest_trie_walk in "%.s-%s: swap3 %c and %c", at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that suggest_trie_walk passes to "%.s-%s: swap3 %c and %c", at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-FP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2333	2333
Object	"%.s-%s: swap3 %c and %c"	"%.s-%s: swap3 %c and %c"

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c

Method suggest_trie_walk(

```
....  
2333.          sprintf(changename[depth], "%.s-%s: swap3 %c and %c",
```

Potential Precision Problem\Path 10:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=14>

Status New

The size of the buffer used by suggest_trie_walk in "%.s-%s: rotate left %c%c%c", at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that suggest_trie_walk passes to "%.s-%s: rotate left %c%c%c", at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-FP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2401	2401
Object	"%.s-%s: rotate left %c%c%c"	"%.s-%s: rotate left %c%c%c"

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
....  
2401.          sprintf(changename[depth], "%.s-%s: rotate left  
%c%c%c",
```

Potential Precision Problem\Path 11:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=15>
Status New

The size of the buffer used by suggest_trie_walk in "%.s-%s: rotate right %c%c%c", at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that suggest_trie_walk passes to "%.s-%s: rotate right %c%c%c", at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-FP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2462	2462
Object	"%.s-%s: rotate right %c%c%c"	"%.s-%s: rotate right %c%c%c"

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
....  
2462.          sprintf(changename[depth], "%.s-%s: rotate right  
%c%c%c",
```

Potential Precision Problem\Path 12:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=16>
Status New

The size of the buffer used by suggest_trie_walk in "%.s-%s: replace %s with %s", at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that suggest_trie_walk passes to "%.s-%s: replace %s with %s", at line 1234 of vim@@vim-v8.2.1000-CVE-2022-2126-FP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	2574	2574

Object	"%.*s-%s: replace %s with %s"	"%.*s-%s: replace %s with %s"
--------	-------------------------------	-------------------------------

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method suggest_trie_walk(

```
....
2574.             sprintf(changename[depth], "%.s-%s: replace %s
with %s",
```

Potential Precision Problem\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=17
Status	New

The size of the buffer used by ex_compiler in "compiler/%s.vim", at line 722 of vim@@vim-v8.2.1000-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 722 of vim@@vim-v8.2.1000-CVE-2022-2816-FP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2816-FP.c	vim@@vim-v8.2.1000-CVE-2022-2816-FP.c
Line	762	762
Object	"compiler/%s.vim"	"compiler/%s.vim"

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2816-FP.c
Method ex_compiler(exarg_T *eap)

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

Potential Precision Problem\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=18
Status	New

The size of the buffer used by spell_read_aff in "^%s", at line 2175 of vim@@vim-v8.2.1000-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 2175 of vim@@vim-v8.2.1000-CVE-2022-2923-FP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-	vim@@vim-v8.2.1000-CVE-2022-2923-

	FP.c	FP.c
Line	2735	2735
Object	"^%s"	"^%s"

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
....
2735.                                sprintf((char *)buf, "^%s", items[4]);
```

Potential Precision Problem\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=19
Status	New

The size of the buffer used by spell_read_aff in "%s\$", at line 2175 of vim@@vim-v8.2.1000-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 2175 of vim@@vim-v8.2.1000-CVE-2022-2923-FP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	2737	2737
Object	"%s\$"	"%s\$"

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
....
2737.                                sprintf((char *)buf, "%s$", items[4]);
```

Potential Precision Problem\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=20
Status	New

The size of the buffer used by spell_read_aff in "^%s", at line 2175 of vim@@vim-v8.2.1000-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 2175 of vim@@vim-v8.2.1000-CVE-2022-2923-FP.c, to overwrite the target buffer.

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

File	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c	vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Line	2796	2796
Object	"^%s"	"^%s"

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2923-FP.c
Method spell_read_aff(spellinfo_T *spin, char_u *fname)

```
....
2796.                                sprintf((char *)buf, "%s",
```

Potential Precision Problem\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=21
Status	New

The size of the buffer used by jumpto_tag in "^%s\\s*(", at line 3308 of vim@@vim-v8.2.1000-CVE-2022-2946-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that jumpto_tag passes to "^%s\\s*(", at line 3308 of vim@@vim-v8.2.1000-CVE-2022-2946-FP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	3571	3571
Object	"^%s\\s*("	"^%s\\s*("

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Method jumpto_tag(

```
....
3571.                                sprintf((char *)pbuf, "%s\\s\\*(", tagp.tagname);
```

Potential Precision Problem\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=22
Status	New

The size of the buffer used by jumpto_tag in "^\\[#a-zA-Z_]\\.\\.*\\<%s\\s*(", at line 3308 of vim@@vim-v8.2.1000-CVE-2022-2946-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that jumpto_tag passes to "^\\[#a-zA-Z_]\\.\\.*\\<%s\\s*(", at line 3308 of vim@@vim-v8.2.1000-CVE-2022-2946-FP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c	vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Line	3576	3576
Object	"^\\[#a-zA-Z_]\\.\\.*\\<%s\\s*("	"^\\[#a-zA-Z_]\\.\\.*\\<%s\\s*("

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2946-FP.c
Method jumpto_tag(

```
....  
3576.                                sprintf((char *)pbuf, "^\\[#a-zA-  
z_]\\.\\.\\*\\<%s\\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=1030076&projectid=30065&pathid=275
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Line	1730	1730
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Method do_put(

```
....  
1730.                                coladvance_force(getviscol() + (dir == FORWARD));
```

Arithmenic Operation On Boolean\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=275

Status	065&pathid=276 New
--------	---

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Line	2082	2082
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Method do_put(

```
....  
2082.          if (curbuf->b_op_start.lnum + (y_type == MCHAR) - 1 +  
nr_lines
```

Arithmenic Operation On Boolean\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=277
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c	vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Line	2088	2088
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2264-FP.c
Method do_put(

```
....  
2088.          mark_adjust(curbuf->b_op_start.lnum + (y_type ==  
MCHAR) ,
```

Arithmenic Operation On Boolean\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=278
Status	New

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

File	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c
Line	1333	1333
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2345-FP.c
Method reg_match_visual(void)

```
....  
1333.          if (cols < start || cols > end - (*p_sel == 'e'))
```

Arithmenic Operation On Boolean\Path 5:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=279>
Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c
Line	1333	1333
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2581-FP.c
Method reg_match_visual(void)

```
....  
1333.          if (cols < start || cols > end - (*p_sel == 'e'))
```

Arithmenic Operation On Boolean\Path 6:

Severity Low
Result State To Verify
Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=280>
Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c
Line	1730	1730
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-3520-FP.c

Method do_put(

```
....  
1730.                coladvance_force(getviscol() + (dir == FORWARD));
```

Arithmenic Operation On Boolean\Path 7:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=281>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c
Line	2082	2082
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-3520-FP.c

Method do_put(

```
....  
2082.                if (curbuf->b_op_start.lnum + (y_type == MCHAR) - 1 +  
nr_lines
```

Arithmenic Operation On Boolean\Path 8:

Severity Low

Result State To Verify

Online Results <http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=282>

Status New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c	vim@@vim-v8.2.1000-CVE-2022-3520-FP.c
Line	2088	2088
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-3520-FP.c

Method do_put(

```
....  
2088.                if (curbuf->b_op_start.lnum + (y_type == MCHAR) - 1 +  
nr_lines
```

```
.....
2088.                mark_adjust(curbuf->b_op_start.lnum + (y_type ==
MCHAR),
```

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=1030076&projectid=30065&pathid=193
Status	New

The buffer allocated by <= in vim@@vim-v8.2.1000-CVE-2022-2125-FP.c at line 278 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2125-FP.c	vim@@vim-v8.2.1000-CVE-2022-2125-FP.c
Line	287	287
Object	<=	<=

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2125-FP.c
Method tabstop_copy(int *oldts)

```
.....
287.                for (t = 0; t <= oldts[0]; ++t)
```

Potential Off by One Error in Loops\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=194
Status	New

The buffer allocated by <= in vim@@vim-v8.2.1000-CVE-2022-2126-FP.c at line 4289 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

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

File	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	4376	4376
Object	<=	<=

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method spell_edit_score_limit_w(

```
....  
4376.          for (round = 0; round <= 1; ++round)
```

Potential Off by One Error in Loops\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=195
Status	New

The buffer allocated by <= in vim@@vim-v8.2.1000-CVE-2022-2126-FP.c at line 4114 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c	vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Line	4191	4191
Object	<=	<=

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2126-FP.c
Method spell_edit_score_limit(

```
....  
4191.          for (round = 0; round <= 1; ++round)
```

Potential Off by One Error in Loops\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1030076&projectid=30065&pathid=196
Status	New

The buffer allocated by <= in vim@@vim-v8.2.1000-CVE-2022-2183-FP.c at line 278 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

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

File	vim@@vim-v8.2.1000-CVE-2022-2183-FP.c	vim@@vim-v8.2.1000-CVE-2022-2183-FP.c
Line	287	287
Object	<=	<=

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2183-FP.c
Method tabstop_copy(int *oldts)

```
....  
287.          for (t = 0; t <= oldts[0]; ++t)
```

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=1030076&projectid=30065&pathid=341
Status	New

	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c
Line	2667	2667
Object	f	f

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2345-FP.c
Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....  
2667.          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=1030076&projectid=30065&pathid=342

Status	New	
	Source	Destination
File	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c
Line	2667	2667
Object	f	f

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2581-FP.c
Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....  
2667.          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=1030076&projectid=30065&pathid=343
Status	New

The vim_regcomp method in vim@@vim-v8.2.1000-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-v8.2.1000-CVE-2022-2345-FP.c	vim@@vim-v8.2.1000-CVE-2022-2345-FP.c
Line	2667	2667
Object	fopen	fopen

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2345-FP.c
Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....  
2667.          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=1030076&projectid=30065&pathid=343

[065&pathid=344](#)

Status New

The vim_regcomp method in vim@@vim-v8.2.1000-CVE-2022-2581-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-v8.2.1000-CVE-2022-2581-FP.c	vim@@vim-v8.2.1000-CVE-2022-2581-FP.c
Line	2667	2667
Object	fopen	fopen

Code Snippet

File Name vim@@vim-v8.2.1000-CVE-2022-2581-FP.c
Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....  
2667.          f = fopen(BT_REGEX_DEBUG_LOG_NAME, "a");
```

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;  
}
```

Buffer Overflow AddressOfLocalVarReturned

Risk

What might happen

A use after free error will cause code to use an area of memory previously assigned with a specific value, which has since been freed and may have been overwritten by another value. This error will likely cause unexpected behavior, memory corruption and crash errors. In some cases where the freed and used section of memory is used to determine execution flow, and the error can be induced by an attacker, this may result in execution of malicious code.

Cause

How does it happen

Pointers to variables allow code to have an address with a set size to a dynamically allocated variable. Eventually, the pointer's destination may become free - either explicitly in code, such as when programmatically freeing this variable, or implicitly, such as when a local variable is returned - once it is returned, the variable's scope is released. Once freed, this memory will be re-used by the application, overwritten with new data. At this point, dereferencing this pointer will potentially resolve newly written and unexpected data.

General Recommendations

How to avoid it

- Do not return local variables or pointers
 - Review code to ensure no flow allows use of a pointer after it has been explicitly freed
-

Source Code Examples

CPP

Use of Variable after It was Freed

```
free(input);  
printf("%s", input);
```

Use of Pointer to Local Variable That Was Freed On Return

```
int* func1()  
{  
    int i;  
    i = 1;  
    return &i;  
}  
  
void func2()
```

```
{  
    int j;  
    j = 5;  
}  
  
//..  
int * i = func1();  
printf("%d\r\n", *i); // Output could be 1 or Segmentation Fault  
func2();  
printf("%d\r\n", *i); // Output is 5, which is j's value, as func2() overwrote data in  
the stack  
//..
```

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

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;
}
```

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;
}
```

Failure to Release Memory Before Removing Last Reference ('Memory Leak')

Weakness ID: 401 (*Weakness Base*)

Status: Draft

Description

Description Summary

The software does not sufficiently track and release allocated memory after it has been used, which slowly consumes remaining memory.

Extended Description

This is often triggered by improper handling of malformed data or unexpectedly interrupted sessions.

Terminology Notes

"memory leak" has sometimes been used to describe other kinds of issues, e.g. for information leaks in which the contents of memory are inadvertently leaked (CVE-2003-0400 is one such example of this terminology conflict).

Time of Introduction

- Architecture and Design
- Implementation

Applicable Platforms

Languages

C

C++

Modes of Introduction

Memory leaks have two common and sometimes overlapping causes:

- Error conditions and other exceptional circumstances
- Confusion over which part of the program is responsible for freeing the memory

Common Consequences

Scope	Effect
Availability	Most memory leaks result in general software reliability problems, but if an attacker can intentionally trigger a memory leak, the attacker might be able to launch a denial of service attack (by crashing or hanging the program) or take advantage of other unexpected program behavior resulting from a low memory condition.

Likelihood of Exploit

Medium

Demonstrative Examples

Example 1

The following C function leaks a block of allocated memory if the call to read() fails to return the expected number of bytes:

(*Bad Code*)

Example Language: C

```
char* getBlock(int fd) {
char* buf = (char*) malloc(BLOCK_SIZE);
if (!buf) {
return NULL;
}
if (read(fd, buf, BLOCK_SIZE) != BLOCK_SIZE) {

return NULL;
}
```

```
return buf;
}
```

Example 2

Here the problem is that every time a connection is made, more memory is allocated. So if one just opened up more and more connections, eventually the machine would run out of memory.

(Bad Code)

Example Language: C

```
bar connection() {
foo = malloc(1024);
return foo;
}

endConnection(bar foo) {

free(foo);
}

int main() {

while(1) //thread 1
//On a connection
foo=connection(); //thread 2
//When the connection ends
endConnection(foo)
}
```

Observed Examples

Reference	Description
CVE-2005-3119	Memory leak because function does not free() an element of a data structure.
CVE-2004-0427	Memory leak when counter variable is not decremented.
CVE-2002-0574	Memory leak when counter variable is not decremented.
CVE-2005-3181	Kernel uses wrong function to release a data structure, preventing data from being properly tracked by other code.
CVE-2004-0222	Memory leak via unknown manipulations as part of protocol test suite.
CVE-2001-0136	Memory leak via a series of the same command.

Potential Mitigations

Pre-design: Use a language or compiler that performs automatic bounds checking.

Phase: Architecture and Design

Use an abstraction library to abstract away risky APIs. Not a complete solution.

Pre-design through Build: The Boehm-Demers-Weiser Garbage Collector or valgrind can be used to detect leaks in code. This is not a complete solution as it is not 100% effective.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	398	Indicator of Poor Code Quality	Seven Pernicious Kingdoms (primary)700
ChildOf	Category	399	Resource Management Errors	Development Concepts (primary)699
ChildOf	Category	633	Weaknesses that Affect Memory	Resource-specific Weaknesses (primary)631
ChildOf	Category	730	OWASP Top Ten 2004 Category A9 - Denial of Service	Weaknesses in OWASP Top Ten (2004) (primary)711
ChildOf	Weakness Base	772	Missing Release of Resource after Effective	Research Concepts (primary)1000

MemberOf	View	630	Lifetime Weaknesses Examined by SAMATE	Weaknesses Examined by SAMATE (primary) 630 Research Concepts1000
CanFollow	Weakness Class	390	Detection of Error Condition Without Action	

Relationship Notes

This is often a resultant weakness due to improper handling of malformed data or early termination of sessions.

Affected Resources

- Memory

Functional Areas

- Memory management

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
PLOVER			Memory leak
7 Pernicious Kingdoms			Memory Leak
CLASP			Failure to deallocate data
OWASP Top Ten 2004	A9	CWE More Specific	Denial of Service

White Box Definitions

A weakness where the code path has:

1. start statement that allocates dynamically allocated memory resource
2. end statement that loses identity of the dynamically allocated memory resource creating situation where dynamically allocated memory resource is never relinquished

Where "loses" is defined through the following scenarios:

1. identity of the dynamic allocated memory resource never obtained
2. the statement assigns another value to the data element that stored the identity of the dynamically allocated memory resource and there are no aliases of that data element
3. identity of the dynamic allocated memory resource obtained but never passed on to function for memory resource release
4. the data element that stored the identity of the dynamically allocated resource has reached the end of its scope at the statement and there are no aliases of that data element

References

J. Whittaker and H. Thompson. "How to Break Software Security". Addison Wesley. 2003.

Content History

Submissions			
Submission Date	Submitter	Organization	Source
	PLOVER		Externally Mined
Modifications			
Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-08-15		Veracode	External
	Suggested OWASP Top Ten 2004 mapping		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Relationships, Other Notes, References, Relationship Notes, Taxonomy Mappings, Terminology Notes		
2008-10-14	CWE Content Team	MITRE	Internal
	updated Description		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Other Notes		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Name		
2009-07-17	KDM Analytics		External
	Improved the White Box Definition		

2009-07-27	CWE Content Team	MITRE	Internal	
	updated White Box Definitions			
2009-10-29	CWE Content Team	MITRE	Internal	
	updated Modes of Introduction, Other Notes			
2010-02-16	CWE Content Team	MITRE	Internal	
	updated Relationships			
Previous Entry Names				
Change Date	Previous Entry Name			
2008-04-11	Memory Leak			
2009-05-27	Failure to Release Memory Before Removing Last Reference (aka 'Memory Leak')			

[BACK TO TOP](#)

Use of Uninitialized Pointer

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
 - Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
 - Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.
-

Source Code Examples

Use of Uninitialized Variable

Weakness ID: 457 (Weakness Variant)

Status: Draft

Description

Description Summary

The code uses a variable that has not been initialized, leading to unpredictable or unintended results.

Extended Description

In some languages, such as C, an uninitialized variable contains contents of previously-used memory. An attacker can sometimes control or read these contents.

Time of Introduction

Implementation

Applicable Platforms

Languages

C: (Sometimes)

C++: (Sometimes)

Perl: (Often)

All

Common Consequences

Scope	Effect
Availability Integrity	Initial variables usually contain junk, which can not be trusted for consistency. This can lead to denial of service conditions, or modify control flow in unexpected ways. In some cases, an attacker can "pre-initialize" the variable using previous actions, which might enable code execution. This can cause a race condition if a lock variable check passes when it should not.
Authorization	Strings that are not initialized are especially dangerous, since many functions expect a null at the end -- and only at the end - of a string.

Likelihood of Exploit

High

Demonstrative Examples

Example 1

The following switch statement is intended to set the values of the variables aN and bN, but in the default case, the programmer has accidentally set the value of aN twice. As a result, bN will have an undefined value.

(Bad Code)

Example Language: C

```
switch (ctl) {  
case -1:  
aN = 0;  
bN = 0;  
break;  
case 0:  
aN = i;  
bN = -i;  
break;  
case 1:  
aN = i + NEXT_SZ;  
bN = i - NEXT_SZ;  
break;  
default:  
aN = 0;  
bN = 0;  
break;  
}
```

```
aN = -1;
aN = -1;
break;
}
repaint(aN, bN);
```

Most uninitialized variable issues result in general software reliability problems, but if attackers can intentionally trigger the use of an uninitialized variable, they might be able to launch a denial of service attack by crashing the program. Under the right circumstances, an attacker may be able to control the value of an uninitialized variable by affecting the values on the stack prior to the invocation of the function.

Example 2

Example Languages: C++ and Java

```
int foo;
void bar() {
if (foo==0)
/.../
/..//
}
```

Observed Examples

Reference	Description
CVE-2008-0081	Uninitialized variable leads to code execution in popular desktop application.
CVE-2007-4682	Crafted input triggers dereference of an uninitialized object pointer.
CVE-2007-3468	Crafted audio file triggers crash when an uninitialized variable is used.
CVE-2007-2728	Uninitialized random seed variable used.

Potential Mitigations

Phase: Implementation

Assign all variables to an initial value.

Phase: Build and Compilation

Most compilers will complain about the use of uninitialized variables if warnings are turned on.

Phase: Requirements

The choice could be made to use a language that is not susceptible to these issues.

Phase: Architecture and Design

Mitigating technologies such as safe string libraries and container abstractions could be introduced.

Other Notes

Before variables are initialized, they generally contain junk data of what was left in the memory that the variable takes up. This data is very rarely useful, and it is generally advised to pre-initialize variables or set them to their first values early. If one forgets -- in the C language -- to initialize, for example a char *, many of the simple string libraries may often return incorrect results as they expect the null termination to be at the end of a string.

Stack variables in C and C++ are not initialized by default. Their initial values are determined by whatever happens to be in their location on the stack at the time the function is invoked. Programs should never use the value of an uninitialized variable. It is not uncommon for programmers to use an uninitialized variable in code that handles errors or other rare and exceptional circumstances. Uninitialized variable warnings can sometimes indicate the presence of a typographic error in the code.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	398	Indicator of Poor Code Quality	Seven Pernicious Kingdoms (primary)700
ChildOf	Weakness Base	456	Missing Initialization	Development Concepts (primary)699 Research Concepts

MemberOf	View	630	Weaknesses Examined by SAMATE	(primary)1000 Weaknesses Examined by SAMATE (primary)630
----------	------	-----	---	---

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Uninitialized variable
7 Pernicious Kingdoms			Uninitialized Variable

White Box Definitions

A weakness where the code path has:

1. start statement that defines variable
2. end statement that accesses the variable
3. the code path does not contain a statement that assigns value to the variable

References

mercy. "Exploiting Uninitialized Data". Jan 2006. < <http://www.felinemenace.org/~mercy/papers/UBehavior/UBehavior.zip>>.

Microsoft Security Vulnerability Research & Defense. "MS08-014 : The Case of the Uninitialized Stack Variable Vulnerability". 2008-03-11. <<http://blogs.technet.com/swi/archive/2008/03/11/the-case-of-the-uninitialized-stack-variable-vulnerability.aspx>>.

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, Description, Relationships, Observed Example, Other Notes, References, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences, Demonstrative Examples, Potential Mitigations		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
Previous Entry Names			
Change Date	Previous Entry Name		
2008-04-11	Uninitialized Variable		

[BACK TO TOP](#)

Use of Zero Initialized Pointer

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
 - Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
 - Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.
-

Source Code Examples

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());
```



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));  
    }  
}
```


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

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	Development

			Dereference	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