vul_files_72 Scan Report

Project Name	vul_files_72
Scan Start	Thursday, January 9, 2025 11:07:50 AM
Preset	Checkmarx Default
Scan Time	01h:34m:59s
Lines Of Code Scanned	291676
Files Scanned	59
Report Creation Time	Thursday, January 9, 2025 12:08:08 PM
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066
Team	CxServer
Checkmarx Version	8.7.0
Scan Type	Full
Source Origin	LocalPath
Density	3/1000 (Vulnerabilities/LOC)
Visibility	Public

Filter Settings

Severity

Included: High, Medium, Low, Information

Excluded: None

Result State

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

Excluded: None

Assigned to

Included: All

Categories

Included:

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

Excluded:

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

NIST SP 800-53   None

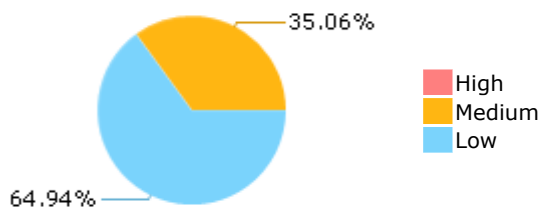OWASP Top 10 2017   None

OWASP Mobile Top 10   None
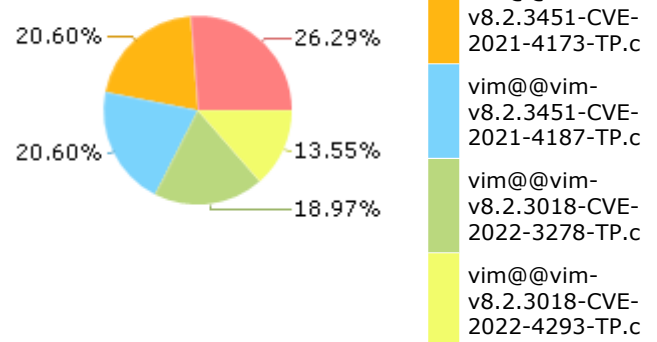2016

Results Limit

Results limit per query was set to 50

Selected Queries

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

Result Summary

High
Medium
Low

35.06%
64.94%

Most Vulnerable Files

20.60%
26.29%
20.60%
13.55%
18.97%

vim@@vim-v8.2.3451-CVE-2022-0156-TP.c

vim@@vim-v8.2.3451-CVE-2021-4173-TP.c

vim@@vim-v8.2.3451-CVE-2021-4187-TP.c

vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Top 5 Vulnerabilities

Use of Zero Initialized Pointer

Dangerous Functions

Integer Overflow

MemoryFree on StackVariable

Use of Uninitialized Pointer

0   32   64   96   128   160
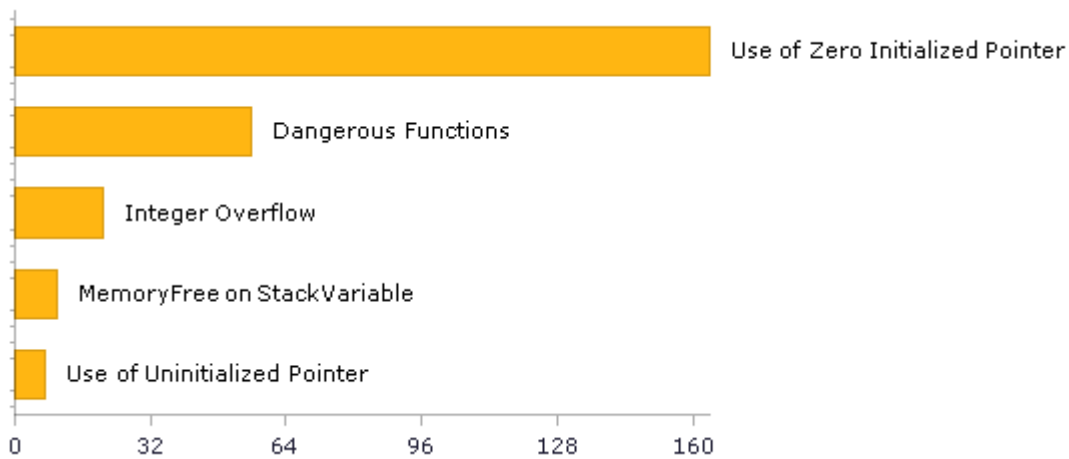
Scan Summary - OWASP Top 10 2017

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	308	109
A2-Broken Authentication	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	38	38
A3-Sensitive Data Exposure	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	App. Specific	0	0
A4-XML External Entities (XXE)	App. Specific	AVERAGE	COMMON	EASY	SEVERE	App. Specific	0	0
A5-Broken Access Control*	App. Specific	AVERAGE	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A6-Security Misconfiguration	App. Specific	EASY	WIDESPREAD	EASY	MODERATE	App. Specific	0	0
A7-Cross-Site Scripting (XSS)	App. Specific	EASY	WIDESPREAD	EASY	MODERATE	App. Specific	0	0
A8-Insecure Deserialization	App. Specific	DIFFICULT	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A9-Using Components with Known Vulnerabilities*	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	MODERATE	App. Specific	56	56
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	56	56
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	9	9
PCI DSS (3.2) - 6.5.2 - Buffer overflows	25	25
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.	18	18
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.	36	36
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.	21	21

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

Scan Summary - NIST SP 800-53

Category	Issues Found	Best Fix Locations
AC-12 Session Termination (P2)	0	0
AC-3 Access Enforcement (P1)	38	38
AC-4 Information Flow Enforcement (P1)	0	0
AC-6 Least Privilege (P1)	0	0
AU-9 Protection of Audit Information (P1)	0	0
CM-6 Configuration Settings (P2)	0	0
IA-5 Authenticator Management (P1)	0	0
IA-6 Authenticator Feedback (P2)	0	0
IA-8 Identification and Authentication (Non-Organizational Users) (P1)	0	0
SC-12 Cryptographic Key Establishment and Management (P1)	0	0
SC-13 Cryptographic Protection (P1)	0	0
SC-17 Public Key Infrastructure Certificates (P1)	0	0
SC-18 Mobile Code (P2)	0	0
SC-23 Session Authenticity (P1)*	0	0
SC-28 Protection of Information at Rest (P1)	0	0
SC-4 Information in Shared Resources (P1)	0	0
SC-5 Denial of Service Protection (P1)*	469	176
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*	64	64
SI-11 Error Handling (P2)*	44	44
SI-15 Information Output Filtering (P0)	0	0
SI-16 Memory Protection (P1)	9	9

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

Scan Summary - OWASP Mobile Top 10 2016

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

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

Scan Summary - Custom

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

Results Distribution By Status   First scan of the project

	High	Medium	Low	Information	Total
New Issues	0	271	502	0	773
Recurrent Issues	0	0	0	0	0
Total	0	271	502	0	773

	High	Medium	Low	Information	Total
Fixed Issues	0	0	0	0	0
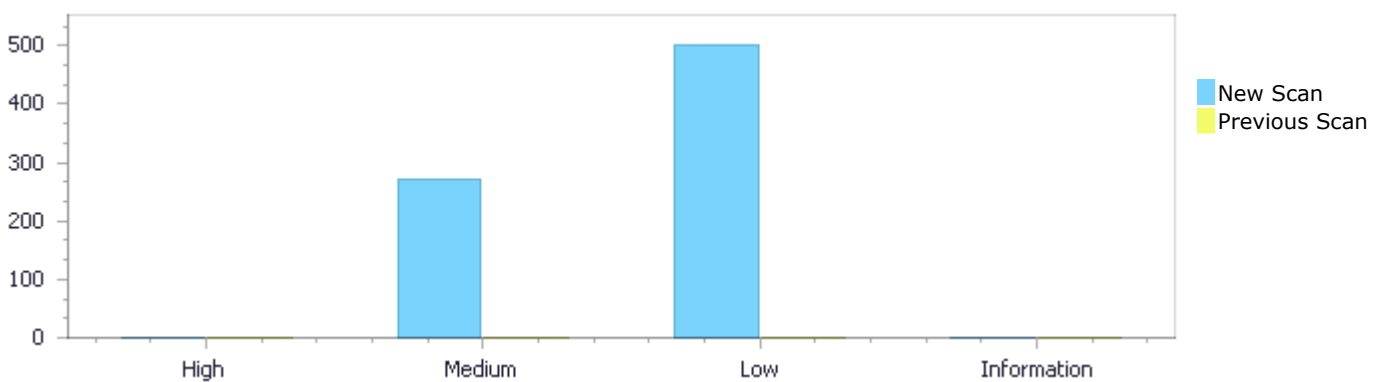
Results Distribution By State

	High	Medium	Low	Information	Total
Confirmed	0	0	0	0	0
Not Exploitable	0	0	0	0	0
To Verify	0	271	502	0	773
Urgent	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0
Total	0	271	502	0	773

Result Summary

Vulnerability Type	Occurrences	Severity
Use of Zero Initialized Pointer	164	Medium
Dangerous Functions	56	Medium
Integer Overflow	21	Medium
MemoryFree on StackVariable	10	Medium
Use of Uninitialized Pointer	7	Medium

[Divide By Zero](#)	6	Medium
[Char Overflow](#)	3	Medium
[Use of Uninitialized Variable](#)	2	Medium
[Buffer Overflow AddressOfLocalVarReturned](#)	1	Medium
[Memory Leak](#)	1	Medium
[NULL Pointer Dereference](#)	276	Low
[Use of Sizeof On a Pointer Type](#)	75	Low
[Unchecked Return Value](#)	44	Low
[Improper Resource Access Authorization](#)	36	Low
[Potential Precision Problem](#)	22	Low
[Arithmenic Operation On Boolean](#)	18	Low
[Unchecked Array Index](#)	18	Low
[Potential Off by One Error in Loops](#)	9	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-v8.2.3018-CVE-2022-3278-TP.c	30
vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	30
vim@@vim-v8.2.3451-CVE-2021-4187-TP.c	30
vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	18
vim@@vim-v8.2.3018-CVE-2023-0054-TP.c	18
vim@@vim-v8.2.3451-CVE-2021-4136-TP.c	17
vim@@vim-v8.2.3451-CVE-2022-0351-FP.c	17
vim@@vim-v8.2.3451-CVE-2022-1420-FP.c	17
vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	12
vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	12

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=1050077&projectid=50066&pathid=157
Status	New

The variable declared in type at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 65 is not initialized when it is used by ga_data at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 65.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	130	163
Object	type	ga_data

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   one_function_arg(

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

Use of Zero Initialized Pointer\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=158
Status	New

The variable declared in Pointer at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614 is not initialized when it is used by ga_data at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614.

Source	Destination

File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	801	968
Object	Pointer	ga_data

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   get_function_body(

```
....
801.                          *line_to_free = NULL;
....
968.          ((char_u **)(newlines->ga_data))[newlines->ga_len++] = p;
```

Use of Zero Initialized Pointer\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=159
Status	New

The variable declared in cmdline at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 999 is not initialized when it is used by ga_data at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	1015	968
Object	cmdline	ga_data

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   lambda_function_body(

```
....
1015.        char_u *cmdline = NULL;
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   get_function_body(

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

Use of Zero Initialized Pointer\Path 4:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=160
Status	New

The variable declared in Pointer at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614 is not initialized when it is used by heredoc_trimmed at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	801	937
Object	Pointer	heredoc_trimmed

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   get_function_body(

```
....
801.                          *line_to_free = NULL;
....
937.                     heredoc_trimmed = vim_strnsave(theline,
```

Use of Zero Initialized Pointer\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=161
Status	New

The variable declared in cmdline at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 999 is not initialized when it is used by heredoc_trimmed at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	1015	937
Object	cmdline	heredoc_trimmed

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   lambda_function_body(

```
....
1015.      char_u *cmdline = NULL;
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method	get_function_body(

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

Use of Zero Initialized Pointer\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=162
Status	New

The variable declared in Pointer at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614 is not initialized when it is used by heredoc_trimmed at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	801	898
Object	Pointer	heredoc_trimmed

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	get_function_body(

```
....
801.                        *line_to_free = NULL;
....
898.                  heredoc_trimmed = vim_strnsave(theline,
```

Use of Zero Initialized Pointer\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=163
Status	New

The variable declared in cmdline at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 999 is not initialized when it is used by heredoc_trimmed at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	1015	898
Object	cmdline	heredoc_trimmed

Code Snippet	

File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	lambda_function_body(

```
....
1015.        char_u *cmdline = NULL;
```

▼

File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	get_function_body(

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

Use of Zero Initialized Pointer\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=164
Status	New

The variable declared in Pointer at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614 is not initialized when it is used by name at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3387.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	801	3506
Object	Pointer	name

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	get_function_body(

```
....
801.                     *line_to_free = NULL;
```

▼

File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	trans_function_name(

```
....
3506.        name = deref_func_name(*pp, &len, partial, type,
```

Use of Zero Initialized Pointer\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50

Status	New

The variable declared in cmdline at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 999 is not initialized when it is used by name at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3387.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	1015	3506
Object	cmdline	name

Code Snippet

File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	lambda_function_body(

```
....
1015.      char_u *cmdline = NULL;
```

▼

File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	trans_function_name(

```
....
3506.       name = deref_func_name(*pp, &len, partial, type,
```

Use of Zero Initialized Pointer\Path 10:

Severity	Medium
Result State	To Verify
Online Results	
Status	New

The variable declared in Pointer at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614 is not initialized when it is used by name at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3387.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	801	3615
Object	Pointer	name

Code Snippet

File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	get_function_body(

```
....
801.                         *line_to_free = NULL;
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   trans_function_name(

```
....
3615.      name = alloc(len + lead + extra + 1);
```

Use of Zero Initialized Pointer\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=167
Status	New

The variable declared in cmdline at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 999 is not initialized when it is used by name at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3387.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	1015	3615
Object	cmdline	name

Code Snippet

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   lambda_function_body(

```
....
1015.      char_u *cmdline = NULL;
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   trans_function_name(

```
....
3615.      name = alloc(len + lead + extra + 1);
```

Use of Zero Initialized Pointer\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=168
Status	New

The variable declared in Pointer at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614 is not initialized when it is used by nextcmd at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	801	778
Object	Pointer	nextcmd

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	get_function_body(

```
....
801.                          *line_to_free = NULL;
....
778.                      nextcmd = p + 1;
```

Use of Zero Initialized Pointer\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=169
Status	New

The variable declared in cmdline at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 999 is not initialized when it is used by nextcmd at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	1015	778
Object	cmdline	nextcmd

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	lambda_function_body(

```
....
1015.        char_u *cmdline = NULL;
```

▼

File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	get_function_body(

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

Use of Zero Initialized Pointer\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=170
Status	New

The variable declared in cmdline at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 999 is not initialized when it is used by ga_data at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 1186.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	1015	1354
Object	cmdline	ga_data

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   lambda_function_body(

```
....
1015.       char_u *cmdline = NULL;
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   get_lambda_tv(

```
....
1354.                ((char_u **)(newlines.ga_data))[newlines.ga_len++] =
```

Use of Zero Initialized Pointer\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=171
Status	New

The variable declared in Pointer at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614 is not initialized when it is used by ga_data at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 1186.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Line	801	1354
Object	Pointer	ga_data

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   get_function_body(

```
....
801.                         *line_to_free = NULL;
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   get_lambda_tv(

```
....
1354.              ((char_u **)(newlines.ga_data))[newlines.ga_len++] =
```

Use of Zero Initialized Pointer\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=172
Status	New

The variable declared in cmdline at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 999 is not initialized when it is used by ga_data at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 1186.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	1015	1338
Object	cmdline	ga_data

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   lambda_function_body(

```
....
1015.       char_u *cmdline = NULL;
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   get_lambda_tv(

```
....
1338.       ((char_u **)(newlines.ga_data))[newlines.ga_len++] = p;
```

Use of Zero Initialized Pointer\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=173
Status	New

The variable declared in Pointer at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614 is not initialized when it is used by ga_data at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 1186.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	801	1338
Object	Pointer	ga_data

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	get_function_body(

```
....
801.                            *line_to_free = NULL;
```

▼

File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	get_lambda_tv(

```
....
1338.        ((char_u **)(newlines.ga_data))[newlines.ga_len++] = p;
```

Use of Zero Initialized Pointer\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=174
Status	New

The variable declared in pt at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 1186 is not initialized when it is used by pt at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 1186.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	1200	1320
Object	pt	pt

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   get_lambda_tv(

```
....
1200.        partial_T   *pt = NULL;
....
1320.         pt = ALLOC_CLEAR_ONE(partial_T);
```

Use of Zero Initialized Pointer\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=175
Status	New

The variable declared in fname at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3072 is not initialized when it is used by after_script at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 1730.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	3087	1757
Object	fname	after_script

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   call_func(

```
....
3087.        char_u *fname = NULL;
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   find_func_even_dead(char_u *name, int is_global, cctx_T *cctx)

```
....
1757.            after_script = name + 3;
```

Use of Zero Initialized Pointer\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=176
Status	New

The variable declared in tofree at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 5027 is not initialized when it is used by s at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 5027.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	5030	5034
Object	tofree	s

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   get_return_cmd(void *rettv)

```
....
5030.        char_u *tofree = NULL;
....
5034.         s = echo_string((typval_T *)rettv, &tofree, numbuf, 0);
```

Use of Zero Initialized Pointer\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=177
Status	New

The variable declared in ht at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 809 is not initialized when it is used by ll_tv at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 809.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	828	924
Object	ht	ll_tv

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   get_lval(

```
....
828.      hashtab_T    *ht = NULL;
....
924.      lp->ll_tv = &v->di_tv;
```

Use of Zero Initialized Pointer\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=178
Status	New

The variable declared in ht_stack at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4616 is not initialized when it is used by ht_stack at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4616.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4622	4649
Object	ht_stack	ht_stack

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
4622.       ht_stack_T  *ht_stack = NULL;
....
4649.        ht_stack = ht_stack->prev;
```

Use of Zero Initialized Pointer\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=179
Status	New

The variable declared in ht_stack at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4616 is not initialized when it is used by ht_stack at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4616.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4622	4647
Object	ht_stack	ht_stack

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
4622.       ht_stack_T  *ht_stack = NULL;
....
4647.        cur_ht = ht_stack->ht;
```

Use of Zero Initialized Pointer\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=180

Status	New

The variable declared in list_stack at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4693 is not initialized when it is used by list_stack at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4693.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4698	4717
Object	list_stack	list_stack

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
4698.        list_stack_T *list_stack = NULL;
....
4717.         list_stack = list_stack->prev;
```

Use of Zero Initialized Pointer\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=181
Status	New

The variable declared in list_stack at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4693 is not initialized when it is used by list_stack at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4693.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4698	4715
Object	list_stack	list_stack

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
4698.        list_stack_T *list_stack = NULL;
....
4715.         cur_l = list_stack->list;
```

Use of Zero Initialized Pointer\Path 26:

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

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=182
Status	New

The variable declared in ht at vim@@vim-v8.2.3018-CVE-2023-0054-TP.c in line 809 is not initialized when it is used by ll_tv at vim@@vim-v8.2.3018-CVE-2023-0054-TP.c in line 809.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c
Line	828	924
Object	ht	ll_tv

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-0054-TP.c
Method   get_lval(

```
....
828.        hashtab_T     *ht = NULL;
....
924.        lp->ll_tv = &v->di_tv;
```

Use of Zero Initialized Pointer\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=183
Status	New

The variable declared in ht_stack at vim@@vim-v8.2.3018-CVE-2023-0054-TP.c in line 4616 is not initialized when it is used by ht_stack at vim@@vim-v8.2.3018-CVE-2023-0054-TP.c in line 4616.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c
Line	4622	4649
Object	ht_stack	ht_stack

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-0054-TP.c
Method   set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
4622.        ht_stack_T   *ht_stack = NULL;
....
4649.         ht_stack = ht_stack->prev;
```

Use of Zero Initialized Pointer\Path 28:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=184
Status	New

The variable declared in ht_stack at vim@@vim-v8.2.3018-CVE-2023-0054-TP.c in line 4616 is not initialized when it is used by ht_stack at vim@@vim-v8.2.3018-CVE-2023-0054-TP.c in line 4616.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c
Line	4622	4647
Object	ht_stack	ht_stack

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-0054-TP.c
Method   set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
4622.        ht_stack_T   *ht_stack = NULL;
....
4647.         cur_ht = ht_stack->ht;
```

Use of Zero Initialized Pointer\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=185
Status	New

The variable declared in list_stack at vim@@vim-v8.2.3018-CVE-2023-0054-TP.c in line 4693 is not initialized when it is used by list_stack at vim@@vim-v8.2.3018-CVE-2023-0054-TP.c in line 4693.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c
Line	4698	4717
Object	list_stack	list_stack

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-0054-TP.c
Method   set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
4698.        list_stack_T *list_stack = NULL;
....
4717.         list_stack = list_stack->prev;
```

Use of Zero Initialized Pointer\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=186
Status	New

The variable declared in list_stack at vim@@vim-v8.2.3018-CVE-2023-0054-TP.c in line 4693 is not initialized when it is used by list_stack at vim@@vim-v8.2.3018-CVE-2023-0054-TP.c in line 4693.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c
Line	4698	4715
Object	list_stack	list_stack

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c
Method	set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
4698.        list_stack_T *list_stack = NULL;
....
4715.         cur_l = list_stack->list;
```

Use of Zero Initialized Pointer\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=187
Status	New

The variable declared in next_leader_flags at vim@@vim-v8.2.3018-CVE-2023-0433-TP.c in line 931 is not initialized when it is used by flags at vim@@vim-v8.2.3018-CVE-2023-0433-TP.c in line 454.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-0433-TP.c	vim@@vim-v8.2.3018-CVE-2023-0433-TP.c
Line	999	472
Object	next_leader_flags	flags

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2023-0433-TP.c
Method	format_lines(

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

File Name	vim@@@vim-v8.2.3018-CVE-2023-0433-TP.c
Method	fmt_check_par(

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

Use of Zero Initialized Pointer\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=188
Status	New

The variable declared in leader_flags at vim@@@vim-v8.2.3018-CVE-2023-0433-TP.c in line 931 is not initialized when it is used by flags at vim@@@vim-v8.2.3018-CVE-2023-0433-TP.c in line 454.

	Source	Destination
File	vim@@@vim-v8.2.3018-CVE-2023-0433-TP.c	vim@@@vim-v8.2.3018-CVE-2023-0433-TP.c
Line	943	472
Object	leader_flags	flags

Code Snippet

File Name	vim@@@vim-v8.2.3018-CVE-2023-0433-TP.c
Method	format_lines(

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


File Name	vim@@@vim-v8.2.3018-CVE-2023-0433-TP.c
Method	fmt_check_par(

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

Use of Zero Initialized Pointer\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=189
Status	New

The variable declared in ht at vim@@@vim-v8.2.3451-CVE-2021-4136-TP.c in line 816 is not initialized when it is used by ll_tv at vim@@@vim-v8.2.3451-CVE-2021-4136-TP.c in line 816.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Line	834	933
Object	ht	ll_tv

Code Snippet

File Name   vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Method   get_lval(

```
....
834.        hashtab_T      *ht = NULL;
....
933.            lp->ll_tv = &v->di_tv;
```

Use of Zero Initialized Pointer\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=190
Status	New

The variable declared in ht_stack at vim@@vim-v8.2.3451-CVE-2021-4136-TP.c in line 4621 is not initialized when it is used by ht_stack at vim@@vim-v8.2.3451-CVE-2021-4136-TP.c in line 4621.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Line	4627	4654
Object	ht_stack	ht_stack

Code Snippet

File Name   vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Method   set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
4627.       ht_stack_T   *ht_stack = NULL;
....
4654.          ht_stack = ht_stack->prev;
```

Use of Zero Initialized Pointer\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=191
Status	New

The variable declared in ht_stack at vim@@vim-v8.2.3451-CVE-2021-4136-TP.c in line 4621 is not initialized when it is used by ht_stack at vim@@vim-v8.2.3451-CVE-2021-4136-TP.c in line 4621.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Line	4627	4652
Object	ht_stack	ht_stack

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Method   set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
4627.        ht_stack_T   *ht_stack = NULL;
....
4652.          cur_ht = ht_stack->ht;
```

Use of Zero Initialized Pointer\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=192
Status	New

The variable declared in list_stack at vim@@vim-v8.2.3451-CVE-2021-4136-TP.c in line 4698 is not initialized when it is used by list_stack at vim@@vim-v8.2.3451-CVE-2021-4136-TP.c in line 4698.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Line	4703	4722
Object	list_stack	list_stack

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Method   set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
4703.        list_stack_T *list_stack = NULL;
....
4722.          list_stack = list_stack->prev;
```

Use of Zero Initialized Pointer\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=193

Status	New

The variable declared in list_stack at vim@@vim-v8.2.3451-CVE-2021-4136-TP.c in line 4698 is not initialized when it is used by list_stack at vim@@vim-v8.2.3451-CVE-2021-4136-TP.c in line 4698.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Line	4703	4720
Object	list_stack	list_stack

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Method   set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
4703.       list_stack_T *list_stack = NULL;
....
4720.        cur_l = list_stack->list;
```

Use of Zero Initialized Pointer\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=194
Status	New

The variable declared in type at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 65 is not initialized when it is used by ga_data at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 65.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	130	163
Object	type	ga_data

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   one_function_arg(

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

Use of Zero Initialized Pointer\Path 39:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=195
Status	New

The variable declared in Pointer at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 652 is not initialized when it is used by ga_data at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 652.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	844	1052
Object	Pointer	ga_data

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   get_function_body(

```
....
844.                          *line_to_free = NULL;
....
1052.        ((char_u **)(newlines->ga_data))[newlines->ga_len++] = p;
```

Use of Zero Initialized Pointer\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=196
Status	New

The variable declared in cmdline at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 1083 is not initialized when it is used by ga_data at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 652.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	1099	1052
Object	cmdline	ga_data

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   lambda_function_body(

```
....
1099.      char_u *cmdline = NULL;
```

▼

File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c

Method	get_function_body(

```
....
1052.          ((char_u **)(newlines->ga_data))[newlines->ga_len++] = p;
```

Use of Zero Initialized Pointer\Path 41:

Severity	Medium
Result State	To Verify
Online Results	
Status	New

The variable declared in Pointer at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 652 is not initialized when it is used by heredoc_trimmed at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 652.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	844	1021
Object	Pointer	heredoc_trimmed

Code Snippet

File Name	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method	get_function_body(

```
....
844.                     *line_to_free = NULL;
....
1021.                    heredoc_trimmed = vim_strnsave(theline,
```

Use of Zero Initialized Pointer\Path 42:

Severity	Medium
Result State	To Verify
Online Results	
Status	New

The variable declared in cmdline at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 1083 is not initialized when it is used by heredoc_trimmed at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 652.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	1099	1021
Object	cmdline	heredoc_trimmed

Code Snippet

File Name	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method	lambda_function_body(

```
....
1099.       char_u *cmdline = NULL;
```

▼

File Name	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method	get_function_body(

```
....
1021.              heredoc_trimmed = vim_strnsave(theline,
```

Use of Zero Initialized Pointer\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=199
Status	New

The variable declared in Pointer at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 652 is not initialized when it is used by heredoc_trimmed at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 652.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	844	982
Object	Pointer	heredoc_trimmed

Code Snippet
File Name	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method	get_function_body(

```
....
844.                    *line_to_free = NULL;
....
982.              heredoc_trimmed = vim_strnsave(theline,
```

Use of Zero Initialized Pointer\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=200
Status	New

The variable declared in cmdline at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 1083 is not initialized when it is used by heredoc_trimmed at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 652.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	1099	982
Object	cmdline	heredoc_trimmed

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   lambda_function_body(

```
....
1099.        char_u *cmdline = NULL;
```

                            ▼

File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c

Method   get_function_body(

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

Use of Zero Initialized Pointer\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=201
Status	New

The variable declared in Pointer at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 652 is not initialized when it is used by name at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 3496.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	844	3615
Object	Pointer	name

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   get_function_body(

```
....
844.                    *line_to_free = NULL;
```

                            ▼

File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c

Method   trans_function_name(

```
....
3615.          name = deref_func_name(*pp, &len, partial, type,
```

Use of Zero Initialized Pointer\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=202
Status	New

The variable declared in cmdline at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 1083 is not initialized when it is used by name at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 3496.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	1099	3615
Object	cmdline	name

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   lambda_function_body(

```
....
1099.        char_u *cmdline = NULL;
```

▼

File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c

Method   trans_function_name(

```
....
3615.          name = deref_func_name(*pp, &len, partial, type,
```

Use of Zero Initialized Pointer\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=203
Status	New

The variable declared in Pointer at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 652 is not initialized when it is used by name at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 3496.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c

Line	844	3725
Object	Pointer	name

Code Snippet
File Name   vim@@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   get_function_body(

```
....
844.                          *line_to_free = NULL;
```

▼

File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c

Method   trans_function_name(

```
....
3725.       name = alloc(len + lead + extra + 1);
```

Use of Zero Initialized Pointer\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=204
Status	New

The variable declared in cmdline at vim@@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 1083 is not initialized when it is used by name at vim@@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 3496.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	1099	3725
Object	cmdline	name

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   lambda_function_body(

```
....
1099.       char_u *cmdline = NULL;
```

▼

File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c

Method   trans_function_name(

```
....
3725.       name = alloc(len + lead + extra + 1);
```

Use of Zero Initialized Pointer\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=205
Status	New

The variable declared in Pointer at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 652 is not initialized when it is used by nextcmd at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 652.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	844	819
Object	Pointer	nextcmd

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method	get_function_body(

```
....
844.                        *line_to_free = NULL;
....
819.                        nextcmd = p + 1;
```

Use of Zero Initialized Pointer\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=206
Status	New

The variable declared in cmdline at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 1083 is not initialized when it is used by nextcmd at vim@@vim-v8.2.3451-CVE-2021-4173-TP.c in line 652.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	1099	819
Object	cmdline	nextcmd

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method	lambda_function_body(

```
....
1099.      char_u *cmdline = NULL;
```

▼

File Name	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method	get_function_body(

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

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=1050077&projectid=50066&pathid=91
Status	New

The dangerous function, sprintf, was found in use at line 490 in vim@@vim-v8.2.3018-CVE-2022-3278-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	495	495
Object	sprintf	sprintf

Code Snippet

File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	get_lambda_name(void)

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

Dangerous Functions\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=92

Status	New

The dangerous function, sprintf, was found in use at line 1656 in vim@@vim-v8.2.3018-CVE-2022-3278-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	1675	1675
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   fname_trans_sid(char_u *name, char_u *fname_buf, char_u **tofree, int *error)

```
....
1675.              sprintf((char *)fname_buf + 3, "%ld_",
```

Dangerous Functions\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=93
Status	New

The dangerous function, sprintf, was found in use at line 2288 in vim@@vim-v8.2.3018-CVE-2022-3278-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	2475	2475
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   call_user_func(

```
....
2475.              sprintf((char *)numbuf, "%d", ai + 1);
```

Dangerous Functions\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=94
Status	New

The dangerous function, sprintf, was found in use at line 2799 in vim@@vim-v8.2.3018-CVE-2022-3278-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	2810	2810
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   delete_script_functions(int sid)

```
....
2810.        sprintf((char *)buf + 3, "%d_", sid);
```

Dangerous Functions\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=95
Status	New

The dangerous function, sprintf, was found in use at line 3387 in vim@@vim-v8.2.3018-CVE-2022-3278-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	3591	3591
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   trans_function_name(

```
....
3591.            sprintf((char *)sid_buf, "%ld_",
(long)current_sctx.sc_sid);
```

Dangerous Functions\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=96
Status	New

The dangerous function, sprintf, was found in use at line 3700 in vim@@vim-v8.2.3018-CVE-2022-3278-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	4159	4159
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   define_function(exarg_T *eap, char_u *name_arg)

```
....
4159.        sprintf(numbuf, "%d", ++func_nr);
```

Dangerous Functions\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=97
Status	New

The dangerous function, sprintf, was found in use at line 499 in vim@@vim-v8.2.3451-CVE-2021-4173-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	504	504
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   get_lambda_name(void)

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

Dangerous Functions\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=98
Status	New

The dangerous function, sprintf, was found in use at line 1753 in vim@@vim-v8.2.3451-CVE-2021-4173-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	1772	1772
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   fname_trans_sid(char_u *name, char_u *fname_buf, char_u **tofree, int *error)

```
....
1772.              sprintf((char *)fname_buf + 3, "%ld_",
```

Dangerous Functions\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=99
Status	New

The dangerous function, sprintf, was found in use at line 2385 in vim@@vim-v8.2.3451-CVE-2021-4173-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	2572	2572
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   call_user_func(

```
....
2572.              sprintf((char *)numbuf, "%d", ai + 1);
```

Dangerous Functions\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=100
Status	New

The dangerous function, sprintf, was found in use at line 2896 in vim@@vim-v8.2.3451-CVE-2021-4173-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	2907	2907
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   delete_script_functions(int sid)

```
....
2907.        sprintf((char *)buf + 3, "%d_", sid);
```

Dangerous Functions\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=101
Status	New

The dangerous function, sprintf, was found in use at line 3496 in vim@@vim-v8.2.3451-CVE-2021-4173-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	3700	3700
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   trans_function_name(

```
....
3700.            sprintf((char *)sid_buf, "%ld_",
(long)current_sctx.sc_sid);
```

Dangerous Functions\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=102
Status	New

The dangerous function, sprintf, was found in use at line 3810 in vim@@vim-v8.2.3451-CVE-2021-4173-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	4276	4276
Object	sprintf	sprintf

Code Snippet
File Name vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method define_function(exarg_T *eap, char_u *name_arg)

```
....
4276.        sprintf(numbuf, "%d", ++func_nr);
```

Dangerous Functions\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=103
Status	New

The dangerous function, sprintf, was found in use at line 499 in vim@@vim-v8.2.3451-CVE-2021-4187-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Line	504	504
Object	sprintf	sprintf

Code Snippet
File Name vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Method get_lambda_name(void)

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

Dangerous Functions\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=104
Status	New

The dangerous function, sprintf, was found in use at line 1753 in vim@@vim-v8.2.3451-CVE-2021-4187-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Line	1772	1772
Object	sprintf	sprintf

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Method	fname_trans_sid(char_u *name, char_u *fname_buf, char_u **tofree, int *error)

```
....
1772.               sprintf((char *)fname_buf + 3, "%ld_",
```

Dangerous Functions\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=105
Status	New

The dangerous function, sprintf, was found in use at line 2385 in vim@@vim-v8.2.3451-CVE-2021-4187-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Line	2572	2572
Object	sprintf	sprintf

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Method	call_user_func(

```
....
2572.               sprintf((char *)numbuf, "%d", ai + 1);
```

Dangerous Functions\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=106
Status	New

The dangerous function, sprintf, was found in use at line 2896 in vim@@vim-v8.2.3451-CVE-2021-4187-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Line	2907	2907
Object	sprintf	sprintf

Code Snippet
File Name vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Method delete_script_functions(int sid)

```
....
2907.      sprintf((char *)buf + 3, "%d_", sid);
```

Dangerous Functions\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=107
Status	New

The dangerous function, sprintf, was found in use at line 3496 in vim@@vim-v8.2.3451-CVE-2021-4187-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Line	3700	3700
Object	sprintf	sprintf

Code Snippet
File Name vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Method trans_function_name(

```
....
3700.        sprintf((char *)sid_buf, "%ld_",
(long)current_sctx.sc_sid);
```

Dangerous Functions\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=108
Status	New

The dangerous function, sprintf, was found in use at line 3810 in vim@@vim-v8.2.3451-CVE-2021-4187-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Line	4276	4276
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Method   define_function(exarg_T *eap, char_u *name_arg)

```
....
4276.          sprintf(numbuf, "%d", ++func_nr);
```

Dangerous Functions\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=109
Status	New

The dangerous function, sprintf, was found in use at line 822 in vim@@vim-v8.2.3451-CVE-2022-0156-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	911	911
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   write_viminfo_history(FILE *fp, int merge)

```
....
911.                              sprintf(cbuf, "%d", c);
```

Dangerous Functions\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=110
Status	New

The dangerous function, sprintf, was found in use at line 1311 in vim@@vim-v8.2.3451-CVE-2022-0156-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	1388	1388
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   write_viminfo_varlist(FILE *fp)

```
....
1388.                    sprintf((char *)numbuf, "%ld",
```

Dangerous Functions\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=111
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0408-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	1365	1365
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
1365.                    sprintf(changename[depth], "prefix");
```

Dangerous Functions\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=112
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0408-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	1764	1764
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
1764.                          sprintf(changename[depth], "%.*s-%s: split",
```

Dangerous Functions\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=113
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0408-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	1767	1767
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
1767.                          sprintf(changename[depth], "%.*s-%s:
compound",
```

Dangerous Functions\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=114
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0408-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	1913	1913
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

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

Dangerous Functions\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=115
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0408-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	1917	1917
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
1917.                    sprintf(changename[depth], "%.*s-%s: accept %c",
```

Dangerous Functions\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=116
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0408-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	2052	2052
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
2052.               sprintf(changename[depth], "%.*s-%s: delete %c",
```

Dangerous Functions\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=117
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0408-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	2149	2149
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
2149.               sprintf(changename[depth], "%.*s-%s: insert %c",
```

Dangerous Functions\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=118
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0408-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	2249	2249
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
2249.               sprintf(changename[depth], "%.*s-%s: swap %c and %c",
```

Dangerous Functions\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=119
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0408-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	2337	2337
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
2337.               sprintf(changename[depth], "%.*s-%s: swap3 %c and %c",
```

Dangerous Functions\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=120
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0408-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	2405	2405
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
2405.              sprintf(changename[depth], "%.*s-%s: rotate left
%c%c%c",
```

Dangerous Functions\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=121
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0408-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	2466	2466
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
2466.              sprintf(changename[depth], "%.*s-%s: rotate right
%c%c%c",
```

Dangerous Functions\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=122
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0408-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	2578	2578
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
2578.                    sprintf(changename[depth], "%.*s-%s: replace %s
with %s",
```

Dangerous Functions\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=123
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0943-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	1365	1365
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
1365.                    sprintf(changename[depth], "prefix");
```

Dangerous Functions\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=124
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0943-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	1764	1764
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
1764.                    sprintf(changename[depth], "%.*s-%s: split",
```

Dangerous Functions\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=125
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0943-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	1767	1767
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
1767.                    sprintf(changename[depth], "%.*s-%s:
compound",
```

Dangerous Functions\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=126
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0943-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	1913	1913
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

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

Dangerous Functions\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=127
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0943-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	1917	1917
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
1917.                    sprintf(changename[depth], "%.*s-%s: accept %c",
```

Dangerous Functions\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=128
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0943-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	2052	2052
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
2052.                  sprintf(changename[depth], "%.*s-%s: delete %c",
```

Dangerous Functions\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=129
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0943-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	2149	2149
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
2149.                  sprintf(changename[depth], "%.*s-%s: insert %c",
```

Dangerous Functions\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=130
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0943-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	2249	2249
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
2249.                sprintf(changename[depth], "%.*s-%s: swap %c and %c",
```

Dangerous Functions\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=131
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0943-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	2337	2337
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
2337.                sprintf(changename[depth], "%.*s-%s: swap3 %c and %c",
```

Dangerous Functions\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=132
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0943-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	2405	2405
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
2405.             sprintf(changename[depth], "%.*s-%s: rotate left
%c%c%c",
```

Dangerous Functions\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=133
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0943-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	2466	2466
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
2466.             sprintf(changename[depth], "%.*s-%s: rotate right
%c%c%c",
```

Dangerous Functions\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=134
Status	New

The dangerous function, sprintf, was found in use at line 1237 in vim@@vim-v8.2.3451-CVE-2022-0943-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	2578	2578
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
2578.                   sprintf(changename[depth], "%.*s-%s: replace %s
with %s",
```

Dangerous Functions\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=135
Status	New

The dangerous function, sscanf, was found in use at line 2229 in vim@@vim-v8.2.3451-CVE-2022-0156-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	2348	2348
Object	sscanf	sscanf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   copy_viminfo_marks(

```
....
2348.                        sscanf((char *)line + 2, "%ld ", &ltime);
```

Dangerous Functions\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=136
Status	New

The dangerous function, sscanf, was found in use at line 2229 in vim@@vim-v8.2.3451-CVE-2022-0156-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	2407	2407
Object	sscanf	sscanf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   copy_viminfo_marks(

```
....
2407.                    sscanf((char *)line + 2, "%ld %u", &pos.lnum, &u);
```

Dangerous Functions\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=137
Status	New

The dangerous function, strlen, was found in use at line 707 in vim@@vim-v8.2.3451-CVE-2021-3984-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-3984-TP.c	vim@@vim-v8.2.3451-CVE-2021-3984-TP.c
Line	723	723
Object	strlen	strlen

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-3984-TP.c
Method   cin_isinit(void)

```
....
723.             l = (int)strlen(skip[i]);
```

Dangerous Functions\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=138
Status	New

The dangerous function, atoi, was found in use at line 2756 in vim@@vim-v8.2.3018-CVE-2022-3520-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Line	2785	2785
Object	atoi	atoi

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Method   write_reg_contents_ex(

```
....
2785.             int    num = atoi((char *)str);
```

Dangerous Functions\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=139
Status	New

The dangerous function, atoi, was found in use at line 2756 in vim@@vim-v8.2.3018-CVE-2023-1170-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c
Line	2785	2785
Object	atoi	atoi

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-1170-TP.c
Method   write_reg_contents_ex(

```
....
2785.             int    num = atoi((char *)str);
```

Dangerous Functions\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=140
Status	New

The dangerous function, atoi, was found in use at line 2756 in vim@@vim-v8.2.3018-CVE-2023-1175-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Line	2785	2785
Object	atoi	atoi

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Method   write_reg_contents_ex(

```
....
2785.            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=1050077&projectid=50066&pathid=5
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 2836 of vim@@vim-v8.2.3018-CVE-2022-3520-TP.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-v8.2.3018-CVE-2022-3520-TP.c	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Line	2956	2956
Object	AssignExpr	AssignExpr

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Method   str_to_reg(

```
....
2956.            extra += i;
```

Integer Overflow\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=6
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1520 of vim@@vim-v8.2.3018-CVE-2022-3520-TP.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-v8.2.3018-CVE-2022-3520-TP.c	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Line	1878	1878
Object	AssignExpr	AssignExpr

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Method   do_put(

```
....
1878.           totlen = count * (yanklen + spaces) + bd.startspaces +
bd.endspaces;
```

Integer Overflow\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=7
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1520 of vim@@vim-v8.2.3018-CVE-2022-3520-TP.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-v8.2.3018-CVE-2022-3520-TP.c	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Line	2004	2004
Object	AssignExpr	AssignExpr

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Method   do_put(

```
....
2004.           totlen = count * yanklen;
```

Integer Overflow\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=8
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1520 of vim@@vim-v8.2.3018-CVE-2022-3520-TP.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-v8.2.3018-CVE-2022-3520-TP.c	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Line	1870	1870
Object	AssignExpr	AssignExpr

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Method	do_put(

```
....
1870.              spaces = y_width + 1;
```

Integer Overflow\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=9
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 2836 of vim@@vim-v8.2.3018-CVE-2023-1170-TP.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-v8.2.3018-CVE-2023-1170-TP.c	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c
Line	2956	2956
Object	AssignExpr	AssignExpr

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c
Method	str_to_reg(

```
....
2956.           extra += i;
```

Integer Overflow\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=10
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1520 of vim@@vim-v8.2.3018-CVE-2023-1170-TP.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-v8.2.3018-CVE-2023-1170-TP.c	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c
Line	1878	1878
Object	AssignExpr	AssignExpr

Code Snippet

File Name	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c
Method	do_put(

```
....
1878.            totlen = count * (yanklen + spaces) + bd.startspaces +
bd.endspaces;
```

Integer Overflow\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=11
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1520 of vim@@vim-v8.2.3018-CVE-2023-1170-TP.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-v8.2.3018-CVE-2023-1170-TP.c	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c
Line	2004	2004
Object	AssignExpr	AssignExpr

Code Snippet

File Name	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c
Method	do_put(

```
....
2004.            totlen = count * yanklen;
```

Integer Overflow\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=12
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1520 of vim@@vim-v8.2.3018-CVE-2023-1170-TP.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-v8.2.3018-CVE-2023-1170-TP.c	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c
Line	1870	1870
Object	AssignExpr	AssignExpr

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c
Method	do_put(

```
....
1870.               spaces = y_width + 1;
```

Integer Overflow\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=13
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 2836 of vim@@vim-v8.2.3018-CVE-2023-1175-TP.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-v8.2.3018-CVE-2023-1175-TP.c	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Line	2956	2956
Object	AssignExpr	AssignExpr

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Method	str_to_reg(

```
....
2956.             extra += i;
```

Integer Overflow\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=14
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1520 of vim@@vim-v8.2.3018-CVE-2023-1175-TP.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-v8.2.3018-CVE-2023-1175-TP.c	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Line	1878	1878
Object	AssignExpr	AssignExpr

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Method	do_put(

```
....
1878.           totlen = count * (yanklen + spaces) + bd.startspaces +
bd.endspaces;
```

Integer Overflow\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=15
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1520 of vim@@vim-v8.2.3018-CVE-2023-1175-TP.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-v8.2.3018-CVE-2023-1175-TP.c	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Line	2004	2004
Object	AssignExpr	AssignExpr

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Method	do_put(

```
....
2004.               totlen = count * yanklen;
```

Integer Overflow\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=16
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1520 of vim@@vim-v8.2.3018-CVE-2023-1175-TP.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-v8.2.3018-CVE-2023-1175-TP.c	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Line	1870	1870
Object	AssignExpr	AssignExpr

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Method   do_put(

```
....
1870.              spaces = y_width + 1;
```

Integer Overflow\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=17
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 2836 of vim@@vim-v8.2.3018-CVE-2023-2609-TP.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-v8.2.3018-CVE-2023-2609-TP.c	vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Line	2956	2956
Object	AssignExpr	AssignExpr

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Method   str_to_reg(

```
....
2956.              extra += i;
```

Integer Overflow\Path 14:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=18
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1520 of vim@@vim-v8.2.3018-CVE-2023-2609-TP.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-v8.2.3018-CVE-2023-2609-TP.c	vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Line	1878	1878
Object	AssignExpr	AssignExpr

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Method   do_put(

```
....
1878.           totlen = count * (yanklen + spaces) + bd.startspaces +
bd.endspaces;
```

Integer Overflow\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=19
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1520 of vim@@vim-v8.2.3018-CVE-2023-2609-TP.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-v8.2.3018-CVE-2023-2609-TP.c	vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Line	2004	2004
Object	AssignExpr	AssignExpr

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Method   do_put(

```
....
2004.           totlen = count * yanklen;
```

Integer Overflow\Path 16:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=20
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1520 of vim@@vim-v8.2.3018-CVE-2023-2609-TP.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-v8.2.3018-CVE-2023-2609-TP.c	vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Line	1870	1870
Object	AssignExpr	AssignExpr

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Method   do_put(

```
....
1870.              spaces = y_width + 1;
```

Integer Overflow\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=21
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 3060 of vim@@vim-v8.2.3451-CVE-2022-0392-TP.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-v8.2.3451-CVE-2022-0392-TP.c	vim@@vim-v8.2.3451-CVE-2022-0392-TP.c
Line	3065	3065
Object	AssignExpr	AssignExpr

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0392-TP.c
Method   replace_join(

```
....
3065.      for (i = replace_stack_nr; --i >= 0; )
```

Integer Overflow\Path 18:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=22
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 2846 of vim@@vim-v8.2.3451-CVE-2022-0407-TP.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-v8.2.3451-CVE-2022-0407-TP.c	vim@@vim-v8.2.3451-CVE-2022-0407-TP.c
Line	2966	2966
Object	AssignExpr	AssignExpr

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0407-TP.c
Method   str_to_reg(

```
....
2966.           extra += i;
```

Integer Overflow\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=23
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1529 of vim@@vim-v8.2.3451-CVE-2022-0407-TP.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-v8.2.3451-CVE-2022-0407-TP.c	vim@@vim-v8.2.3451-CVE-2022-0407-TP.c
Line	1888	1888
Object	AssignExpr	AssignExpr

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0407-TP.c
Method   do_put(

```
....
1888.           totlen = count * (yanklen + spaces) + bd.startspaces +
bd.endspaces;
```

Integer Overflow\Path 20:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=24
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1529 of vim@@vim-v8.2.3451-CVE-2022-0407-TP.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-v8.2.3451-CVE-2022-0407-TP.c	vim@@vim-v8.2.3451-CVE-2022-0407-TP.c
Line	2014	2014
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.3451-CVE-2022-0407-TP.c
Method do_put(

```
....
2014.            totlen = count * yanklen;
```

Integer Overflow\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=25
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1529 of vim@@vim-v8.2.3451-CVE-2022-0407-TP.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-v8.2.3451-CVE-2022-0407-TP.c	vim@@vim-v8.2.3451-CVE-2022-0407-TP.c
Line	1880	1880
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v8.2.3451-CVE-2022-0407-TP.c
Method do_put(

```
....
1880.            spaces = y_width + 1;
```

MemoryFree on StackVariable

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

MemoryFree on StackVariable\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=41
Status	New

Calling free() (line 4616) on a variable that was not dynamically allocated (line 4616) in file vim@@vim-v8.2.3018-CVE-2022-4293-TP.c may result with a crash.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4650	4650
Object	tempitem	tempitem

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
4650.          free(tempitem);
```

MemoryFree on StackVariable\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=42
Status	New

Calling free() (line 4693) on a variable that was not dynamically allocated (line 4693) in file vim@@vim-v8.2.3018-CVE-2022-4293-TP.c may result with a crash.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4718	4718
Object	tempitem	tempitem

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
4718.          free(tempitem);
```

MemoryFree on StackVariable\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=43
Status	New

Calling free() (line 4616) on a variable that was not dynamically allocated (line 4616) in file vim@@vim-v8.2.3018-CVE-2023-0054-TP.c may result with a crash.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c
Line	4650	4650
Object	tempitem	tempitem

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c
Method	set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
4650.        free(tempitem);
```

MemoryFree on StackVariable\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=44
Status	New

Calling free() (line 4693) on a variable that was not dynamically allocated (line 4693) in file vim@@vim-v8.2.3018-CVE-2023-0054-TP.c may result with a crash.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c
Line	4718	4718
Object	tempitem	tempitem

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c
Method	set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
4718.        free(tempitem);
```

MemoryFree on StackVariable\Path 5:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=45
Status	New

Calling free() (line 4621) on a variable that was not dynamically allocated (line 4621) in file vim@@vim-v8.2.3451-CVE-2021-4136-TP.c may result with a crash.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Line	4655	4655
Object	tempitem	tempitem

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Method	set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
4655.        free(tempitem);
```

MemoryFree on StackVariable\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=46
Status	New

Calling free() (line 4698) on a variable that was not dynamically allocated (line 4698) in file vim@@vim-v8.2.3451-CVE-2021-4136-TP.c may result with a crash.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Line	4723	4723
Object	tempitem	tempitem

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Method	set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
4723.        free(tempitem);
```

MemoryFree on StackVariable\Path 7:

Severity	Medium
Result State	To Verify

Online Results	
Status	New

Calling free() (line 4621) on a variable that was not dynamically allocated (line 4621) in file vim@@vim-v8.2.3451-CVE-2022-0351-FP.c may result with a crash.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0351-FP.c	vim@@vim-v8.2.3451-CVE-2022-0351-FP.c
Line	4655	4655
Object	tempitem	tempitem

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0351-FP.c
Method   set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
4655.        free(tempitem);
```

MemoryFree on StackVariable\Path 8:

Severity	Medium
Result State	To Verify
Online Results	
Status	New

Calling free() (line 4698) on a variable that was not dynamically allocated (line 4698) in file vim@@vim-v8.2.3451-CVE-2022-0351-FP.c may result with a crash.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0351-FP.c	vim@@vim-v8.2.3451-CVE-2022-0351-FP.c
Line	4723	4723
Object	tempitem	tempitem

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0351-FP.c
Method   set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
4723.        free(tempitem);
```

MemoryFree on StackVariable\Path 9:

Severity	Medium
Result State	To Verify
Online Results	

Status	New

Calling free() (line 4621) on a variable that was not dynamically allocated (line 4621) in file vim@@vim-v8.2.3451-CVE-2022-1420-FP.c may result with a crash.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-1420-FP.c	vim@@vim-v8.2.3451-CVE-2022-1420-FP.c
Line	4655	4655
Object	tempitem	tempitem

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-1420-FP.c
Method   set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
4655.        free(tempitem);
```

MemoryFree on StackVariable\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=50
Status	New

Calling free() (line 4698) on a variable that was not dynamically allocated (line 4698) in file vim@@vim-v8.2.3451-CVE-2022-1420-FP.c may result with a crash.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-1420-FP.c	vim@@vim-v8.2.3451-CVE-2022-1420-FP.c
Line	4723	4723
Object	tempitem	tempitem

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-1420-FP.c
Method   set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
4723.        free(tempitem);
```

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=1050077&projectid=50066&pathid=148
Status	New

The variable declared in wp at vim@@@vim-v8.2.3018-CVE-2023-0433-TP.c in line 803 is not initialized when it is used by w_old_cursor_lnum at vim@@@vim-v8.2.3018-CVE-2023-0433-TP.c in line 803.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-0433-TP.c	vim@@vim-v8.2.3018-CVE-2023-0433-TP.c
Line	854	862
Object	wp	w_old_cursor_lnum

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-0433-TP.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=1050077&projectid=50066&pathid=149
Status	New

The variable declared in wp at vim@@@vim-v8.2.3018-CVE-2023-0433-TP.c in line 803 is not initialized when it is used by w_old_cursor_lnum at vim@@@vim-v8.2.3018-CVE-2023-0433-TP.c in line 803.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-0433-TP.c	vim@@vim-v8.2.3018-CVE-2023-0433-TP.c
Line	854	858
Object	wp	w_old_cursor_lnum

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-0433-TP.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=1050077&projectid=50066&pathid=150
Status	New

The variable declared in wp at vim@@vim-v8.2.3018-CVE-2023-0433-TP.c in line 803 is not initialized when it is used by w_old_visual_lnum at vim@@vim-v8.2.3018-CVE-2023-0433-TP.c in line 803.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-0433-TP.c	vim@@vim-v8.2.3018-CVE-2023-0433-TP.c
Line	854	862
Object	wp	w_old_visual_lnum

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2023-0433-TP.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=1050077&projectid=50066&pathid=151
Status	New

The variable declared in wp at vim@@vim-v8.2.3018-CVE-2023-0433-TP.c in line 803 is not initialized when it is used by w_old_cursor_lnum at vim@@vim-v8.2.3018-CVE-2023-0433-TP.c in line 803.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-0433-TP.c	vim@@vim-v8.2.3018-CVE-2023-0433-TP.c
Line	854	863
Object	wp	w_old_cursor_lnum

Code Snippet	

| File Name | vim@@vim-v8.2.3018-CVE-2023-0433-TP.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=1050077&projectid=50066&pathid=152
Status	New

The variable declared in wp at vim@@vim-v8.2.3018-CVE-2023-0433-TP.c in line 803 is not initialized when it is used by w_old_visual_lnum at vim@@vim-v8.2.3018-CVE-2023-0433-TP.c in line 803.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-0433-TP.c	vim@@vim-v8.2.3018-CVE-2023-0433-TP.c
Line	854	865
Object	wp	w_old_visual_lnum

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2023-0433-TP.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=1050077&projectid=50066&pathid=153
Status	New

The variable declared in wp at vim@@vim-v8.2.3451-CVE-2022-0156-TP.c in line 2229 is not initialized when it is used by wp at vim@@vim-v8.2.3451-CVE-2022-0156-TP.c in line 2229.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	2446	2451
Object	wp	wp

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   copy_viminfo_marks(

```
....
2446.              win_T   *wp;
....
2451.                     wp->w_changelistidx = curbuf->b_changelistlen;
```

Use of Uninitialized Pointer\Path 7:

The variable declared in wp at vim@@vim-v8.2.3451-CVE-2022-0156-TP.c in line 2229 is not initialized when it is used by w_buffer at vim@@vim-v8.2.3451-CVE-2022-0156-TP.c in line 2229.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	2446	2450
Object	wp	w_buffer

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   copy_viminfo_marks(

```
....
2446.              win_T   *wp;
....
2450.                     if (wp->w_buffer == curbuf)
```

Divide By Zero

Query Path:
CPP\Cx\CPP Medium Threat\Divide By Zero Version:1
Description

Divide By Zero\Path 1:

The application performs an illegal operation in parse_cino, in vim@@vim-v8.2.3451-CVE-2021-3984-TP.c. In line 1709, the program attempts to divide by divider, 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 divider in parse_cino of vim@@vim-v8.2.3451-CVE-2021-3984-TP.c, at line 1709.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-3984-TP.c	vim@@vim-v8.2.3451-CVE-2021-3984-TP.c
Line	1879	1879
Object	divider	divider

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-3984-TP.c
Method   parse_cino(buf_T *buf)

```
....
1879.                   n += (sw * fraction + divider / 2) / divider;
```

Divide By Zero\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=27
Status	New

The application performs an illegal operation in num_divide, in vim@@vim-v8.2.3451-CVE-2021-4136-TP.c. In line 66, the program attempts to divide by n2, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input n2 in num_divide of vim@@vim-v8.2.3451-CVE-2021-4136-TP.c, at line 66.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Line	86	86
Object	n2	n2

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Method   num_divide(varnumber_T n1, varnumber_T n2, int *failed)

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

Divide By Zero\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=28
Status	New

The application performs an illegal operation in num_divide, in vim@@vim-v8.2.3451-CVE-2022-0351-FP.c. In line 66, the program attempts to divide by n2, which might be evaluate to 0 (zero) at time of division. This

value could be a hard-coded zero value, or received from external, untrusted input n2 in num_divide of vim@@vim-v8.2.3451-CVE-2022-0351-FP.c, at line 66.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0351-FP.c	vim@@vim-v8.2.3451-CVE-2022-0351-FP.c
Line	86	86
Object	n2	n2

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0351-FP.c
Method   num_divide(varnumber_T n1, varnumber_T n2, int *failed)

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

Divide By Zero\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=29
Status	New

The application performs an illegal operation in tabstop_padding, in vim@@vim-v8.2.3451-CVE-2022-0572-TP.c. In line 97, 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-v8.2.3451-CVE-2022-0572-TP.c, at line 97.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0572-TP.c	vim@@vim-v8.2.3451-CVE-2022-0572-TP.c
Line	106	106
Object	ts	ts

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0572-TP.c
Method   tabstop_padding(colnr_T col, int ts_arg, int *vts)

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

Divide By Zero\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=30
Status	New

The application performs an illegal operation in tabstop_padding, in vim@@vim-v8.2.3451-CVE-2022-0714-TP.c. In line 97, 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-v8.2.3451-CVE-2022-0714-TP.c, at line 97.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0714-TP.c	vim@@vim-v8.2.3451-CVE-2022-0714-TP.c
Line	106	106
Object	ts	ts

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0714-TP.c
Method   tabstop_padding(colnr_T col, int ts_arg, int *vts)

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

Divide By Zero\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=31
Status	New

The application performs an illegal operation in num_divide, in vim@@vim-v8.2.3451-CVE-2022-1420-FP.c. In line 66, the program attempts to divide by n2, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input n2 in num_divide of vim@@vim-v8.2.3451-CVE-2022-1420-FP.c, at line 66.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-1420-FP.c	vim@@vim-v8.2.3451-CVE-2022-1420-FP.c
Line	86	86
Object	n2	n2

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-1420-FP.c
Method   num_divide(varnumber_T n1, varnumber_T n2, int *failed)

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

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

Categories

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

Description

Char Overflow\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=2
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 614 of vim@@vim-v8.2.3018-CVE-2022-3278-TP.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-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	652	652
Object	AssignExpr	AssignExpr

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	get_function_body(

```
....
652.        nesting_def[nesting] = vim9_function;
```

Char Overflow\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=3
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 652 of vim@@vim-v8.2.3451-CVE-2021-4173-TP.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-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	691	691
Object	AssignExpr	AssignExpr

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method	get_function_body(

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

Char Overflow\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=4
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 652 of vim@@vim-v8.2.3451-CVE-2021-4187-TP.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-v8.2.3451-CVE-2021-4187-TP.c	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Line	691	691
Object	AssignExpr	AssignExpr

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Method   get_function_body(

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

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=1050077&projectid=50066&pathid=155
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0572-TP.c	vim@@vim-v8.2.3451-CVE-2022-0572-TP.c
Line	1437	1566
Object	ind_len	ind_len

Code Snippet

File Name   vim@@vim-v8.2.3451-CVE-2022-0572-TP.c
Method   copy_indent(int size, char_u *src)

```
....
1437.      int          ind_len;
....
1566.      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=1050077&projectid=50066&pathid=156
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0714-TP.c	vim@@vim-v8.2.3451-CVE-2022-0714-TP.c
Line	1437	1566
Object	ind_len	ind_len

Code Snippet

File Name   vim@@vim-v8.2.3451-CVE-2022-0714-TP.c
Method   copy_indent(int size, char_u *src)

```
....
1437.      int          ind_len;
....
1566.      curwin->w_cursor.col = ind_len;
```

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=1050077&projectid=50066&pathid=1
Status	New

The pointer trypos at vim@@@vim-v8.2.3451-CVE-2021-3984-TP.c in line 1638 is being used after it has been freed.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-3984-TP.c	vim@@vim-v8.2.3451-CVE-2021-3984-TP.c
Line	1660	1660
Object	trypos	trypos

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-3984-TP.c
Method   find_start_brace(void)   // XXX

```
....
1660.        return trypos;
```

Memory Leak

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=1050077&projectid=50066&pathid=147
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-0433-TP.c	vim@@vim-v8.2.3018-CVE-2023-0433-TP.c
Line	627	627
Object	neW	neW

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-0433-TP.c
Method   auto_format(

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

NULL Pointer Dereference

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=1050077&projectid=50066&pathid=478
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614 is not initialized when it is used by eap at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	801	687
Object	null	eap

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	get_function_body(

```
....
801.                          *line_to_free = NULL;
....
687.              theline = eap->getline(':', eap->cookie, indent,
```

NULL Pointer Dereference\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=479
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 999 is not initialized when it is used by eap at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	1017	687
Object	null	eap

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method	lambda_function_body(

```
....
1017.      char_u *line_to_free = NULL;
```

▼

File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	get_function_body(

```
....
687.            theline = eap->getline(':', eap->cookie, indent,
```

NULL Pointer Dereference\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=480
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3700 is not initialized when it is used by eap at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	3702	687
Object	null	eap

Code Snippet
File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	define_function(exarg_T *eap, char_u *name_arg)

```
....
3702.      char_u *line_to_free = NULL;
```

▼

File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	get_function_body(

```
....
687.            theline = eap->getline(':', eap->cookie, indent,
```

NULL Pointer Dereference\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=481

Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 4478 is not initialized when it is used by fdp at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3387.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	4485	3478
Object	null	fdp

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   get_expanded_name(char_u *name, int check)

```
....
4485.                        TFN_INT|TFN_QUIET, NULL, NULL,
NULL);
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   trans_function_name(

```
....
3478.                        || lv.ll_dict == NULL || fdp->fd_newkey ==
NULL))
```

NULL Pointer Dereference\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=482
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 614 is not initialized when it is used by fdp at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3387.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	841	3478
Object	null	fdp

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   get_function_body(

```
....
841.                 vim_free(trans_function_name(&p, NULL, TRUE, 0, NULL,
```

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   trans_function_name(

```
....
3478.                          || lv.ll_dict == NULL || fdp->fd_newkey ==
NULL))
```

NULL Pointer Dereference\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=483
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 4454 is not initialized when it is used by fdp at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3387.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	4465	3478
Object	null	fdp

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   function_exists(char_u *name, int no_deref)

```
....
4465.      p = trans_function_name(&nm, &is_global, FALSE, flag, NULL,
NULL, NULL);
```

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   trans_function_name(

```
....
3478.                          || lv.ll_dict == NULL || fdp->fd_newkey ==
NULL))
```

NULL Pointer Dereference\Path 7:

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

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=484
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 1186 is not initialized when it is used by argtypes at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 365.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	1274	418
Object	null	argtypes

Code Snippet

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   get_lambda_tv(

```
....
1274.                    types_optional ? &argtypes : NULL, varargs,
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   parse_argument_types(ufunc_T *fp, garray_T *argtypes, int varargs)

```
....
418.                         ((char_u **)argtypes-
>ga_data)[len]);
```

NULL Pointer Dereference\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=485
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 2156 is not initialized when it is used by fp at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 2156.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	2189	2189
Object	null	fp

Code Snippet

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   copy_func(char_u *lambda, char_u *global, ectx_T *ectx)

```
....
2189.           fp->uf_name_exp = ufunc->uf_name_exp == NULL ? NULL
```

NULL Pointer Dereference\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=486
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3700 is not initialized when it is used by uf_args at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 365.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	4115	407
Object	null	uf_args

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   define_function(exarg_T *eap, char_u *name_arg)

```
....
4115.              fp = NULL;
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   parse_argument_types(ufunc_T *fp, garray_T *argtypes, int varargs)

```
....
407.          --fp->uf_args.ga_len;
```

NULL Pointer Dereference\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=487
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3700 is not initialized when it is used by uf_args at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 365.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Line	4140	407
Object	null	uf_args

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   define_function(exarg_T *eap, char_u *name_arg)

```
....
4140.        fp = NULL;
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   parse_argument_types(ufunc_T *fp, garray_T *argtypes, int varargs)

```
....
407.          --fp->uf_args.ga_len;
```

NULL Pointer Dereference\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=488
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3700 is not initialized when it is used by uf_args at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 365.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	4243	407
Object	null	uf_args

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   define_function(exarg_T *eap, char_u *name_arg)

```
....
4243.            fp = NULL;
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   parse_argument_types(ufunc_T *fp, garray_T *argtypes, int varargs)

```
....
407.        --fp->uf_args.ga_len;
```

NULL Pointer Dereference\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=489
Status	New

The variable declared in null at vim@@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3700 is not initialized when it is used by uf_args at vim@@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 365.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	4219	407
Object	null	uf_args

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   define_function(exarg_T *eap, char_u *name_arg)

```
....
4219.                    fp = NULL;
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   parse_argument_types(ufunc_T *fp, garray_T *argtypes, int varargs)

```
....
407.        --fp->uf_args.ga_len;
```

NULL Pointer Dereference\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=490
Status	New

The variable declared in null at vim@@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3700 is not initialized when it is used by uf_args at vim@@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 365.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	4212	407
Object	null	uf_args

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	define_function(exarg_T *eap, char_u *name_arg)

```
....
4212.                  fp = NULL;
```

▼

File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	parse_argument_types(ufunc_T *fp, garray_T *argtypes, int varargs)

```
....
407.          --fp->uf_args.ga_len;
```

NULL Pointer Dereference\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=491
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3700 is not initialized when it is used by fp at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 365.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	4115	369
Object	null	fp

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	define_function(exarg_T *eap, char_u *name_arg)

```
....
4115.                  fp = NULL;
```

▼

File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	parse_argument_types(ufunc_T *fp, garray_T *argtypes, int varargs)

```
....
369.      ga_init2(&fp->uf_type_list, sizeof(type_T *), 10);
```

NULL Pointer Dereference\Path 15:

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

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3700 is not initialized when it is used by fp at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 365.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	4140	369
Object	null	fp

Code Snippet

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   define_function(exarg_T *eap, char_u *name_arg)

```
....
4140.        fp = NULL;
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   parse_argument_types(ufunc_T *fp, garray_T *argtypes, int varargs)

```
....
369.        ga_init2(&fp->uf_type_list, sizeof(type_T *), 10);
```

NULL Pointer Dereference\Path 16:

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3700 is not initialized when it is used by fp at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 365.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	4243	369
Object	null	fp

Code Snippet

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   define_function(exarg_T *eap, char_u *name_arg)

```
....
4243.                fp = NULL;
```

▼

File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	parse_argument_types(ufunc_T *fp, garray_T *argtypes, int varargs)

```
....
369.        ga_init2(&fp->uf_type_list, sizeof(type_T *), 10);
```

NULL Pointer Dereference\Path 17:

Severity	Low
Result State	To Verify
Online Results	
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3700 is not initialized when it is used by fp at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 365.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	4212	369
Object	null	fp

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	define_function(exarg_T *eap, char_u *name_arg)

```
....
4212.                   fp = NULL;
```

▼

File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	parse_argument_types(ufunc_T *fp, garray_T *argtypes, int varargs)

```
....
369.        ga_init2(&fp->uf_type_list, sizeof(type_T *), 10);
```

NULL Pointer Dereference\Path 18:

Severity	Low
Result State	To Verify
Online Results	
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3700 is not initialized when it is used by fp at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 365.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	4219	369
Object	null	fp

Code Snippet

File Name vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method define_function(exarg_T *eap, char_u *name_arg)

```
....
4219.                    fp = NULL;
```

▼

File Name vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method parse_argument_types(ufunc_T *fp, garray_T *argtypes, int varargs)

```
....
369.         ga_init2(&fp->uf_type_list, sizeof(type_T *), 10);
```

NULL Pointer Dereference\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=496
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3700 is not initialized when it is used by fp at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 469.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	4219	473
Object	null	fp

Code Snippet

File Name vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method define_function(exarg_T *eap, char_u *name_arg)

```
....
4219.                    fp = NULL;
```

File Name	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method	set_ufunc_name(ufunc_T *fp, char_u *name)

```
....
473.        STRCPY((void *)fp->uf_name, name);
```

NULL Pointer Dereference\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=497
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 3700 is not initialized when it is used by fp at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 469.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	4212	473
Object	null	fp

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   define_function(exarg_T *eap, char_u *name_arg)

```
....
4212.                    fp = NULL;
```

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   set_ufunc_name(ufunc_T *fp, char_u *name)

```
....
473.        STRCPY((void *)fp->uf_name, name);
```

NULL Pointer Dereference\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=498
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 4546 is not initialized when it is used by fp at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 2032.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	4548	2052
Object	null	fp

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   ex_delfunction(exarg_T *eap)

```
....
4548.        ufunc_T        *fp = NULL;
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   func_remove(ufunc_T *fp)

```
....
2052.         fp->uf_flags |= FC_DELETED;
```

NULL Pointer Dereference\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=499
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 4711 is not initialized when it is used by current_funccal at vim@@vim-v8.2.3018-CVE-2022-3278-TP.c in line 4936.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	4750	5004
Object	null	current_funccal

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   ex_return(exarg_T *eap)

```
....
4750.            returning = do_return(eap, FALSE, TRUE, NULL);
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c

Method   do_return(

```
....
5004.              *current_funccal->rettv = *(typval_T *)rettv;
```

NULL Pointer Dereference\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=500
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 243 is not initialized when it is used by eap at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 200.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	295	208
Object	null	eap

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   eval_expr_typval(typval_T *expr, typval_T *argv, int argc, typval_T *rettv)

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

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method   eval1_emsg(char_u **arg, typval_T *rettv, exarg_T *eap)

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

NULL Pointer Dereference\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=501
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 382 is not initialized when it is used by gap at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 382.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-	vim@@vim-v8.2.3018-CVE-2022-4293-

	TP.c	TP.c
Line	439	439
Object	null	gap

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   skip_expr_concatenate(

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

NULL Pointer Dereference\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=502
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 540 is not initialized when it is used by eap at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 517.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	544	526
Object	null	eap

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   eval_to_string(

```
....
544.      return eval_to_string_eap(arg, convert, NULL);
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method   eval_to_string_eap(

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

NULL Pointer Dereference\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by ht_stack at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4456	4761
Object	null	ht_stack

Code Snippet

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   garbage_collect(int testing)

```
....
4456.                                                    NULL, NULL);
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method   set_ref_in_item(

```
....
4761.                 newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 27:

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by ht_stack at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4461	4761
Object	null	ht_stack

Code Snippet

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   garbage_collect(int testing)

```
             ....
4461.                                               NULL, NULL);
```

▼

File Name	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method	set_ref_in_item(

```
             ....
4761.                    newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=505
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by ht_stack at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4464	4761
Object	null	ht_stack

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method	garbage_collect(int testing)

```
             ....
4464.                                               NULL, NULL);
```

▼

File Name	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method	set_ref_in_item(

```
             ....
4761.                    newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=506
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by ht_stack at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4468	4761
Object	null	ht_stack

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   garbage_collect(int testing)

```
....
4468.                                           NULL, NULL);
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method   set_ref_in_item(

```
....
4761.               newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=507
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by ht_stack at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4472	4761
Object	null	ht_stack

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   garbage_collect(int testing)

```
....
4472.                                           NULL, NULL);
```

File Name	vim@@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method	set_ref_in_item(

```
....
4761.                    newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=508
Status	New

The variable declared in null at vim@@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by ht_stack at vim@@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4478	4761
Object	null	ht_stack

Code Snippet
File Name   vim@@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   garbage_collect(int testing)

```
....
4478.                                        NULL, NULL);
```

File Name	vim@@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method	set_ref_in_item(

```
....
4761.              newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=509
Status	New

The variable declared in null at vim@@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by newitem at vim@@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4461	4761
Object	null	newitem

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   garbage_collect(int testing)

```
....
4461.                                          NULL, NULL);
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method   set_ref_in_item(

```
....
4761.               newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=510
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by newitem at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4456	4761
Object	null	newitem

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   garbage_collect(int testing)

```
....
4456.                                          NULL, NULL);
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method   set_ref_in_item(

```
....
4761.                       newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=511
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by newitem at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4464	4761
Object	null	newitem

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   garbage_collect(int testing)

```
....
4464.                                              NULL, NULL);
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method   set_ref_in_item(

```
....
4761.                       newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=512
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by newitem at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Line	4468	4761
Object	null	newitem

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   garbage_collect(int testing)

```
....
4468.                                          NULL, NULL);
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method   set_ref_in_item(

```
....
4761.                newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=513
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by newitem at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4478	4761
Object	null	newitem

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   garbage_collect(int testing)

```
....
4478.                                          NULL, NULL);
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method   set_ref_in_item(

```
....
4761.                newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=514
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by newitem at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4472	4761
Object	null	newitem

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method	garbage_collect(int testing)

```
....
4472.                                    NULL, NULL);
```

▼

File Name	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method	set_ref_in_item(

```
....
4761.              newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=515
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by list_stack at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4456	4788
Object	null	list_stack

Code Snippet

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method   garbage_collect(int testing)

```
....
4456.                                              NULL, NULL);
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method   set_ref_in_item(

```
....
4788.                 newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=516
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by list_stack at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4461	4788
Object	null	list_stack

Code Snippet

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method   garbage_collect(int testing)

```
....
4461.                                              NULL, NULL);
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method   set_ref_in_item(

```
....
4788.                 newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 40:

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

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=517
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by list_stack at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4464	4788
Object	null	list_stack

Code Snippet

File Name vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method garbage_collect(int testing)

```
....
4464.                                        NULL, NULL);
```

▼

File Name vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method set_ref_in_item(

```
....
4788.                 newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=518
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by list_stack at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4468	4788
Object	null	list_stack

Code Snippet

File Name vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method garbage_collect(int testing)

```
....
4468.                                              NULL, NULL);
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method   set_ref_in_item(

```
....
4788.                      newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=519
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by list_stack at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4472	4788
Object	null	list_stack

Code Snippet

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   garbage_collect(int testing)

```
....
4472.                                              NULL, NULL);
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method   set_ref_in_item(

```
....
4788.                      newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=520
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by list_stack at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4478	4788
Object	null	list_stack

Code Snippet

File Name	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method	garbage_collect(int testing)

```
....
4478.                                              NULL, NULL);
```

▼

File Name	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method	set_ref_in_item(

```
....
4788.                 newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=521
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by newitem at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4456	4788
Object	null	newitem

Code Snippet

File Name	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method	garbage_collect(int testing)

```
....
4456.                                              NULL, NULL);
```

File Name	vim@@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method	set_ref_in_item(

```
....
4788.                    newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 45:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=522
Status	New

The variable declared in null at vim@@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by newitem at vim@@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4461	4788
Object	null	newitem

Code Snippet

File Name	vim@@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method	garbage_collect(int testing)

```
....
4461.                                          NULL, NULL);
```

File Name	vim@@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method	set_ref_in_item(

```
....
4788.               newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=523
Status	New

The variable declared in null at vim@@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by newitem at vim@@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4464	4788
Object	null	newitem

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   garbage_collect(int testing)

```
....
4464.                                              NULL, NULL);
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method   set_ref_in_item(

```
....
4788.                  newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=524
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by newitem at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4468	4788
Object	null	newitem

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   garbage_collect(int testing)

```
....
4468.                                              NULL, NULL);
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method   set_ref_in_item(

```
....
4788.                      newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 48:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=525
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by newitem at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	4478	4788
Object	null	newitem

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   garbage_collect(int testing)

```
....
4478.                                       NULL, NULL);
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method   set_ref_in_item(

```
....
4788.                 newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 49:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=526
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4395 is not initialized when it is used by newitem at vim@@vim-v8.2.3018-CVE-2022-4293-TP.c in line 4732.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Line	4472	4788
Object	null	newitem

Code Snippet

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method   garbage_collect(int testing)

```
....
4472.                                            NULL, NULL);
```

▼

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c

Method   set_ref_in_item(

```
....
4788.              newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=527
Status	New

The variable declared in null at vim@@vim-v8.2.3018-CVE-2023-0054-TP.c in line 243 is not initialized when it is used by eap at vim@@vim-v8.2.3018-CVE-2023-0054-TP.c in line 200.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c
Line	295	208
Object	null	eap

Code Snippet

File Name   vim@@vim-v8.2.3018-CVE-2023-0054-TP.c

Method   eval_expr_typval(typval_T *expr, typval_T *argv, int argc, typval_T *rettv)

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

▼

File Name   vim@@vim-v8.2.3018-CVE-2023-0054-TP.c

Method   eval1_emsg(char_u **arg, typval_T *rettv, exarg_T *eap)

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

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=1050077&projectid=50066&pathid=403
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	198	198
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   get_function_args(

```
....
198.         ga_init2(newargs, (int)sizeof(char_u *), 3);
```

Use of Sizeof On a Pointer Type\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=404
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	200	200
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   get_function_args(

```
....
200.         ga_init2(argtypes, (int)sizeof(char_u *), 3);
```

Use of Sizeof On a Pointer Type\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=405
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	202	202
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   get_function_args(

```
....
202.         ga_init2(default_args, (int)sizeof(char_u *), 3);
```

Use of Sizeof On a Pointer Type\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=406
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	369	369
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   parse_argument_types(ufunc_T *fp, garray_T *argtypes, int varargs)

```
....
369.         ga_init2(&fp->uf_type_list, sizeof(type_T *), 10);
```

Use of Sizeof On a Pointer Type\Path 5:

Severity	Low

	Source	Destination

Result State · To Verify
Online Results · http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=407
Status · New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	1042	1042
Object	sizeof	sizeof

Code Snippet
File Name · vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method · lambda_function_body(

```
....
1042.        ga_init2(&newlines, (int)sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 6:

Severity · Low
Result State · To Verify
Online Results · http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=408
Status · New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	1324	1324
Object	sizeof	sizeof

Code Snippet
File Name · vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method · get_lambda_tv(

```
....
1324.        ga_init2(&newlines, (int)sizeof(char_u *), 1);
```

Use of Sizeof On a Pointer Type\Path 7:

Severity · Low
Result State · To Verify
Online Results · http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=409
Status · New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	1610	1610
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   get_func_tv(

```
....
1610.                   ga_init2(&funcargs, (int)sizeof(typval_T *), 50);
```

Use of Sizeof On a Pointer Type\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=410
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	2197	2197
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   copy_func(char_u *lambda, char_u *global, ectx_T *ectx)

```
....
2197.                             sizeof(type_T *) * fp-
>uf_args.ga_len);
```

Use of Sizeof On a Pointer Type\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=411
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	2341	2341

Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   call_user_func(

```
....
2341.       ga_init2(&fc->fc_funcs, sizeof(ufunc_T *), 1);
```

Use of Sizeof On a Pointer Type\Path 10:

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	3919	3919
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   define_function(exarg_T *eap, char_u *name_arg)

```
....
3919.       ga_init2(&newlines, (int)sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 11:

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Line	1185	1185
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Method   op_yank(oparg_T *oap, int deleting, int mess)

```
....
1185.       y_current->y_array = lalloc_clear(sizeof(char_u *) *
yanklines, TRUE);
```

Use of Sizeof On a Pointer Type\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=414
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Line	1496	1496
Object	sizeof	sizeof

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Method	copy_yank_reg(yankreg_T *reg)

```
....
1496.                          sizeof(char_u *) * y_current->y_size,
TRUE);
```

Use of Sizeof On a Pointer Type\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=415
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Line	2900	2900
Object	sizeof	sizeof

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Method	str_to_reg(

```
....
2900.       pp = lalloc_clear((y_ptr->y_size + newlines) * sizeof(char_u
*), TRUE);
```

Use of Sizeof On a Pointer Type\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=416
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	400	400
Object	sizeof	sizeof

Code Snippet

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   skip_expr_concatenate(

```
....
400.          ga_init2(gap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=417
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	404	404
Object	sizeof	sizeof

Code Snippet

File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   skip_expr_concatenate(

```
....
404.          ga_init2(freegap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=418

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c	vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Line	3277	3277
Object	sizeof	sizeof

Status New

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-4293-TP.c
Method   eval7t(

```
....
3277.        ga_init2(&type_list, sizeof(type_T *), 10);
```

Use of Sizeof On a Pointer Type\Path 17:

Severity   Low
Result State   To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=419
Status   New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c
Line	400	400
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-0054-TP.c
Method   skip_expr_concatenate(

```
....
400.         ga_init2(gap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 18:

Severity   Low
Result State   To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=420
Status   New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c

Line	404	404
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-0054-TP.c
Method   skip_expr_concatenate(

```
....
404.          ga_init2(freegap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=421
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c	vim@@vim-v8.2.3018-CVE-2023-0054-TP.c
Line	3277	3277
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-0054-TP.c
Method   eval7t(

```
....
3277.          ga_init2(&type_list, sizeof(type_T *), 10);
```

Use of Sizeof On a Pointer Type\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=422
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c
Line	1185	1185
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-1170-TP.c

Method	op_yank(oparg_T *oap, int deleting, int mess)

```
....
1185.      y_current->y_array = lalloc_clear(sizeof(char_u *) *
yanklines, TRUE);
```

Use of Sizeof On a Pointer Type\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=423
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c
Line	1496	1496
Object	sizeof	sizeof

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c
Method	copy_yank_reg(yankreg_T *reg)

```
....
1496.                            sizeof(char_u *) * y_current->y_size,
TRUE);
```

Use of Sizeof On a Pointer Type\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=424
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c
Line	2900	2900
Object	sizeof	sizeof

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c
Method	str_to_reg(

```
....
2900.      pp = lalloc_clear((y_ptr->y_size + newlines) * sizeof(char_u
*), TRUE);
```

Use of Sizeof On a Pointer Type\Path 23:

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Line	1185	1185
Object	sizeof	sizeof

Code Snippet

File Name   vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Method   op_yank(oparg_T *oap, int deleting, int mess)

```
....
1185.      y_current->y_array = lalloc_clear(sizeof(char_u *) *
yanklines, TRUE);
```

Use of Sizeof On a Pointer Type\Path 24:

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Line	1496	1496
Object	sizeof	sizeof

Code Snippet

File Name   vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Method   copy_yank_reg(yankreg_T *reg)

```
....
1496.                          sizeof(char_u *) * y_current->y_size,
TRUE);
```

Use of Sizeof On a Pointer Type\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=427
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Line	2900	2900
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Method   str_to_reg(

```
....
2900.      pp = lalloc_clear((y_ptr->y_size + newlines) * sizeof(char_u
*), TRUE);
```

Use of Sizeof On a Pointer Type\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=428
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-2609-TP.c	vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Line	1185	1185
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Method   op_yank(oparg_T *oap, int deleting, int mess)

```
....
1185.      y_current->y_array = lalloc_clear(sizeof(char_u *) *
yanklines, TRUE);
```

Use of Sizeof On a Pointer Type\Path 27:

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

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=429
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-2609-TP.c	vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Line	1496	1496
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Method   copy_yank_reg(yankreg_T *reg)

```
....
1496.                          sizeof(char_u *) * y_current->y_size,
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=1050077&projectid=50066&pathid=430
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-2609-TP.c	vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Line	2900	2900
Object	sizeof	sizeof

Code Snippet

File Name vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Method   str_to_reg(

```
....
2900.        pp = lalloc_clear((y_ptr->y_size + newlines) * sizeof(char_u
*), 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=1050077&projectid=50066&pathid=431
Status	New

	Source	Destination

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-2610-TP.c	vim@@vim-v8.2.3018-CVE-2023-2610-TP.c
Line	1386	1386
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-2610-TP.c
Method   cleanup_subexpr(void)

```
....
1386.             vim_memset(rex.reg_startp, 0, sizeof(char_u *) *
NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 30:

Severity	Low
Result State	To Verify
Online Results	
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-2610-TP.c	vim@@vim-v8.2.3018-CVE-2023-2610-TP.c
Line	1387	1387
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-2610-TP.c
Method   cleanup_subexpr(void)

```
....
1387.             vim_memset(rex.reg_endp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 31:

Severity	Low
Result State	To Verify
Online Results	
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-2610-TP.c	vim@@vim-v8.2.3018-CVE-2023-2610-TP.c
Line	1407	1407
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-2610-TP.c
Method   cleanup_zsubexpr(void)

```
....
1407.               vim_memset(reg_startzp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=434
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-2610-TP.c	vim@@vim-v8.2.3018-CVE-2023-2610-TP.c
Line	1408	1408
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-2610-TP.c
Method   cleanup_zsubexpr(void)

```
....
1408.               vim_memset(reg_endzp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=435
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Line	404	404
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Method   skip_expr_concatenate(

```
....
404.        ga_init2(gap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=436
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Line	408	408
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Method   skip_expr_concatenate(

```
....
408.        ga_init2(freegap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=437
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c	vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Line	3274	3274
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4136-TP.c
Method   eval7t(

```
....
3274.       ga_init2(&type_list, sizeof(type_T *), 10);
```

Use of Sizeof On a Pointer Type\Path 36:

Severity	Low

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	198	198
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   get_function_args(

```
....
198.          ga_init2(newargs, (int)sizeof(char_u *), 3);
```

Use of Sizeof On a Pointer Type\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=439
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	200	200
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   get_function_args(

```
....
200.          ga_init2(argtypes, (int)sizeof(char_u *), 3);
```

Use of Sizeof On a Pointer Type\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=440
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	202	202
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   get_function_args(

```
....
202.         ga_init2(default_args, (int)sizeof(char_u *), 3);
```

Use of Sizeof On a Pointer Type\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=441
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	378	378
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   parse_argument_types(ufunc_T *fp, garray_T *argtypes, int varargs)

```
....
378.        ga_init2(&fp->uf_type_list, sizeof(type_T *), 10);
```

Use of Sizeof On a Pointer Type\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=442
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	1126	1126

Object	sizeof	sizeof

Code Snippet
File Name   vim@@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   lambda_function_body(

```
....
1126.        ga_init2(&newlines, (int)sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=443
Status	New

	Source	Destination
File	vim@@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	1410	1410
Object	sizeof	sizeof

Code Snippet
File Name   vim@@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   get_lambda_tv(

```
....
1410.        ga_init2(&newlines, (int)sizeof(char_u *), 1);
```

Use of Sizeof On a Pointer Type\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=444
Status	New

	Source	Destination
File	vim@@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	1700	1700
Object	sizeof	sizeof

Code Snippet
File Name   vim@@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   get_func_tv(

```
....
1700.            ga_init2(&funcargs, (int)sizeof(typval_T *), 50);
```

Use of Sizeof On a Pointer Type\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=445
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	2294	2294
Object	sizeof	sizeof

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method	copy_func(char_u *lambda, char_u *global, ectx_T *ectx)

```
....
2294.                        sizeof(type_T *) * fp-
>uf_args.ga_len);
```

Use of Sizeof On a Pointer Type\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=446
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	2438	2438
Object	sizeof	sizeof

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method	call_user_func(

```
....
2438.        ga_init2(&fc->fc_funcs, sizeof(ufunc_T *), 1);
```

Use of Sizeof On a Pointer Type\Path 45:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=447
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	4029	4029
Object	sizeof	sizeof

Code Snippet

File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   define_function(exarg_T *eap, char_u *name_arg)

```
....
4029.        ga_init2(&newlines, (int)sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=448
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Line	198	198
Object	sizeof	sizeof

Code Snippet

File Name   vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Method   get_function_args(

```
....
198.         ga_init2(newargs, (int)sizeof(char_u *), 3);
```

Use of Sizeof On a Pointer Type\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=449
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Line	200	200
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Method   get_function_args(

```
....
200.        ga_init2(argtypes, (int)sizeof(char_u *), 3);
```

Use of Sizeof On a Pointer Type\Path 48:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=450
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Line	202	202
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Method   get_function_args(

```
....
202.        ga_init2(default_args, (int)sizeof(char_u *), 3);
```

Use of Sizeof On a Pointer Type\Path 49:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=451
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Line	378	378

Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Method   parse_argument_types(ufunc_T *fp, garray_T *argtypes, int varargs)

```
....
378.        ga_init2(&fp->uf_type_list, sizeof(type_T *), 10);
```

Use of Sizeof On a Pointer Type\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=452
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Line	1126	1126
Object	sizeof	sizeof

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Method   lambda_function_body(

```
....
1126.       ga_init2(&newlines, (int)sizeof(char_u *), 10);
```

Unchecked Return Value

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=1050077&projectid=50066&pathid=359
Status	New

The get_lambda_name method calls the sprintf function, at line 490 of vim@@vim-v8.2.3018-CVE-2022-3278-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	495	495
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   get_lambda_name(void)

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

Unchecked Return Value\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=360
Status	New

The fname_trans_sid method calls the sprintf function, at line 1656 of vim@@vim-v8.2.3018-CVE-2022-3278-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	1675	1675
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   fname_trans_sid(char_u *name, char_u *fname_buf, char_u **tofree, int *error)

```
....
1675.              sprintf((char *)fname_buf + 3, "%ld_",
```

Unchecked Return Value\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=361
Status	New

The call_user_func method calls the sprintf function, at line 2288 of vim@@vim-v8.2.3018-CVE-2022-3278-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	2475	2475
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   call_user_func(

```
....
2475.            sprintf((char *)numbuf, "%d", ai + 1);
```

Unchecked Return Value\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=362
Status	New

The delete_script_functions method calls the sprintf function, at line 2799 of vim@@vim-v8.2.3018-CVE-2022-3278-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	2810	2810
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   delete_script_functions(int sid)

```
....
2810.        sprintf((char *)buf + 3, "%d_", sid);
```

Unchecked Return Value\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=363
Status	New

The trans_function_name method calls the sprintf function, at line 3387 of vim@@vim-v8.2.3018-CVE-2022-3278-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	3591	3591
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   trans_function_name(

```
....
3591.           sprintf((char *)sid_buf, "%ld_",
(long)current_sctx.sc_sid);
```

Unchecked Return Value\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=364
Status	New

The define_function method calls the sprintf function, at line 3700 of vim@@vim-v8.2.3018-CVE-2022-3278-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	4159	4159
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   define_function(exarg_T *eap, char_u *name_arg)

```
....
4159.         sprintf(numbuf, "%d", ++func_nr);
```

Unchecked Return Value\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=365
Status	New

The get_lambda_name method calls the sprintf function, at line 499 of vim@@vim-v8.2.3451-CVE-2021-4173-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	504	504
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   get_lambda_name(void)

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

Unchecked Return Value\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=366
Status	New

The fname_trans_sid method calls the sprintf function, at line 1753 of vim@@vim-v8.2.3451-CVE-2021-4173-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	1772	1772
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   fname_trans_sid(char_u *name, char_u *fname_buf, char_u **tofree, int *error)

```
....
1772.            sprintf((char *)fname_buf + 3, "%ld_",
```

Unchecked Return Value\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=367
Status	New

The call_user_func method calls the sprintf function, at line 2385 of vim@@vim-v8.2.3451-CVE-2021-4173-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	2572	2572
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   call_user_func(

```
....
2572.            sprintf((char *)numbuf, "%d", ai + 1);
```

Unchecked Return Value\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=368
Status	New

The delete_script_functions method calls the sprintf function, at line 2896 of vim@@vim-v8.2.3451-CVE-2021-4173-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	2907	2907
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   delete_script_functions(int sid)

```
....
2907.       sprintf((char *)buf + 3, "%d_", sid);
```

Unchecked Return Value\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=369

Status	New

The trans_function_name method calls the sprintf function, at line 3496 of vim@@vim-v8.2.3451-CVE-2021-4173-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	3700	3700
Object	sprintf	sprintf

Code Snippet
File Name vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method trans_function_name(

```
....
3700.            sprintf((char *)sid_buf, "%ld_",
(long)current_sctx.sc_sid);
```

Unchecked Return Value\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=370
Status	New

The define_function method calls the sprintf function, at line 3810 of vim@@vim-v8.2.3451-CVE-2021-4173-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	4276	4276
Object	sprintf	sprintf

Code Snippet
File Name vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method define_function(exarg_T *eap, char_u *name_arg)

```
....
4276.         sprintf(numbuf, "%d", ++func_nr);
```

Unchecked Return Value\Path 13:

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

Status	New

The get_lambda_name method calls the sprintf function, at line 499 of vim@@vim-v8.2.3451-CVE-2021-4187-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Line	504	504
Object	sprintf	sprintf

Code Snippet

File Name	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Method	get_lambda_name(void)

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

Unchecked Return Value\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=372
Status	New

The fname_trans_sid method calls the sprintf function, at line 1753 of vim@@vim-v8.2.3451-CVE-2021-4187-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Line	1772	1772
Object	sprintf	sprintf

Code Snippet

File Name	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Method	fname_trans_sid(char_u *name, char_u *fname_buf, char_u **tofree, int *error)

```
....
1772.              sprintf((char *)fname_buf + 3, "%ld_",
```

Unchecked Return Value\Path 15:

Severity	Low
Result State	To Verify

Online Results	
Status	New

The call_user_func method calls the sprintf function, at line 2385 of vim@@vim-v8.2.3451-CVE-2021-4187-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Line	2572	2572
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Method   call_user_func(

```
....
2572.              sprintf((char *)numbuf, "%d", ai + 1);
```

Unchecked Return Value\Path 16:

Severity	Low
Result State	To Verify
Online Results	
Status	New

The delete_script_functions method calls the sprintf function, at line 2896 of vim@@vim-v8.2.3451-CVE-2021-4187-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Line	2907	2907
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Method   delete_script_functions(int sid)

```
....
2907.      sprintf((char *)buf + 3, "%d_", sid);
```

Unchecked Return Value\Path 17:

Severity	Low

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=375
Status	New

The trans_function_name method calls the sprintf function, at line 3496 of vim@@vim-v8.2.3451-CVE-2021-4187-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Line	3700	3700
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Method   trans_function_name(

```
....
3700.          sprintf((char *)sid_buf, "%ld_",
(long)current_sctx.sc_sid);
```

Unchecked Return Value\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=376
Status	New

The define_function method calls the sprintf function, at line 3810 of vim@@vim-v8.2.3451-CVE-2021-4187-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Line	4276	4276
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Method   define_function(exarg_T *eap, char_u *name_arg)

```
....
4276.         sprintf(numbuf, "%d", ++func_nr);
```

Unchecked Return Value\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=377
Status	New

The write_viminfo_history method calls the sprintf function, at line 822 of vim@@vim-v8.2.3451-CVE-2022-0156-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	911	911
Object	sprintf	sprintf

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method	write_viminfo_history(FILE *fp, int merge)

```
....
911.                              sprintf(cbuf, "%d", c);
```

Unchecked Return Value\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=378
Status	New

The write_viminfo_varlist method calls the sprintf function, at line 1311 of vim@@vim-v8.2.3451-CVE-2022-0156-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	1388	1388
Object	sprintf	sprintf

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method	write_viminfo_varlist(FILE *fp)

```
....
1388.                            sprintf((char *)numbuf, "%ld",
```

Unchecked Return Value\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=379
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	1365	1365
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
1365.                    sprintf(changename[depth], "prefix");
```

Unchecked Return Value\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=380
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	1764	1764
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
1764.                          sprintf(changename[depth], "%.*s-%s: split",
```

Unchecked Return Value\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=381
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	1767	1767
Object	sprintf	sprintf

Code Snippet

File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c

Method   suggest_trie_walk(

```
....
1767.                          sprintf(changename[depth], "%.*s-%s:
compound",
```

Unchecked Return Value\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=382
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	1913	1913
Object	sprintf	sprintf

Code Snippet

File Name	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method	suggest_trie_walk(

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

Unchecked Return Value\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=383
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	1917	1917
Object	sprintf	sprintf

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method	suggest_trie_walk(

```
....
1917.                  sprintf(changename[depth], "%.*s-%s: accept %c",
```

Unchecked Return Value\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=384
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	2052	2052
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
2052.                    sprintf(changename[depth], "%.*s-%s: delete %c",
```

Unchecked Return Value\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=385
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	2149	2149
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
2149.                    sprintf(changename[depth], "%.*s-%s: insert %c",
```

Unchecked Return Value\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=386
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	2249	2249

Object	sprintf	sprintf

Code Snippet
File Name vim@@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method suggest_trie_walk(

```
....
2249.            sprintf(changename[depth], "%.*s-%s: swap %c and %c",
```

Unchecked Return Value\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=387
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@@vim-v8.2.3451-CVE-2022-0408-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	2337	2337
Object	sprintf	sprintf

Code Snippet
File Name vim@@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method suggest_trie_walk(

```
....
2337.            sprintf(changename[depth], "%.*s-%s: swap3 %c and %c",
```

Unchecked Return Value\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=388
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@@vim-v8.2.3451-CVE-2022-0408-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@@vim-v8.2.3451-CVE-2022-0408-TP.c

Line	2405	2405
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
2405.                sprintf(changename[depth], "%.*s-%s: rotate left
%c%c%c",
```

Unchecked Return Value\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=389
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	2466	2466
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
2466.                sprintf(changename[depth], "%.*s-%s: rotate right
%c%c%c",
```

Unchecked Return Value\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=390
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

Source	Destination

![Checkmarx logo]

File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	2578	2578
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
2578.                    sprintf(changename[depth], "%.*s-%s: replace %s
with %s",
```

Unchecked Return Value\Path 33:

Severity	Low
Result State	To Verify
Online Results	
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	1365	1365
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
1365.                    sprintf(changename[depth], "prefix");
```

Unchecked Return Value\Path 34:

Severity	Low
Result State	To Verify
Online Results	
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	1764	1764
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
1764.                        sprintf(changename[depth], "%.*s-%s: split",
```

Unchecked Return Value\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=393
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	1767	1767
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
1767.                        sprintf(changename[depth], "%.*s-%s:
compound",
```

Unchecked Return Value\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=394
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	1913	1913
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

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

Unchecked Return Value\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=395
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	1917	1917
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
1917.                    sprintf(changename[depth], "%.*s-%s: accept %c",
```

Unchecked Return Value\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=396

Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	2052	2052
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
2052.                  sprintf(changename[depth], "%.*s-%s: delete %c",
```

Unchecked Return Value\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=397
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	2149	2149
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
2149.                  sprintf(changename[depth], "%.*s-%s: insert %c",
```

Unchecked Return Value\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50

Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	2249	2249
Object	sprintf	sprintf

Code Snippet
File Name vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method suggest_trie_walk(

```
....
2249.                    sprintf(changename[depth], "%.*s-%s: swap %c and %c",
```

Unchecked Return Value\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=399
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	2337	2337
Object	sprintf	sprintf

Code Snippet
File Name vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method suggest_trie_walk(

```
....
2337.                    sprintf(changename[depth], "%.*s-%s: swap3 %c and %c",
```

Unchecked Return Value\Path 42:

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

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	2405	2405
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
2405.              sprintf(changename[depth], "%.*s-%s: rotate left
%c%c%c",
```

Unchecked Return Value\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=401
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	2466	2466
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
2466.              sprintf(changename[depth], "%.*s-%s: rotate right
%c%c%c",
```

Unchecked Return Value\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=402
Status	New

The suggest_trie_walk method calls the sprintf function, at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	2578	2578
Object	sprintf	sprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
2578.                 sprintf(changename[depth], "%.*s-%s: replace %s
with %s",
```

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=1050077&projectid=50066&pathid=321
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-2610-TP.c	vim@@vim-v8.2.3018-CVE-2023-2610-TP.c
Line	2681	2681
Object	fprintf	fprintf

Code Snippet

File Name	vim@@vim-v8.2.3018-CVE-2023-2610-TP.c
Method	vim_regcomp(char_u *expr_arg, int re_flags)

```
....
2681.              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=1050077&projectid=50066&pathid=322
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4192-TP.c	vim@@vim-v8.2.3451-CVE-2021-4192-TP.c
Line	2678	2678
Object	fprintf	fprintf

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2021-4192-TP.c
Method	vim_regcomp(char_u *expr_arg, int re_flags)

```
....
2678.              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=1050077&projectid=50066&pathid=323
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	204	204
Object	fprintf	fprintf

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method	barline_writestring(FILE *fd, char_u *s, int remaining_start)

```
....
204.        fprintf(fd, ">%d\n|<", len);
```

Improper Resource Access Authorization\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=324
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	844	844
Object	fprintf	fprintf

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method	write_viminfo_history(FILE *fp, int merge)

```
....
844.        fprintf(fp, _("\n# %s History (newest to oldest):\n"),
```

Improper Resource Access Authorization\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=325
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	912	912
Object	fprintf	fprintf

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method	write_viminfo_history(FILE *fp, int merge)

```
....
912.                        fprintf(fp, "|%d,%d,%ld,%s,",
BARTYPE_HISTORY,
```

Improper Resource Access Authorization\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=326

	Source	Destination
Status	New	

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	1151	1151
Object	fprintf	fprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   write_viminfo_version(FILE *fp_out)

```
....
1151.        fprintf(fp_out, "# Viminfo version\n|%d,%d\n\n",
```

Improper Resource Access Authorization\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=327
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	1383	1383
Object	fprintf	fprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   write_viminfo_varlist(FILE *fp)

```
....
1383.                fprintf(fp, "!%s\t%s\t", this_var->di_key, s);
```

Improper Resource Access Authorization\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=328
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c

Line	1522	1522
Object	fprintf	fprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   wvsp_one(

```
....
1522.        fprintf(fp, _("\n# Last %sSearch Pattern:\n~"), s);
```

Improper Resource Access Authorization\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=329
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	1524	1524
Object	fprintf	fprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   wvsp_one(

```
....
1524.        fprintf(fp, "%c%c%c%c%ld%s%c",
```

Improper Resource Access Authorization\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=330
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	1542	1542
Object	fprintf	fprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c

Method	write_viminfo_search_pattern(FILE *fp)

```
....
1542.        fprintf(fp, "\n# hlsearch on (H) or off (h):\n~%c",
```

Improper Resource Access Authorization\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=331
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	1899	1899
Object	fprintf	fprintf

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method	write_viminfo_registers(FILE *fp)

```
....
1899.          fprintf(fp, "\"");
```

Improper Resource Access Authorization\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=332
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	1901	1901
Object	fprintf	fprintf

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method	write_viminfo_registers(FILE *fp)

```
....
1901.       fprintf(fp, "\"%c", c);
```

Improper Resource Access Authorization\Path 13:

	Severity	Low
	Result State	To Verify
	Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=333
	Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	1903	1903
Object	fprintf	fprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   write_viminfo_registers(FILE *fp)

```
....
1903.            fprintf(fp, "@");
```

Improper Resource Access Authorization\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=334
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	1904	1904
Object	fprintf	fprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   write_viminfo_registers(FILE *fp)

```
....
1904.        fprintf(fp, "\t%s\t%d\n", type, (int)y_ptr->y_width);
```

Improper Resource Access Authorization\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=335
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	1928	1928
Object	fprintf	fprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   write_viminfo_registers(FILE *fp)

```
....
1928.            fprintf(fp, "|%d,%d,%d,%d,%d,%d,%ld", BARTYPE_REGISTER,
flags,
```

Improper Resource Access Authorization\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=336
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	1959	1959
Object	fprintf	fprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   write_one_mark(FILE *fp_out, int c, pos_T *pos)

```
....
1959.        fprintf(fp_out, "\t%c\t%ld\t%d\n", c, (long)pos->lnum,
(int)pos->col);
```

Improper Resource Access Authorization\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=337
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c

Line	1969	1969
Object	fprintf	fprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   write_buffer_marks(buf_T *buf, FILE *fp_out)

```
....
1969.       fprintf(fp_out, "\n> ");
```

Improper Resource Access Authorization\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=338
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	2076	2076
Object	fprintf	fprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   write_one_filemark(

```
....
2076.       fprintf(fp, "%c%c  %ld  %ld  ", c1, c2, (long)fm->fmark.mark.lnum,
```

Improper Resource Access Authorization\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=339
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	2082	2082
Object	fprintf	fprintf

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c

Method	write_one_filemark(

```
....
2082.        fprintf(fp, "|%d,%d,%ld,%ld,%ld,", BARTYPE_MARK, c2,
```

Improper Resource Access Authorization\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=340
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	2967	2967
Object	fprintf	fprintf

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method	do_viminfo(FILE *fp_in, FILE *fp_out, int flags)

```
....
2967.        fprintf(fp_out, _("# This viminfo file was generated by Vim
%s.\n"),
```

Improper Resource Access Authorization\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=341
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	2972	2972
Object	fprintf	fprintf

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method	do_viminfo(FILE *fp_in, FILE *fp_out, int flags)

```
....
2972.        fprintf(fp_out, "*encoding=%s\n\n", p_enc);
```

Improper Resource Access Authorization\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=342
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	432	432
Object	fputs	fputs

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method	write_viminfo_bufferlist(FILE *fp)

```
....
432.        fputs(_("\n# Buffer list:\n"), fp);
```

Improper Resource Access Authorization\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=343
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	952	952
Object	fputs	fputs

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method	write_viminfo_barlines(vir_T *virp, FILE *fp_out)

```
....
952.        fputs(_("\n# Bar lines, copied verbatim:\n"), fp_out);
```

Improper Resource Access Authorization\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=344
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	960	960
Object	fputs	fputs

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   write_viminfo_barlines(vir_T *virp, FILE *fp_out)

```
....
960.              fputs(line, fp_out);
```

Improper Resource Access Authorization\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=345
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	1325	1325
Object	fputs	fputs

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   write_viminfo_varlist(FILE *fp)

```
....
1325.        fputs(_("\n# global variables:\n"), fp);
```

Improper Resource Access Authorization\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=346
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	1419	1419

Object	fputs	fputs

Code Snippet
File Name   vim@@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   write_viminfo_sub_string(FILE *fp)

```
....
1419.        fputs(_("\n# Last Substitute String:\n$"), fp);
```

Improper Resource Access Authorization\Path 27:

Severity   Low
Result State   To Verify
Online Results   http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=347
Status   New

	Source	Destination
File	vim@@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	1829	1829
Object	fputs	fputs

Code Snippet
File Name   vim@@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   write_viminfo_registers(FILE *fp)

```
....
1829.        fputs(_("\n# Registers:\n"), fp);
```

Improper Resource Access Authorization\Path 28:

Severity   Low
Result State   To Verify
Online Results   http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=348
Status   New

	Source	Destination
File	vim@@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	2026	2026
Object	fputs	fputs

Code Snippet
File Name   vim@@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   write_viminfo_marks(FILE *fp_out, garray_T *buflist)

```
....
2026.        fputs(_("\n# History of marks within files (newest to
oldest):\n"), fp_out);
```

Improper Resource Access Authorization\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=349
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	2107	2107
Object	fputs	fputs

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method	write_viminfo_filemarks(FILE *fp)

```
....
2107.        fputs(_("\n# File marks:\n"), fp);
```

Improper Resource Access Authorization\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=350
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	2178	2178
Object	fputs	fputs

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method	write_viminfo_filemarks(FILE *fp)

```
....
2178.        fputs(_("\n# Jumplist (newest first):\n"), fp);
```

Improper Resource Access Authorization\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=351
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	2387	2387
Object	fputs	fputs

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   copy_viminfo_marks(

```
....
2387.                  fputs("\n> ", fp_out);
```

Improper Resource Access Authorization\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=352
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	2390	2390
Object	fputs	fputs

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   copy_viminfo_marks(

```
....
2390.                  fputs((char *)line, fp_out);
```

Improper Resource Access Authorization\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=353
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	2440	2440
Object	fputs	fputs

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   copy_viminfo_marks(

```
....
2440.               fputs((char *)line, fp_out);
```

Improper Resource Access Authorization\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=354
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	2969	2969
Object	fputs	fputs

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   do_viminfo(FILE *fp_in, FILE *fp_out, int flags)

```
....
2969.        fputs(_("# You may edit it if you're careful!\n\n"),
fp_out);
```

Improper Resource Access Authorization\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=355
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	2971	2971

Object	fputs	fputs

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method	do_viminfo(FILE *fp_in, FILE *fp_out, int flags)

```
....
2971.        fputs(_("# Value of 'encoding' when this file was
written\n"), fp_out);
```

Improper Resource Access Authorization\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=356
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	893	893
Object	fputc	fputc

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method	write_viminfo_history(FILE *fp, int merge)

```
....
893.                    fputc(hist_type2char(type, TRUE), fp);
```

Potential Precision Problem
Query Path:
CPP\Cx\CPP Buffer Overflow\Potential Precision Problem Version:0

Categories

NIST SP 800-53: SI-10 Information Input Validation (P1)
OWASP Top 10 2017: A1-Injection

Description
Potential Precision Problem\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=51
Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: split", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	1764	1764
Object	"%.*s-%s: split"	"%.*s-%s: split"

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
1764.                       sprintf(changename[depth], "%.*s-%s: split",
```

Potential Precision Problem\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=52
Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: compound", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	1767	1767
Object	"%.*s-%s: compound"	"%.*s-%s: compound"

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
1767.                       sprintf(changename[depth], "%.*s-%s:
compound",
```

Potential Precision Problem\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=53
Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: subst %c to %c", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	1913	1913
Object	"%.*s-%s: subst %c to %c"	"%.*s-%s: subst %c to %c"

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

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

Potential Precision Problem\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=54
Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: accept %c", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	1917	1917
Object	"%.*s-%s: accept %c"	"%.*s-%s: accept %c"

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
1917.                    sprintf(changename[depth], "%.*s-%s: accept %c",
```

Potential Precision Problem\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=55

	Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: delete %c", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	2052	2052
Object	"%.*s-%s: delete %c"	"%.*s-%s: delete %c"

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
2052.              sprintf(changename[depth], "%.*s-%s: delete %c",
```

Potential Precision Problem\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=56
Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: insert %c", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	2149	2149
Object	"%.*s-%s: insert %c"	"%.*s-%s: insert %c"

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
2149.              sprintf(changename[depth], "%.*s-%s: insert %c",
```

Potential Precision Problem\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50

Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: swap %c and %c", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	2249	2249
Object	"%.*s-%s: swap %c and %c"	"%.*s-%s: swap %c and %c"

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
2249.                sprintf(changename[depth], "%.*s-%s: swap %c and %c",
```

Potential Precision Problem\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=58
Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: swap3 %c and %c", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	2337	2337
Object	"%.*s-%s: swap3 %c and %c"	"%.*s-%s: swap3 %c and %c"

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
2337.                sprintf(changename[depth], "%.*s-%s: swap3 %c and %c",
```

Potential Precision Problem\Path 9:

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

Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: rotate left %c%c%c", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	2405	2405
Object	"%.*s-%s: rotate left %c%c%c"	"%.*s-%s: rotate left %c%c%c"

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
2405.              sprintf(changename[depth], "%.*s-%s: rotate left
%c%c%c",
```

Potential Precision Problem\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=60
Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: rotate right %c%c%c", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	2466	2466
Object	"%.*s-%s: rotate right %c%c%c"	"%.*s-%s: rotate right %c%c%c"

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
2466.              sprintf(changename[depth], "%.*s-%s: rotate right
%c%c%c",
```

Potential Precision Problem\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=61
Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: replace %s with %s", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0408-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	2578	2578
Object	"%.*s-%s: replace %s with %s"	"%.*s-%s: replace %s with %s"

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   suggest_trie_walk(

```
....
2578.                    sprintf(changename[depth], "%.*s-%s: replace %s
with %s",
```

Potential Precision Problem\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=62
Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: split", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	1764	1764
Object	"%.*s-%s: split"	"%.*s-%s: split"

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
1764.                    sprintf(changename[depth], "%.*s-%s: split",
```

Potential Precision Problem\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=63
Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: compound", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	1767	1767
Object	"%.*s-%s: compound"	"%.*s-%s: compound"

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method	suggest_trie_walk(

```
....
1767.                       sprintf(changename[depth], "%.*s-%s:
compound",
```

Potential Precision Problem\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=64
Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: subst %c to %c", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	1913	1913
Object	"%.*s-%s: subst %c to %c"	"%.*s-%s: subst %c to %c"

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method	suggest_trie_walk(

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

Potential Precision Problem\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=65
Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: accept %c", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	1917	1917
Object	"%.*s-%s: accept %c"	"%.*s-%s: accept %c"

Code Snippet

File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
1917.                        sprintf(changename[depth], "%.*s-%s: accept %c",
```

Potential Precision Problem\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=66
Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: delete %c", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	2052	2052
Object	"%.*s-%s: delete %c"	"%.*s-%s: delete %c"

Code Snippet

File Name	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method	suggest_trie_walk(

```
....
2052.              sprintf(changename[depth], "%.*s-%s: delete %c",
```

Potential Precision Problem\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=67
Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: insert %c", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	2149	2149
Object	"%.*s-%s: insert %c"	"%.*s-%s: insert %c"

Code Snippet	
File Name	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method	suggest_trie_walk(

```
....
2149.              sprintf(changename[depth], "%.*s-%s: insert %c",
```

Potential Precision Problem\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=68
Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: swap %c and %c", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	2249	2249
Object	"%.*s-%s: swap %c and %c"	"%.*s-%s: swap %c and %c"

Code Snippet

File Name	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method	suggest_trie_walk(

```
....
2249.              sprintf(changename[depth], "%.*s-%s: swap %c and %c",
```

Potential Precision Problem\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=69
Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: swap3 %c and %c", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	2337	2337
Object	"%.*s-%s: swap3 %c and %c"	"%.*s-%s: swap3 %c and %c"

Code Snippet

File Name	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method	suggest_trie_walk(

```
....
2337.              sprintf(changename[depth], "%.*s-%s: swap3 %c and %c",
```

Potential Precision Problem\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=70
Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: rotate left %c%c%c", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	2405	2405
Object	"%.*s-%s: rotate left %c%c%c"	"%.*s-%s: rotate left %c%c%c"

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
2405.                sprintf(changename[depth], "%.*s-%s: rotate left
%c%c%c",
```

Potential Precision Problem\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=71
Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: rotate right %c%c%c", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	2466	2466
Object	"%.*s-%s: rotate right %c%c%c"	"%.*s-%s: rotate right %c%c%c"

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
2466.                sprintf(changename[depth], "%.*s-%s: rotate right
%c%c%c",
```

Potential Precision Problem\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=72
Status	New

The size of the buffer used by suggest_trie_walk in "%.*s-%s: replace %s with %s", at line 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.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 1237 of vim@@vim-v8.2.3451-CVE-2022-0943-TP.c, to overwrite the target buffer.

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c

Line	2578	2578
Object	"%.*s-%s: replace %s with %s"	"%.*s-%s: replace %s with %s"

Code Snippet
File Name   vim@@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   suggest_trie_walk(

```
....
2578.                    sprintf(changename[depth], "%.*s-%s: replace %s
with %s",
```

Arithmenic Operation On Boolean

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=1050077&projectid=50066&pathid=73
Status	New

	Source	Destination
File	vim@@@vim-v8.2.3018-CVE-2022-3520-TP.c	vim@@@vim-v8.2.3018-CVE-2022-3520-TP.c
Line	1764	1764
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name   vim@@@vim-v8.2.3018-CVE-2022-3520-TP.c
Method   do_put(

```
....
1764.                    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=1050077&projectid=50066&pathid=74
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Line	2143	2143
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Method   do_put(

```
....
2143.           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=1050077&projectid=50066&pathid=75
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Line	2149	2149
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Method   do_put(

```
....
2149.           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=1050077&projectid=50066&pathid=76
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c

Line	1764	1764
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name   vim@@@vim-v8.2.3018-CVE-2023-1170-TP.c
Method   do_put(

```
....
1764.              coladvance_force(getviscol() + (dir == FORWARD));
```

Arithmenic Operation On Boolean\Path 5:

	Source	Destination
File	vim@@@vim-v8.2.3018-CVE-2023-1170-TP.c	vim@@@vim-v8.2.3018-CVE-2023-1170-TP.c
Line	2143	2143
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name   vim@@@vim-v8.2.3018-CVE-2023-1170-TP.c
Method   do_put(

```
....
2143.              if (curbuf->b_op_start.lnum + (y_type == MCHAR) - 1 +
nr_lines
```

Arithmenic Operation On Boolean\Path 6:

	Source	Destination
File	vim@@@vim-v8.2.3018-CVE-2023-1170-TP.c	vim@@@vim-v8.2.3018-CVE-2023-1170-TP.c
Line	2149	2149
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name   vim@@@vim-v8.2.3018-CVE-2023-1170-TP.c

Method	do_put(

```
....
2149.               mark_adjust(curbuf->b_op_start.lnum + (y_type ==
MCHAR),
```

Arithmenic Operation On Boolean\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=79
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Line	1764	1764
Object	BinaryExpr	BinaryExpr

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Method	do_put(

```
....
1764.               coladvance_force(getviscol() + (dir == FORWARD));
```

Arithmenic Operation On Boolean\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=80
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Line	2143	2143
Object	BinaryExpr	BinaryExpr

Code Snippet	
File Name	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Method	do_put(

```
....
2143.               if (curbuf->b_op_start.lnum + (y_type == MCHAR) - 1 +
nr_lines
```

Arithmenic Operation On Boolean\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=81
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Line	2149	2149
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Method   do_put(

```
....
2149.              mark_adjust(curbuf->b_op_start.lnum + (y_type ==
MCHAR),
```

Arithmenic Operation On Boolean\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=82
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-2609-TP.c	vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Line	1764	1764
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Method   do_put(

```
....
1764.            coladvance_force(getviscol() + (dir == FORWARD));
```

Arithmenic Operation On Boolean\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50

Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-2609-TP.c	vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Line	2143	2143
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Method   do_put(

```
....
2143.            if (curbuf->b_op_start.lnum + (y_type == MCHAR) - 1 +
nr_lines
```

Arithmenic Operation On Boolean\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=84
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-2609-TP.c	vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Line	2149	2149
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Method   do_put(

```
....
2149.                mark_adjust(curbuf->b_op_start.lnum + (y_type ==
MCHAR),
```

Arithmenic Operation On Boolean\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=85
Status	New

	Source	Destination
	Source	Destination

File	vim@@vim-v8.2.3018-CVE-2023-2610-TP.c	vim@@vim-v8.2.3018-CVE-2023-2610-TP.c
Line	1340	1340
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-2610-TP.c
Method   reg_match_visual(void)

```
....
1340.          if (cols < start || cols > end - (*p_sel == 'e'))
```

Arithmenic Operation On Boolean\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=86
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4192-TP.c	vim@@vim-v8.2.3451-CVE-2021-4192-TP.c
Line	1337	1337
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4192-TP.c
Method   reg_match_visual(void)

```
....
1337.          if (cols < start || cols > end - (*p_sel == 'e'))
```

Arithmenic Operation On Boolean\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=87
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	552	552
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method	read_viminfo_history(vir_T *virp, int writing)

```
....
552.                  if (!in_history(type, val + (type == HIST_SEARCH),
```

Arithmenic Operation On Boolean\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=88
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0407-TP.c	vim@@vim-v8.2.3451-CVE-2022-0407-TP.c
Line	1774	1774
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name	vim@@vim-v8.2.3451-CVE-2022-0407-TP.c
Method	do_put(

```
....
1774.                  coladvance_force(getviscol() + (dir == FORWARD));
```

Arithmenic Operation On Boolean\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=89
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0407-TP.c	vim@@vim-v8.2.3451-CVE-2022-0407-TP.c
Line	2153	2153
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name	vim@@vim-v8.2.3451-CVE-2022-0407-TP.c
Method	do_put(

```
....
2153.              if (curbuf->b_op_start.lnum + (y_type == MCHAR) - 1 +
nr_lines
```

Arithmenic Operation On Boolean\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=90
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0407-TP.c	vim@@vim-v8.2.3451-CVE-2022-0407-TP.c
Line	2159	2159
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0407-TP.c
Method   do_put(

```
....
2159.              mark_adjust(curbuf->b_op_start.lnum + (y_type ==
MCHAR),
```

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=1050077&projectid=50066&pathid=754
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Line	1461	1461
Object	y_idx	y_idx

Code Snippet

File Name   vim@@vim-v8.2.3018-CVE-2022-3520-TP.c

Method   yank_copy_line(struct block_def *bd, long y_idx, int exclude_trailing_space)

```
....
1461.        y_current->y_array[y_idx] = pnew;
```

Unchecked Array Index\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=755
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c	vim@@vim-v8.2.3018-CVE-2022-3520-TP.c
Line	2648	2648
Object	len	len

Code Snippet

File Name   vim@@vim-v8.2.3018-CVE-2022-3520-TP.c

Method   get_reg_contents(int regname, int flags)

```
....
2648.         retval[len] = NUL;
```

Unchecked Array Index\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=756
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c
Line	1461	1461
Object	y_idx	y_idx

Code Snippet

File Name   vim@@vim-v8.2.3018-CVE-2023-1170-TP.c

Method   yank_copy_line(struct block_def *bd, long y_idx, int exclude_trailing_space)

```
....
1461.        y_current->y_array[y_idx] = pnew;
```

Unchecked Array Index\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=757
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c	vim@@vim-v8.2.3018-CVE-2023-1170-TP.c
Line	2648	2648
Object	len	len

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-1170-TP.c
Method   get_reg_contents(int regname, int flags)

```
....
2648.       retval[len] = NUL;
```

Unchecked Array Index\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=758
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Line	1461	1461
Object	y_idx	y_idx

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Method   yank_copy_line(struct block_def *bd, long y_idx, int exclude_trailing_space)

```
....
1461.       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=1050077&projectid=50066&pathid=759

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c	vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Line	2648	2648
Object	len	len

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-1175-TP.c
Method   get_reg_contents(int regname, int flags)

```
....
2648.        retval[len] = NUL;
```

Unchecked Array Index\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=760
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-2609-TP.c	vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Line	1461	1461
Object	y_idx	y_idx

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-2609-TP.c
Method   yank_copy_line(struct block_def *bd, long y_idx, int exclude_trailing_space)

```
....
1461.        y_current->y_array[y_idx] = pnew;
```

Unchecked Array Index\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=761
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-2609-TP.c	vim@@vim-v8.2.3018-CVE-2023-2609-TP.c

Line	2648	2648
Object	len	len

Code Snippet

File Name   vim@@@vim-v8.2.3018-CVE-2023-2609-TP.c
Method   get_reg_contents(int regname, int flags)

```
....
2648.         retval[len] = NUL;
```

Unchecked Array Index\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=762
Status	New

	Source	Destination
File	vim@@@vim-v8.2.3018-CVE-2023-2610-TP.c	vim@@@vim-v8.2.3018-CVE-2023-2610-TP.c
Line	2438	2438
Object	len	len

Code Snippet

File Name   vim@@@vim-v8.2.3018-CVE-2023-2610-TP.c
Method   reg_submatch(int no)

```
....
2438.                 retval[len] = '\n';
```

Unchecked Array Index\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=763
Status	New

	Source	Destination
File	vim@@@vim-v8.2.3018-CVE-2023-2610-TP.c	vim@@@vim-v8.2.3018-CVE-2023-2610-TP.c
Line	2449	2449
Object	len	len

Code Snippet

File Name   vim@@@vim-v8.2.3018-CVE-2023-2610-TP.c

Method	reg_submatch(int no)

```
....
2449.                    retval[len] = '\n';
```

Unchecked Array Index\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=764
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-2610-TP.c	vim@@vim-v8.2.3018-CVE-2023-2610-TP.c
Line	2457	2457
Object	len	len

Code Snippet

File Name	vim@@vim-v8.2.3018-CVE-2023-2610-TP.c
Method	reg_submatch(int no)

```
....
2457.                    retval[len] = NUL;
```

Unchecked Array Index\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=765
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4192-TP.c	vim@@vim-v8.2.3451-CVE-2021-4192-TP.c
Line	2435	2435
Object	len	len

Code Snippet

File Name	vim@@vim-v8.2.3451-CVE-2021-4192-TP.c
Method	reg_submatch(int no)

```
....
2435.                    retval[len] = '\n';
```

Unchecked Array Index\Path 13:

	Severity	Low
	Result State	To Verify
	Online Results	
	Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4192-TP.c	vim@@vim-v8.2.3451-CVE-2021-4192-TP.c
Line	2446	2446
Object	len	len

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4192-TP.c
Method   reg_submatch(int no)

```
....
2446.                     retval[len] = '\n';
```

Unchecked Array Index\Path 14:

	Severity	Low
	Result State	To Verify
	Online Results	
	Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4192-TP.c	vim@@vim-v8.2.3451-CVE-2021-4192-TP.c
Line	2454	2454
Object	len	len

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4192-TP.c
Method   reg_submatch(int no)

```
....
2454.                     retval[len] = NUL;
```

Unchecked Array Index\Path 15:

	Severity	Low
	Result State	To Verify
	Online Results	
	Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	565	565
Object	len	len

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   read_viminfo_history(vir_T *virp, int writing)

```
....
565.                    p[len] = sep;
```

Unchecked Array Index\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=769
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c	vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Line	1103	1103
Object	len	len

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0156-TP.c
Method   barline_parse(vir_T *virp, char_u *text, garray_T *values)

```
....
1103.            s[len] = NUL;
```

Unchecked Array Index\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=770
Status	New

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2022-0407-TP.c	vim@@vim-v8.2.3451-CVE-2022-0407-TP.c
Line	1470	1470

Object	y_idx	y_idx

Code Snippet
File Name   vim@@@vim-v8.2.3451-CVE-2022-0407-TP.c
Method   yank_copy_line(struct block_def *bd, long y_idx, int exclude_trailing_space)

```
....
1470.        y_current->y_array[y_idx] = pnew;
```

Unchecked Array Index\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=771
Status	New

	Source	Destination
File	vim@@@vim-v8.2.3451-CVE-2022-0407-TP.c	vim@@@vim-v8.2.3451-CVE-2022-0407-TP.c
Line	2658	2658
Object	len	len

Code Snippet
File Name   vim@@@vim-v8.2.3451-CVE-2022-0407-TP.c
Method   get_reg_contents(int regname, int flags)

```
....
2658.        retval[len] = NUL;
```

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=1050077&projectid=50066&pathid=32
Status	New

The buffer allocated by <= in vim@@@vim-v8.2.3018-CVE-2022-3278-TP.c at line 3700 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-v8.2.3018-CVE-2022-3278-TP.c	vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Line	4282	4282
Object	<=	<=

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2022-3278-TP.c
Method   define_function(exarg_T *eap, char_u *name_arg)

```
....
4282.              for (i = 0; i <= cstack->cs_idx; ++i)
```

Potential Off by One Error in Loops\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=33
Status	New

The buffer allocated by <= in vim@@vim-v8.2.3451-CVE-2021-4173-TP.c at line 622 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-v8.2.3451-CVE-2021-4173-TP.c	vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Line	640	640
Object	<=	<=

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4173-TP.c
Method   function_using_block_scopes(ufunc_T *fp, cstack_T *cstack)

```
....
640.        for (i = 0; i <= cstack->cs_idx; ++i)
```

Potential Off by One Error in Loops\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=34
Status	New

The buffer allocated by <= in vim@@vim-v8.2.3451-CVE-2021-4187-TP.c at line 622 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-v8.2.3451-CVE-2021-4187-TP.c	vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Line	640	640
Object	<=	<=

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4187-TP.c
Method   function_using_block_scopes(ufunc_T *fp, cstack_T *cstack)

```
....
640.          for (i = 0; i <= cstack->cs_idx; ++i)
```

Potential Off by One Error in Loops\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=35
Status	New

The buffer allocated by <= in vim@@vim-v8.2.3451-CVE-2022-0408-TP.c at line 4296 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-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	4383	4383
Object	<=	<=

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   spell_edit_score_limit_w(

```
....
4383.              for (round = 0; round <= 1; ++round)
```

Potential Off by One Error in Loops\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=36
Status	New

The buffer allocated by <= in vim@@vim-v8.2.3451-CVE-2022-0408-TP.c at line 4121 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-v8.2.3451-CVE-2022-0408-TP.c	vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Line	4198	4198
Object	<=	<=

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0408-TP.c
Method   spell_edit_score_limit(

```
....
4198.            for (round = 0; round <= 1; ++round)
```

Potential Off by One Error in Loops\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=37
Status	New

The buffer allocated by <= in vim@@vim-v8.2.3451-CVE-2022-0572-TP.c at line 289 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-v8.2.3451-CVE-2022-0572-TP.c	vim@@vim-v8.2.3451-CVE-2022-0572-TP.c
Line	298	298
Object	<=	<=

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0572-TP.c
Method   tabstop_copy(int *oldts)

```
....
298.            for (t = 0; t <= oldts[0]; ++t)
```

Potential Off by One Error in Loops\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=38
Status	New

The buffer allocated by <= in vim@@vim-v8.2.3451-CVE-2022-0714-TP.c at line 289 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-v8.2.3451-CVE-2022-0714-TP.c	vim@@vim-v8.2.3451-CVE-2022-0714-TP.c
Line	298	298
Object	<=	<=

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0714-TP.c
Method   tabstop_copy(int *oldts)

```
....
298.          for (t = 0; t <= oldts[0]; ++t)
```

Potential Off by One Error in Loops\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=39
Status	New

The buffer allocated by <= in vim@@vim-v8.2.3451-CVE-2022-0943-TP.c at line 4296 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-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	4383	4383
Object	<=	<=

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   spell_edit_score_limit_w(

```
....
4383.             for (round = 0; round <= 1; ++round)
```

Potential Off by One Error in Loops\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=40
Status	New

The buffer allocated by <= in vim@@vim-v8.2.3451-CVE-2022-0943-TP.c at line 4121 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-v8.2.3451-CVE-2022-0943-TP.c	vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Line	4198	4198
Object	<=	<=

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2022-0943-TP.c
Method   spell_edit_score_limit(

```
....
4198.              for (round = 0; round <= 1; ++round)
```

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=1050077&projectid=50066&pathid=357
Status	New

	Source	Destination
File	vim@@vim-v8.2.3018-CVE-2023-2610-TP.c	vim@@vim-v8.2.3018-CVE-2023-2610-TP.c
Line	2678	2678
Object	f	f

Code Snippet
File Name   vim@@vim-v8.2.3018-CVE-2023-2610-TP.c
Method   vim_regcomp(char_u *expr_arg, int re_flags)

```
....
2678.              f = fopen(BT_REGEXP_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=1050077&projectid=50

Status	New	

	Source	Destination
File	vim@@vim-v8.2.3451-CVE-2021-4192-TP.c	vim@@vim-v8.2.3451-CVE-2021-4192-TP.c
Line	2675	2675
Object	f	f

Code Snippet

File Name   vim@@vim-v8.2.3451-CVE-2021-4192-TP.c
Method   vim_regcomp(char_u *expr_arg, int re_flags)

```
....
2675.              f = fopen(BT_REGEXP_DEBUG_LOG_NAME, "a");
```

TOCTOU

Description

TOCTOU\Path 1:

Severity	Low	
Result State	To Verify	
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=772	
Status	New	

The vim_regcomp method in vim@@vim-v8.2.3018-CVE-2023-2610-TP.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-v8.2.3018-CVE-2023-2610-TP.c	vim@@vim-v8.2.3018-CVE-2023-2610-TP.c
Line	2678	2678
Object	fopen	fopen

Code Snippet

File Name   vim@@vim-v8.2.3018-CVE-2023-2610-TP.c
Method   vim_regcomp(char_u *expr_arg, int re_flags)

```
....
2678.              f = fopen(BT_REGEXP_DEBUG_LOG_NAME, "a");
```

TOCTOU\Path 2:

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

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050077&projectid=50066&pathid=773	
Status	New	

The vim_regcomp method in vim@@vim-v8.2.3451-CVE-2021-4192-TP.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-v8.2.3451-CVE-2021-4192-TP.c	vim@@vim-v8.2.3451-CVE-2021-4192-TP.c
Line	2675	2675
Object	fopen	fopen

Code Snippet
File Name   vim@@vim-v8.2.3451-CVE-2021-4192-TP.c
Method   vim_regcomp(char_u *expr_arg, int re_flags)

```
....
2675.                 f = fopen(BT_REGEXP_DEBUG_LOG_NAME, "a");
```

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
//..
```

Char Overflow

Risk

What might happen

Assigning large data types into smaller data types, without proper checks and explicit casting, will lead to undefined behavior and unintentional effects, such as data corruption (e.g. value wraparound, wherein maximum values become minimum values); system crashes; infinite loops; logic errors, such as bypassing of security mechanisms; or even buffer overflows leading to arbitrary code execution.

Cause

How does it happen

This flaw can occur when implicitly casting numerical data types of a larger size, into a variable with a data type of a smaller size. This forces the program to discard some bits of information from the number. Depending on how the numerical data types are stored in memory, this is often the bits with the highest value, causing substantial corruption of the stored number. Alternatively, the sign bit of a signed integer could be lost, completely reversing the intention of the number.

General Recommendations

How to avoid it

- o Avoid casting larger data types to smaller types.
- o Prefer promoting the target variable to a large enough data type.
- o 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

```cpp
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

```cpp
int safe_addition(short op1, int op2) {

    // total variable is of type int, the largest type that is needed
    int total = 0;

    // check if total will overflow available integer size
    if (INT_MAX - abs(op2) > op1)
```

```
    {
        total = op1 + op2;
    }
    else
    {
        // instead of overflow, saturate (but this is not always a good thing)
        total = INT_MAX
    }

    return total;
}
```

Integer Overflow

Risk

What might happen

Assigning large data types into smaller data types, without proper checks and explicit casting, will lead to undefined behavior and unintentional effects, such as data corruption (e.g. value wraparound, wherein maximum values become minimum values); system crashes; infinite loops; logic errors, such as bypassing of security mechanisms; or even buffer overflows leading to arbitrary code execution.

Cause

How does it happen

This flaw can occur when implicitly casting numerical data types of a larger size, into a variable with a data type of a smaller size. This forces the program to discard some bits of information from the number. Depending on how the numerical data types are stored in memory, this is often the bits with the highest value, causing substantial corruption of the stored number. Alternatively, the sign bit of a signed integer could be lost, completely reversing the intention of the number.

General Recommendations

How to avoid it

- o Avoid casting larger data types to smaller types.
- o Prefer promoting the target variable to a large enough data type.
- o If downcasting is necessary, always check that values are valid and in range of the target type, before casting

Source Code Examples

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 occuring.

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

```java
public float getAverage(HttpServletRequest req) {
    int total = Integer.parseInt(req.getParameter("total"));
    int count = Integer.parseInt(req.getParameter("count"));

    return total / count;
}
```

Checked Division

```java
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;
}
```

PAGE 229 OF 275

MemoryFree on StackVariable

Risk

What might happen

Undefined Behavior may result with a crash. Crashes may give an attacker valuable information about the system and the program internals. Furthermore, it may leave unprotected files (e.g memory) that may be exploited.

Cause

How does it happen

Calling free() on a variable that was not dynamically allocated (e.g. malloc) will result with an Undefined Behavior.

General Recommendations

How to avoid it

Use free() only on dynamically allocated variables in order to prevent unexpected behavior from the compiler.

Source Code Examples

CPP

Bad - Calling free() on a static variable

```cpp
void clean_up(){
  char temp[256];
  do_something();
  free(tmp);
  return;
}
```

Good - Calling free() only on variables that were dynamically allocated

```cpp
void clean_up(){
  char *buff;
  buff = (char*) malloc(1024);
  free(buff);
  return;
}
```

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()

```cpp
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

```c
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

```c
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

```c
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

```c
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;
}
```

Weakness ID: 401 *(Weakness Base)*　　　　　　　　　　　　　　　　　　　　　　**Status:** Draft

Description

Description Summary

The software does not sufficiently track and release allocated memory after it has been used, which slowly consumes remaining memory.

Extended Description

This is often triggered by improper handling of malformed data or unexpectedly interrupted sessions.

Terminology Notes

"memory leak" has sometimes been used to describe other kinds of issues, e.g. for information leaks in which the contents of memory are inadvertently leaked (CVE-2003-0400 is one such example of this terminology conflict).

- -

Time of Introduction

- Architecture and Design
- Implementation

Applicable Platforms

Languages

C

C++

Modes of Introduction

Memory leaks have two common and sometimes overlapping causes:

- Error conditions and other exceptional circumstances
- Confusion over which part of the program is responsible for freeing the memory

- -

Common Consequences

Scope	Effect
Availability	Most memory leaks result in general software reliability problems, but if an attacker can intentionally trigger a memory leak, the attacker might be able to launch a denial of service attack (by crashing or hanging the program) or take advantage of other unexpected program behavior resulting from a low memory condition.

Likelihood of Exploit

Medium

Demonstrative Examples

Example 1

The following C function leaks a block of allocated memory if the call to read() fails to return the expected number of bytes:

(Bad Code)

Example Language: **C**

```
char* getBlock(int fd) {
char* buf = (char*) malloc(BLOCK_SIZE);
if (!buf) {
return NULL;
}
if (read(fd, buf, BLOCK_SIZE) != BLOCK_SIZE) {

return NULL;
}
```

```
return buf;
}
```

Example 2

Here the problem is that every time a connection is made, more memory is allocated. So if one just opened up more and more connections, eventually the machine would run out of memory.

(Bad Code)

Example Language: **C**

```
bar connection(){
foo = malloc(1024);
return foo;
}
endConnection(bar foo) {

free(foo);
}
int main() {

while(1) //thread 1
//On a connection
foo=connection(); //thread 2
//When the connection ends
endConnection(foo)
}
```

Observed Examples

Reference	Description
CVE-2005-3119	Memory leak because function does not free() an element of a data structure.
CVE-2004-0427	Memory leak when counter variable is not decremented.
CVE-2002-0574	Memory leak when counter variable is not decremented.
CVE-2005-3181	Kernel uses wrong function to release a data structure, preventing data from being properly tracked by other code.
CVE-2004-0222	Memory leak via unknown manipulations as part of protocol test suite.
CVE-2001-0136	Memory leak via a series of the same command.

Potential Mitigations

Pre-design: Use a language or compiler that performs automatic bounds checking.

Phase: Architecture and Design

Use an abstraction library to abstract away risky APIs. Not a complete solution.

Pre-design through Build: The Boehm-Demers-Weiser Garbage Collector or valgrind can be used to detect leaks in code. This is not a complete solution as it is not 100% effective.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	398	Indicator of Poor Code Quality	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Category	399	Resource Management Errors	**Development Concepts (primary)699**
ChildOf	Category	633	Weaknesses that Affect Memory	**Resource-specific Weaknesses (primary)631**
ChildOf	Category	730	OWASP Top Ten 2004 Category A9 - Denial of Service	**Weaknesses in OWASP Top Ten (2004) (primary)711**
ChildOf	Weakness Base	772	Missing Release of Resource after Effective	**Research Concepts (primary)1000**

			Lifetime	
MemberOf	View	630	[Weaknesses Examined by SAMATE](#)	**Weaknesses Examined by SAMATE (primary)630**
CanFollow	Weakness Class	390	[Detection of Error Condition Without Action](#)	Research Concepts1000

Relationship Notes

This is often a resultant weakness due to improper handling of malformed data or early termination of sessions.

Affected Resources

‣ Memory

Functional Areas

‣ Memory management

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
PLOVER			Memory leak
7 Pernicious Kingdoms			Memory Leak
CLASP			Failure to deallocate data
OWASP Top Ten 2004	A9	CWE More Specific	Denial of Service

White Box Definitions

A weakness where the code path has:

1. start statement that allocates dynamically allocated memory resource

2. end statement that loses identity of the dynamically allocated memory resource creating situation where dynamically allocated memory resource is never relinquished

Where "loses" is defined through the following scenarios:

1. identity of the dynamic allocated memory resource never obtained

2. the statement assigns another value to the data element that stored the identity of the dynamically allocated memory resource and there are no aliases of that data element

3. identity of the dynamic allocated memory resource obtained but never passed on to function for memory resource release

4. the data element that stored the identity of the dynamically allocated resource has reached the end of its scope at the statement and there are no aliases of that data element

References

J. Whittaker and H. Thompson. "How to Break Software Security". Addison Wesley. 2003.

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	PLOVER		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Eric Dalci	Cigital	External
updated Time of Introduction			
2008-08-01		KDM Analytics	External
added/updated white box definitions			
2008-08-15		Veracode	External
Suggested OWASP Top Ten 2004 mapping			
2008-09-08	CWE Content Team	MITRE	Internal
updated Applicable Platforms, Common Consequences, Relationships, Other Notes, References, Relationship Notes, Taxonomy Mappings, Terminology Notes			
2008-10-14	CWE Content Team	MITRE	Internal
updated Description			
2009-03-10	CWE Content Team	MITRE	Internal
updated Other Notes			
2009-05-27	CWE Content Team	MITRE	Internal
updated Name			
2009-07-17	KDM Analytics		External
Improved the White Box Definition			

2009-07-27	CWE Content Team	MITRE	Internal
updated White Box Definitions			
2009-10-29	CWE Content Team	MITRE	Internal
updated Modes of Introduction, Other Notes			
2010-02-16	CWE Content Team	MITRE	Internal
updated Relationships			

Previous Entry Names

Change Date	Previous Entry Name
2008-04-11	Memory Leak
2009-05-27	Failure to Release Memory Before Removing Last Reference (aka 'Memory Leak')

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

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**

```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 = -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

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

```cpp
char * input = NULL;
printf("%s", input);
```

Implicit NULL Dereference

```cpp
char * input;
printf("%s", input);
```

Java
Explicit Null Dereference

```java
Object o = null;
out.println(o.getClass());
```

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

```cpp
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

```c
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

```c
strncat(buf, input, sizeof(buf) - strlen(buf)); // actual value should be sizeof(buf)-
strlen(buf)-1 - this form will overwrite the terminating nullbyte
```

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 it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.

Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

CPP

Overflowing Buffers

```cpp
const int BUFFER_SIZE = 10;
char buffer[BUFFER_SIZE];

void copyStringToBuffer(char* inputString)

{

    strcpy(buffer, inputString);
}
```

Checked Buffers

```cpp
const int BUFFER_SIZE = 10;
const int MAX_INPUT_SIZE = 256;
char buffer[BUFFER_SIZE];

void copyStringToBuffer(char* inputString)
```

```
{
    if (strnlen(inputString, MAX_INPUT_SIZE) < sizeof(buffer))
    {
        strncpy(buffer, inputString, sizeof(buffer));
    }
}
```

Indicator of Poor Code Quality

Weakness ID: 398 *(Weakness Class)*   **Status:** Draft

Description

Description Summary

The code has features that do not directly introduce a weakness or vulnerability, but indicate that the product has not been carefully developed or maintained.

Extended Description

Programs are more likely to be secure when good development practices are followed. If a program is complex, difficult to maintain, not portable, or shows evidence of neglect, then there is a higher likelihood that weaknesses are buried in the code.

Time of Introduction

- Architecture and Design
- Implementation

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	18	Source Code	**Development Concepts (primary)699**
ChildOf	Weakness Class	710	Coding Standards Violation	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	107	Struts: Unused Validation Form	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	110	Struts: Validator Without Form Field	**Research Concepts (primary)1000**
ParentOf	Category	399	Resource Management Errors	**Development Concepts (primary)699**
ParentOf	Weakness Base	401	Failure to Release Memory Before Removing Last Reference ('Memory Leak')	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	404	Improper Resource Shutdown or Release	Development Concepts699 **Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Variant	415	Double Free	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	416	Use After Free	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Variant	457	Use of Uninitialized Variable	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	474	Use of Function with Inconsistent Implementations	**Development Concepts (primary)699 Seven Pernicious Kingdoms (primary)700 Research Concepts (primary)1000**
ParentOf	Weakness Base	475	Undefined Behavior for Input to API	**Development Concepts (primary)699 Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	476	NULL Pointer Dereference	**Development Concepts**

				(primary)699 Seven Pernicious Kingdoms (primary)700 Research Concepts (primary)1000
ParentOf	Weakness Base	477	Use of Obsolete Functions	**Development Concepts (primary)699 Seven Pernicious Kingdoms (primary)700 Research Concepts (primary)1000**
ParentOf	Weakness Variant	478	Missing Default Case in Switch Statement	**Development Concepts (primary)699**
ParentOf	Weakness Variant	479	Unsafe Function Call from a Signal Handler	**Development Concepts (primary)699**
ParentOf	Weakness Variant	483	Incorrect Block Delimitation	**Development Concepts (primary)699**
ParentOf	Weakness Base	484	Omitted Break Statement in Switch	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Variant	546	Suspicious Comment	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Weakness Variant	547	Use of Hard-coded, Security-relevant Constants	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Weakness Variant	561	Dead Code	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Weakness Base	562	Return of Stack Variable Address	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Variant	563	Unused Variable	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Category	569	Expression Issues	**Development Concepts (primary)699**
ParentOf	Weakness Variant	585	Empty Synchronized Block	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Weakness Variant	586	Explicit Call to Finalize()	**Development Concepts (primary)699**
ParentOf	Weakness Variant	617	Reachable Assertion	**Development Concepts (primary)699**
ParentOf	Weakness Base	676	Use of Potentially Dangerous Function	**Development Concepts (primary)699 Research Concepts (primary)1000**
MemberOf	View	700	Seven Pernicious Kingdoms	**Seven Pernicious Kingdoms (primary)700**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name

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		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Description, Relationships, Taxonomy Mappings		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Relationships		

Previous Entry Names	
Change Date	**Previous Entry Name**
2008-04-11	Code Quality

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**

```perl
sub DisplayPrivateMessage {
my($id) = @_;
my $Message = LookupMessageObject($id);
print "From: " . encodeHTML($Message->{from}) . "<br>\n";
print "Subject: " . encodeHTML($Message->{subject}) . "\n";
print "<hr>\n";
print "Body: " . encodeHTML($Message->{body}) . "\n";
}

my $q = new CGI;
# For purposes of this example, assume that CWE-309 and
# CWE-523 do not apply.
if (! AuthenticateUser($q->param('username'), $q->param('password'))) {
ExitError("invalid username or password");
}

my $id = $q->param('id');
DisplayPrivateMessage($id);
```

While the program properly exits if authentication fails, it does not ensure that the message is addressed to the user. As a result, an authenticated attacker could provide any arbitrary identifier and read private messages that were intended for other users.

One way to avoid this problem would be to ensure that the "to" field in the message object matches the username of the authenticated user.

Observed Examples

Reference	Description
CVE-2009-3168	Web application does not restrict access to admin scripts, allowing authenticated users to reset administrative passwords.

CVE-2009-2960	Web application does not restrict access to admin scripts, allowing authenticated users to modify passwords of other users.
CVE-2009-3597	Web application stores database file under the web root with insufficient access control (CWE-219), allowing direct request.
CVE-2009-2282	Terminal server does not check authorization for guest access.
CVE-2009-3230	Database server does not use appropriate privileges for certain sensitive operations.
CVE-2009-2213	Gateway uses default "Allow" configuration for its authorization settings.
CVE-2009-0034	Chain: product does not properly interpret a configuration option for a system group, allowing users to gain privileges.
CVE-2008-6123	Chain: SNMP product does not properly parse a configuration option for which hosts are allowed to connect, allowing unauthorized IP addresses to connect.
CVE-2008-5027	System monitoring software allows users to bypass authorization by creating custom forms.
CVE-2008-7109	Chain: reliance on client-side security (CWE-602) allows attackers to bypass authorization using a custom client.
CVE-2008-3424	Chain: product does not properly handle wildcards in an authorization policy list, allowing unintended access.
CVE-2009-3781	Content management system does not check access permissions for private files, allowing others to view those files.
CVE-2008-4577	ACL-based protection mechanism treats negative access rights as if they are positive, allowing bypass of intended restrictions.
CVE-2008-6548	Product does not check the ACL of a page accessed using an "include" directive, allowing attackers to read unauthorized files.
CVE-2007-2925	Default ACL list for a DNS server does not set certain ACLs, allowing unauthorized DNS queries.
CVE-2006-6679	Product relies on the X-Forwarded-For HTTP header for authorization, allowing unintended access by spoofing the header.
CVE-2005-3623	OS kernel does not check for a certain privilege before setting ACLs for files.
CVE-2005-2801	Chain: file-system code performs an incorrect comparison (CWE-697), preventing defauls ACLs from being properly applied.
CVE-2001-1155	Chain: product does not properly check the result of a reverse DNS lookup because of operator precedence (CWE-783), allowing bypass of DNS-based access restrictions.

Potential Mitigations

Phase: Architecture and Design

Divide your application into anonymous, normal, privileged, and administrative areas. Reduce the attack surface by carefully mapping roles with data and functionality. Use role-based access control (RBAC) to enforce the roles at the appropriate boundaries.

Note that this approach may not protect against horizontal authorization, i.e., it will not protect a user from attacking others with the same role.

Phase: Architecture and Design

Ensure that you perform access control checks related to your business logic. These checks may be different than the access control checks that you apply to more generic resources such as files, connections, processes, memory, and database records. For example, a database may restrict access for medical records to a specific database user, but each record might only be intended to be accessible to the patient and the patient's doctor.

Phase: Architecture and Design

Strategy: Libraries or Frameworks

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness

----

easier to avoid.

For example, consider using authorization frameworks such as the JAAS Authorization Framework and the OWASP ESAPI Access Control feature.

- -

Phase: Architecture and Design

For web applications, make sure that the access control mechanism is enforced correctly at the server side on every page. Users should not be able to access any unauthorized functionality or information by simply requesting direct access to that page.

One way to do this is to ensure that all pages containing sensitive information are not cached, and that all such pages restrict access to requests that are accompanied by an active and authenticated session token associated with a user who has the required permissions to access that page.

- -

Phases: System Configuration; Installation

Use the access control capabilities of your operating system and server environment and define your access control lists accordingly. Use a "default deny" policy when defining these ACLs.

- -

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	254	Security Features	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Weakness Class	284	Access Control (Authorization) Issues	**Development Concepts (primary)699 Research Concepts (primary)1000**
ChildOf	Category	721	OWASP Top Ten 2007 Category A10 - Failure to Restrict URL Access	**Weaknesses in OWASP Top Ten (2007) (primary)629**
ChildOf	Category	723	OWASP Top Ten 2004 Category A2 - Broken Access Control	**Weaknesses in OWASP Top Ten (2004) (primary)711**
ChildOf	Category	753	2009 Top 25 - Porous Defenses	**Weaknesses in the 2009 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)750**
ChildOf	Category	803	2010 Top 25 - Porous Defenses	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
ParentOf	Weakness Variant	219	Sensitive Data Under Web Root	**Research Concepts (primary)1000**
ParentOf	Weakness Base	551	Incorrect Behavior Order: Authorization Before Parsing and Canonicalization	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Class	638	Failure to Use Complete Mediation	Research Concepts1000
ParentOf	Weakness Base	804	Guessable CAPTCHA	**Development Concepts (primary)699 Research Concepts (primary)1000**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
7 Pernicious Kingdoms			Missing Access Control
OWASP Top Ten 2007	A10	CWE More Specific	Failure to Restrict URL Access
OWASP Top Ten 2004	A2	CWE More Specific	Broken Access Control

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
1	Accessing Functionality Not Properly Constrained by ACLs	
13	Subverting Environment Variable Values	

17	Accessing, Modifying or Executing Executable Files
87	Forceful Browsing
39	Manipulating Opaque Client-based Data Tokens
45	Buffer Overflow via Symbolic Links
51	Poison Web Service Registry
59	Session Credential Falsification through Prediction
60	Reusing Session IDs (aka Session Replay)
77	Manipulating User-Controlled Variables
76	Manipulating Input to File System Calls
104	Cross Zone Scripting

References

NIST. "Role Based Access Control and Role Based Security". <http://csrc.nist.gov/groups/SNS/rbac/>.

- -

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 4, "Authorization" Page 114; Chapter 6, "Determining Appropriate Access Control" Page 171. 2nd Edition. Microsoft. 2002.

- -

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	7 Pernicious Kingdoms		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Eric Dalci	Cigital	External
updated Time of Introduction			
2008-08-15		Veracode	External
Suggested OWASP Top Ten 2004 mapping			
2008-09-08	CWE Content Team	MITRE	Internal
updated Relationships, Other Notes, Taxonomy Mappings			
2009-01-12	CWE Content Team	MITRE	Internal
updated Common Consequences, Description, Likelihood of Exploit, Name, Other Notes, Potential Mitigations, References, Relationships			
2009-03-10	CWE Content Team	MITRE	Internal
updated Potential Mitigations			
2009-05-27	CWE Content Team	MITRE	Internal
updated Description, Related Attack Patterns			
2009-07-27	CWE Content Team	MITRE	Internal
updated Relationships			
2009-10-29	CWE Content Team	MITRE	Internal
updated Type			
2009-12-28	CWE Content Team	MITRE	Internal
updated Applicable Platforms, Common Consequences, Demonstrative Examples, Detection Factors, Modes of Introduction, Observed Examples, Relationships			
2010-02-16	CWE Content Team	MITRE	Internal
updated Alternate Terms, Detection Factors, Potential Mitigations, References, Relationships			
2010-04-05	CWE Content Team	MITRE	Internal
updated Potential Mitigations			
Previous Entry Names			
Change Date	**Previous Entry Name**		
2009-01-12	Missing or Inconsistent Access Control		

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

Unchecked Return Value

Risk

What might happen

A program that does not check function return values could cause the application to enter an undefined state. This could lead to unexpected behavior and unintended consequences, including inconsistent data, system crashes or other error-based exploits.

Cause

How does it happen

The application calls a system function, but does not receive or check the result of this funciton. These functions often return error codes in the result, or share other status codes with it's caller. The application simply ignores this result value, losing this vital information.

General Recommendations

How to avoid it

 - Always check the result of any called function that returns a value, and verify the result is an expected value.

 - Ensure the calling function responds to all possible return values.

 - Expect runtime errors and handle them gracefully. Explicitly define a mechanism for handling unexpected errors.

Source Code Examples

CPP

Unchecked Memory Allocation

```cpp
buff = (char*) malloc(size);
strncpy(buff, source, size);
```

Safer Memory Allocation

```cpp
buff = (char*) malloc(size+1);
if (buff==NULL) exit(1);

strncpy(buff, source, size);
buff[size] = '\0';
```

Use of sizeof() on a Pointer Type

Weakness ID: 467 *(Weakness Variant)*   **Status:** Draft

Description

Description Summary

The code calls sizeof() on a malloced pointer type, which always returns the wordsize/8. This can produce an unexpected result if the programmer intended to determine how much memory has been allocated.

Time of Introduction

- Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

Scope	Effect
Integrity	This error can often cause one to allocate a buffer that is much smaller than what is needed, leading to resultant weaknesses such as buffer overflows.

Likelihood of Exploit

High

Demonstrative Examples

Example 1

Care should be taken to ensure sizeof returns the size of the data structure itself, and not the size of the pointer to the data structure.

In this example, sizeof(foo) returns the size of the pointer.

(Bad Code)
Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(foo));
```

In this example, sizeof(*foo) returns the size of the data structure and not the size of the pointer.

(Good Code)
Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(*foo));
```

Example 2

This example defines a fixed username and password. The AuthenticateUser() function is intended to accept a username and a password from an untrusted user, and check to ensure that it matches the username and password. If the username and password match, AuthenticateUser() is intended to indicate that authentication succeeded.

(Bad Code)

```
/* Ignore CWE-259 (hard-coded password) and CWE-309 (use of password system for authentication) for this example. */

char *username = "admin";
char *pass = "password";

int AuthenticateUser(char *inUser, char *inPass) {
```

```
printf("Sizeof username = %d\n", sizeof(username));
printf("Sizeof pass = %d\n", sizeof(pass));

if (strncmp(username, inUser, sizeof(username))) {
printf("Auth failure of username using sizeof\n");
return(AUTH_FAIL);
}
/* Because of CWE-467, the sizeof returns 4 on many platforms and architectures. */
if (! strncmp(pass, inPass, sizeof(pass))) {
printf("Auth success of password using sizeof\n");
return(AUTH_SUCCESS);
}
else {
printf("Auth fail of password using sizeof\n");
return(AUTH_FAIL);
}
}

int main (int argc, char **argv)
{
int authResult;

if (argc < 3) {
ExitError("Usage: Provide a username and password");
}
authResult = AuthenticateUser(argv[1], argv[2]);
if (authResult != AUTH_SUCCESS) {
ExitError("Authentication failed");
}
else {
DoAuthenticatedTask(argv[1]);
}
}
```

In AuthenticateUser(), because sizeof() is applied to a parameter with an array type, the sizeof() call might return 4 on many modern architectures. As a result, the strncmp() call only checks the first four characters of the input password, resulting in a partial comparison (CWE-187), leading to improper authentication (CWE-287).

Because of the partial comparison, any of these passwords would still cause authentication to succeed for the "admin" user:

(Attack)

```
pass5
passABCDEFGH
passWORD
```

Because only 4 characters are checked, this significantly reduces the search space for an attacker, making brute force attacks more feasible.

The same problem also applies to the username, so values such as "adminXYZ" and "administrator" will succeed for the username.

Potential Mitigations

Phase: Implementation

Use expressions such as "sizeof(*pointer)" instead of "sizeof(pointer)", unless you intend to run sizeof() on a pointer type to gain some platform independence or if you are allocating a variable on the stack.

Other Notes

The use of sizeof() on a pointer can sometimes generate useful information. An obvious case is to find out the wordsize on a platform. More often than not, the appearance of sizeof(pointer) indicates a bug.

Weakness Ordinalities

Ordinality	Description
Primary	*(where the weakness exists independent of other weaknesses)*

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	465	Pointer Issues	**Development Concepts (primary)699**
ChildOf	Weakness Class	682	Incorrect Calculation	**Research Concepts (primary)1000**
ChildOf	Category	737	CERT C Secure Coding Section 03 - Expressions (EXP)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
ChildOf	Category	740	CERT C Secure Coding Section 06 - Arrays (ARR)	Weaknesses Addressed by the CERT C Secure Coding Standard734
CanPrecede	Weakness Base	131	Incorrect Calculation of Buffer Size	Research Concepts1000

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Use of sizeof() on a pointer type
CERT C Secure Coding	ARR01-C		Do not apply the sizeof operator to a pointer when taking the size of an array
CERT C Secure Coding	EXP01-C		Do not take the size of a pointer to determine the size of the pointed-to type

White Box Definitions

A weakness where code path has:

1. end statement that passes an identity of a dynamically allocated memory resource to a sizeof operator

2. start statement that allocates the dynamically allocated memory resource

References

Robert Seacord. "EXP01-A. Do not take the sizeof a pointer to determine the size of a type". <https://www.securecoding.cert.org/confluence/display/seccode/EXP01-A.+Do+not+take+the+sizeof+a+pointer+to+determine+the+size+of+a+type>.

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	CLASP		Externally Mined

Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Eric Dalci	Cigital	External
updated Time of Introduction			
2008-08-01		KDM Analytics	External
added/updated white box definitions			
2008-09-08	CWE Content Team	MITRE	Internal
updated Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities			
2008-11-24	CWE Content Team	MITRE	Internal
updated Relationships, Taxonomy Mappings			
2009-03-10	CWE Content Team	MITRE	Internal
updated Demonstrative Examples			
2009-12-28	CWE Content Team	MITRE	Internal
updated Demonstrative Examples			
2010-02-16	CWE Content Team	MITRE	Internal
updated Relationships			

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

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.

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**

```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**

```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**
```java
public String getValue(int index) {
return array[index];
}
```

If index is outside of the range of the array, this may result in an ArrayIndexOutOfBounds Exception being raised.

Example 3

In the following Java example the method displayProductSummary is called from a Web service servlet to retrieve product summary information for display to the user. The servlet obtains the integer value of the product number from the user and passes it to the displayProductSummary method. The displayProductSummary method passes the integer value of the product number to the getProductSummary method which obtains the product summary from the array object containing the project summaries using the integer value of the product number as the array index.

(Bad Code)
Example Language: **Java**
```java
// Method called from servlet to obtain product information
public String displayProductSummary(int index) {

String productSummary = new String("");

try {
String productSummary = getProductSummary(index);

} catch (Exception ex) {...}

return productSummary;
}

public String getProductSummary(int index) {
return products[index];
}
```

In this example the integer value used as the array index that is provided by the user may be outside the allowable range of indices for the array which may provide unexpected results or may comes the application to fail. The integer value used for the array index should be validated to ensure that it is within the allowable range of indices for the array as in the following code.

(Good Code)
Example Language: **Java**
```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.

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.

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

TOCTOU

Risk

What might happen

At best, a Race Condition may cause errors in accuracy, overidden values or unexpected behavior that may result in denial-of-service. At worst, it may allow attackers to retrieve data or bypass security processes by replaying a controllable Race Condition until it plays out in their favor.

Cause

How does it happen

Race Conditions occur when a public, single instance of a resource is used by multiple concurrent logical processes. If the these logical processes attempt to retrieve and update the resource without a timely management system, such as a lock, a Race Condition will occur.

An example for when a Race Condition occurs is a resource that may return a certain value to a process for further editing, and then updated by a second process, resulting in the original process' data no longer being valid. Once the original process edits and updates the incorrect value back into the resource, the second process' update has been overwritten and lost.

General Recommendations

How to avoid it

When sharing resources between concurrent processes across the application ensure that these resources are either thread-safe, or implement a locking mechanism to ensure expected concurrent activity.

Source Code Examples

Java
Different Threads Increment and Decrement The Same Counter Repeatedly, Resulting in a Race Condition

```java
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	010584964565407	1/6/2025