vul_files_81 Scan Report

Project Name	vul_files_81
Scan Start	Thursday, January 9, 2025 1:15:26 PM
Preset	Checkmarx Default
Scan Time	01h:52m:09s
Lines Of Code Scanned	290594
Files Scanned	49
Report Creation Time	Thursday, January 9, 2025 2:45:05 PM
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075
Team	CxServer
Checkmarx Version	8.7.0
Scan Type	Full
Source Origin	LocalPath
Density	1/1000 (Vulnerabilities/LOC)
Visibility	Public

Filter Settings

Severity

Included: High, Medium, Low, Information

Excluded: None

Result State

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

Excluded: None

Assigned to

Included: All

Categories

Included:

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

Excluded:

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

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

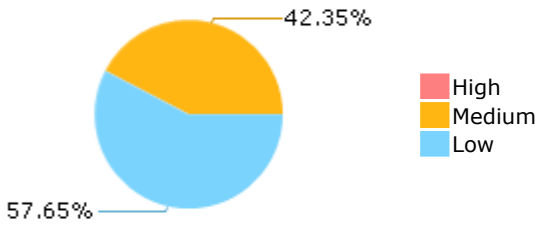
Results Limit

Results limit per query was set to 50

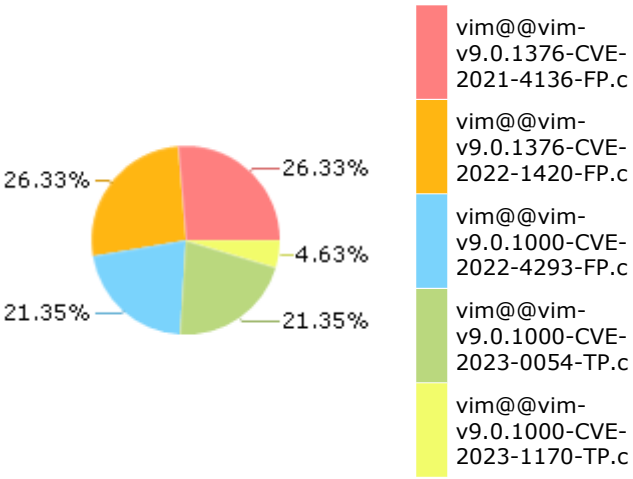
Selected Queries

Selected queries are listed in Result Summary

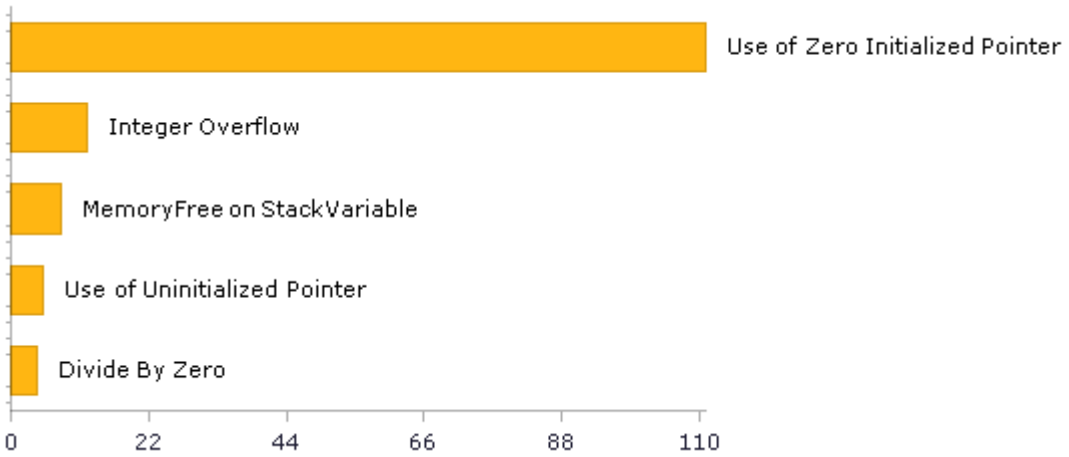
Result Summary



Most Vulnerable Files



Top 5 Vulnerabilities



Scan Summary - OWASP Top 10 2017

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

Category	Threat Agent	Exploitability	Weakness Prevalence	Weakness Detectability	Technical Impact	Business Impact	Issues Found	Best Fix Locations
A1-Injection	App. Specific	EASY	COMMON	EASY	SEVERE	App. Specific	134	30
A2-Broken Authentication	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	4	4
A3-Sensitive Data Exposure	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	App. Specific	0	0
A4-XML External Entities (XXE)	App. Specific	AVERAGE	COMMON	EASY	SEVERE	App. Specific	0	0
A5-Broken Access Control*	App. Specific	AVERAGE	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A6-Security Misconfiguration	App. Specific	EASY	WIDESPREAD	EASY	MODERATE	App. Specific	0	0
A7-Cross-Site Scripting (XSS)	App. Specific	EASY	WIDESPREAD	EASY	MODERATE	App. Specific	0	0
A8-Insecure Deserialization	App. Specific	DIFFICULT	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A9-Using Components with Known Vulnerabilities*	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	MODERATE	App. Specific	3	3
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	3	3
A10-Unvalidated Redirects and Forwards	USERS BROWSERS	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	AFFECTED DATA AND FUNCTIONS	0	0

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

Scan Summary - PCI DSS v3.2

Category	Issues Found	Best Fix Locations
PCI DSS (3.2) - 6.5.1 - Injection flaws - particularly SQL injection	0	0
PCI DSS (3.2) - 6.5.2 - Buffer overflows	12	12
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.	11	11
Configuration Management	Organizations must: (i) establish and maintain baseline configurations and inventories of organizational information systems (including hardware, software, firmware, and documentation) throughout the respective system development life cycles; and (ii) establish and enforce security configuration settings for information technology products employed in organizational information systems.	0	0
Identification And Authentication*	Organizations must identify information system users, processes acting on behalf of users, or devices and authenticate (or verify) the identities of those users, processes, or devices, as a prerequisite to allowing access to organizational information systems.	2	2
Media Protection	Organizations must: (i) protect information system media, both paper and digital; (ii) limit access to information on information system media to authorized users; and (iii) sanitize or destroy information system media before disposal or release for reuse.	0	0
System And Communications Protection	Organizations must: (i) monitor, control, and protect organizational communications (i.e., information transmitted or received by organizational information systems) at the external boundaries and key internal boundaries of the information systems; and (ii) employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational information systems.	0	0
System And Information Integrity	Organizations must: (i) identify, report, and correct information and information system flaws in a timely manner; (ii) provide protection from malicious code at appropriate locations within organizational information systems; and (iii) monitor information system security alerts and advisories and take appropriate actions in response.	12	12

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

Scan Summary - NIST SP 800-53

Category	Issues Found	Best Fix Locations
AC-12 Session Termination (P2)	0	0
AC-3 Access Enforcement (P1)	4	4
AC-4 Information Flow Enforcement (P1)	0	0
AC-6 Least Privilege (P1)	0	0
AU-9 Protection of Audit Information (P1)	0	0
CM-6 Configuration Settings (P2)	0	0
IA-5 Authenticator Management (P1)	0	0
IA-6 Authenticator Feedback (P2)	0	0
IA-8 Identification and Authentication (Non-Organizational Users) (P1)	0	0
SC-12 Cryptographic Key Establishment and Management (P1)	0	0
SC-13 Cryptographic Protection (P1)	0	0
SC-17 Public Key Infrastructure Certificates (P1)	0	0
SC-18 Mobile Code (P2)	0	0
SC-23 Session Authenticity (P1)*	0	0
SC-28 Protection of Information at Rest (P1)	0	0
SC-4 Information in Shared Resources (P1)	0	0
SC-5 Denial of Service Protection (P1)*	262	79
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*	24	24
SI-11 Error Handling (P2)*	0	0
SI-15 Information Output Filtering (P0)	0	0
SI-16 Memory Protection (P1)	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 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

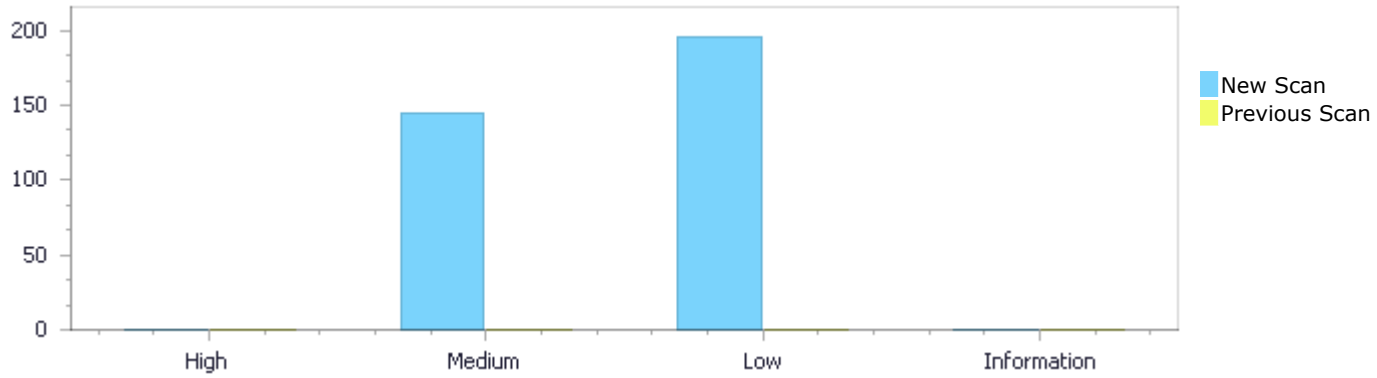
Scan Summary - Custom

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	144	196	0	340
Recurrent Issues	0	0	0	0	0
Total	0	144	196	0	340
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	144	196	0	340
Urgent	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0
Total	0	144	196	0	340

Result Summary

Vulnerability Type	Occurrences	Severity
Use of Zero Initialized Pointer	111	Medium
Integer Overflow	12	Medium
MemoryFree on StackVariable	8	Medium
Use of Uninitialized Pointer	5	Medium
Divide By Zero	4	Medium

Dangerous Functions	3	Medium
Memory Leak	1	Medium
NULL Pointer Dereference	134	Low
Use of Sizeof On a Pointer Type	33	Low
Unchecked Array Index	12	Low
Arithmenic Operation On Boolean	11	Low
Improper Resource Access Authorization	2	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-v9.0.1376-CVE-2021-4136-FP.c	36
vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	36
vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	23
vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	23
vim@@vim-v9.0.1000-CVE-2023-0433-TP.c	11
vim@@vim-v9.0.1000-CVE-2023-1170-TP.c	5
vim@@vim-v9.0.1000-CVE-2023-1175-TP.c	5
vim@@vim-v9.0.1000-CVE-2023-2609-TP.c	5

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=1050086&projectid=50075&pathid=45
Status	New

The variable declared in ht at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 1011 is not initialized when it is used by ll_tv at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 1011.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	1029	1179
Object	ht	ll_tv

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method get_lval(

```
....
1029.       hashtab_T    *ht = NULL;
....
1179.        lp->ll_tv = &v->di_tv;
```

Use of Zero Initialized Pointer\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=46
Status	New

The variable declared in ht_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5261 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5261.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-	vim@@vim-v9.0.1000-CVE-2022-4293-

	FP.c	FP.c
Line	5267	5294
Object	ht_stack	ht_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
5267.       ht_stack_T    *ht_stack = NULL;
....
5294.        ht_stack = ht_stack->prev;
```

Use of Zero Initialized Pointer\Path 3:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=47
Status New

The variable declared in ht_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5261 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5261.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5267	5292
Object	ht_stack	ht_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
5267.       ht_stack_T    *ht_stack = NULL;
....
5292.        cur_ht = ht_stack->ht;
```

Use of Zero Initialized Pointer\Path 4:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=48
Status New

The variable declared in list_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5341 is not initialized when it is used by list_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5341.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5346	5365
Object	list_stack	list_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
5346.       list_stack_T *list_stack = NULL;
....
5365.        list_stack = list_stack->prev;
```

Use of Zero Initialized Pointer\Path 5:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=49
Status New

The variable declared in list_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5341 is not initialized when it is used by list_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5341.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5346	5363
Object	list_stack	list_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
5346.       list_stack_T *list_stack = NULL;
....
5363.        cur_l = list_stack->list;
```

Use of Zero Initialized Pointer\Path 6:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=50
Status New

The variable declared in ht at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 1011 is not initialized when it is used by ll_tv at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 1011.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	1029	1179
Object	ht	ll_tv

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method get_lval(

```
....
1029.       hashtab_T    *ht = NULL;
....
1179.        lp->ll_tv = &v->di_tv;
```

Use of Zero Initialized Pointer\Path 7:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=51
Status New

The variable declared in ht_stack at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5261 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5261.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	5267	5294
Object	ht_stack	ht_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
5267.       ht_stack_T    *ht_stack = NULL;
....
5294.        ht_stack = ht_stack->prev;
```

Use of Zero Initialized Pointer\Path 8:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=52

Status	New

The variable declared in ht_stack at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5261 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5261.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	5267	5292
Object	ht_stack	ht_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
5267.       ht_stack_T    *ht_stack = NULL;
....
5292.        cur_ht = ht_stack->ht;
```

Use of Zero Initialized Pointer\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=53
Status	New

The variable declared in list_stack at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5341 is not initialized when it is used by list_stack at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5341.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	5346	5365
Object	list_stack	list_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
5346.       list_stack_T *list_stack = NULL;
....
5365.        list_stack = list_stack->prev;
```

Use of Zero Initialized Pointer\Path 10:

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

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=54
Status	New

The variable declared in list_stack at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5341 is not initialized when it is used by list_stack at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5341.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	5346	5363
Object	list_stack	list_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
5346.       list_stack_T *list_stack = NULL;
....
5363.        cur_l = list_stack->list;
```

Use of Zero Initialized Pointer\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=55
Status	New

The variable declared in next_leader_flags at vim@@vim-v9.0.1000-CVE-2023-0433-TP.c in line 958 is not initialized when it is used by flags at vim@@vim-v9.0.1000-CVE-2023-0433-TP.c in line 477.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Line	1027	495
Object	next_leader_flags	flags

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Method format_lines(

```
....
1027.           next_leader_flags = NULL;
```

File Name vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Method fmt_check_par(

```
....
495.        flags = *leader_flags;
```

Use of Zero Initialized Pointer\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=56
Status	New

The variable declared in leader_flags at vim@@vim-v9.0.1000-CVE-2023-0433-TP.c in line 958 is not initialized when it is used by flags at vim@@vim-v9.0.1000-CVE-2023-0433-TP.c in line 477.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Line	970	495
Object	leader_flags	flags

Code Snippet

File Name vim@@vim-v9.0.1000-CVE-2023-0433-TP.c

Method format_lines(

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

File Name vim@@vim-v9.0.1000-CVE-2023-0433-TP.c

Method fmt_check_par(

```
....
495.        flags = *leader_flags;
```

Use of Zero Initialized Pointer\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=57
Status	New

The variable declared in next_leader_flags at vim@@vim-v9.0.1000-CVE-2023-0433-TP.c in line 958 is not initialized when it is used by flags at vim@@vim-v9.0.1000-CVE-2023-0433-TP.c in line 477.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0433-	vim@@vim-v9.0.1000-CVE-2023-0433-

	TP.c	TP.c
Line	971	495
Object	next_leader_flags	flags

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Method format_lines(

```
....
971.      char_u  *next_leader_flags = NULL; // flags for leader of next
line
```

File Name vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Method fmt_check_par(

```
....
495.        flags = *leader_flags;
```

Use of Zero Initialized Pointer\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=58
Status	New

The variable declared in next_leader_flags at vim@@vim-v9.0.1000-CVE-2023-0433-TP.c in line 958 is not initialized when it is used by leader_flags at vim@@vim-v9.0.1000-CVE-2023-0433-TP.c in line 958.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Line	1027	1019
Object	next_leader_flags	leader_flags

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Method format_lines(

```
....
1027.           next_leader_flags = NULL;
....
1019.           leader_flags = next_leader_flags;
```

Use of Zero Initialized Pointer\Path 15:

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

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=59
Status	New

The variable declared in next_leader_flags at vim@@vim-v9.0.1000-CVE-2023-0433-TP.c in line 958 is not initialized when it is used by leader_flags at vim@@vim-v9.0.1000-CVE-2023-0433-TP.c in line 958.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Line	971	1019
Object	next_leader_flags	leader_flags

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Method format_lines(

```
....
971.      char_u  *next_leader_flags = NULL; // flags for leader of next
line
....
1019.           leader_flags = next_leader_flags;
```

Use of Zero Initialized Pointer\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=60
Status	New

The variable declared in ht at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 1012 is not initialized when it is used by ll_tv at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 1012.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	1030	1185
Object	ht	ll_tv

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method get_lval(

```
....
1030.      hashtab_T    *ht = NULL;
....
1185.       lp->ll_tv = &v->di_tv;
```

Use of Zero Initialized Pointer\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=61
Status	New

The variable declared in ll_tv at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 1012 is not initialized when it is used by ll_tv at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 1012.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	1474	1202
Object	ll_tv	ll_tv

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method get_lval(

```
....
1474.           lp->ll_tv = NULL;
....
1202.       vartype_T v_type = lp->ll_tv->v_type;
```

Use of Zero Initialized Pointer\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=62
Status	New

The variable declared in ht_stack at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5400 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5400.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	5406	5433
Object	ht_stack	ht_stack

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
5406.      ht_stack_T   *ht_stack = NULL;
....
5433.       ht_stack = ht_stack->prev;
```

Use of Zero Initialized Pointer\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=63
Status	New

The variable declared in ht_stack at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5400 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5400.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	5406	5431
Object	ht_stack	ht_stack

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
5406.       ht_stack_T    *ht_stack = NULL;
....
5431.        cur_ht = ht_stack->ht;
```

Use of Zero Initialized Pointer\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=64
Status	New

The variable declared in list_stack at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5480 is not initialized when it is used by list_stack at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5480.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	5485	5504
Object	list_stack	list_stack

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
5485.      list_stack_T *list_stack = NULL;
....
5504.       list_stack = list_stack->prev;
```

Use of Zero Initialized Pointer\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=65
Status	New

The variable declared in list_stack at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5480 is not initialized when it is used by list_stack at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5480.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	5485	5502
Object	list_stack	list_stack

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
5485.      list_stack_T *list_stack = NULL;
....
5502.       cur_l = list_stack->list;
```

Use of Zero Initialized Pointer\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=66
Status	New

The variable declared in tf at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5865 is not initialized when it is used by r at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5865.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	6102	5905
Object	tf	r

Code Snippet

File Name	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method	echo_string_core(

```
....
6102.                    char_u *tf = NULL;
....
5905.              r = *tofree;
```

Use of Zero Initialized Pointer\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=67
Status	New

The variable declared in tf at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5865 is not initialized when it is used by r at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5865.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	6102	5932
Object	tf	r

Code Snippet	
File Name	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method	echo_string_core(

```
....
6102.                    char_u *tf = NULL;
....
5932.                  r = *tofree;
```

Use of Zero Initialized Pointer\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=68
Status	New

The variable declared in tf at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5865 is not initialized when it is used by r at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5865.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	6102	5988
Object	tf	r

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method echo_string_core(

```
....
6102.                   char_u *tf = NULL;
....
5988.             r = *tofree;
```

Use of Zero Initialized Pointer\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=69
Status	New

The variable declared in tf at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5865 is not initialized when it is used by r at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5865.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	6102	5993
Object	tf	r

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method echo_string_core(

```
....
6102.                   char_u *tf = NULL;
....
5993.           r = blob2string(tv->vval.v_blob, tofree, numbuf);
```

Use of Zero Initialized Pointer\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=70
Status	New

The variable declared in tf at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5865 is not initialized when it is used by r at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5865.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c

Line	6102	6017
Object	tf	r

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method echo_string_core(

```
....
6102.                   char_u *tf = NULL;
....
6017.             r = *tofree;
```

Use of Zero Initialized Pointer\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=71
Status	New

The variable declared in tf at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5865 is not initialized when it is used by r at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5865.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	6102	6042
Object	tf	r

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method echo_string_core(

```
....
6102.                   char_u *tf = NULL;
....
6042.             r = *tofree;
```

Use of Zero Initialized Pointer\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=72
Status	New

The variable declared in Pointer at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5865 is not initialized when it is used by r at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5865.

	Source	Destination

File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	6057	6063
Object	Pointer	r

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method echo_string_core(

```
....
6057.           *tofree = NULL;
....
6063.             r = *tofree;
```

Use of Zero Initialized Pointer\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=73
Status	New

The variable declared in tf at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5865 is not initialized when it is used by r at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5865.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	6102	6063
Object	tf	r

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method echo_string_core(

```
....
6102.                   char_u *tf = NULL;
....
6063.             r = *tofree;
```

Use of Zero Initialized Pointer\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=74
Status	New

The variable declared in tf at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5865 is not initialized when it is used by r at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 5865.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	6102	6077
Object	tf	r

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method echo_string_core(

```
....
6102.                    char_u *tf = NULL;
....
6077.              r = *tofree = alloc(len);
```

Use of Zero Initialized Pointer\Path 31:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=75
Status New

The variable declared in ht at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 1012 is not initialized when it is used by ll_tv at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 1012.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	1030	1185
Object	ht	ll_tv

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method get_lval(

```
....
1030.       hashtab_T    *ht = NULL;
....
1185.        lp->ll_tv = &v->di_tv;
```

Use of Zero Initialized Pointer\Path 32:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=76
Status New

The variable declared in ll_tv at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 1012 is not initialized when it is used by ll_tv at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 1012.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	1474	1202
Object	ll_tv	ll_tv

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method get_lval(

```
....
1474.            lp->ll_tv = NULL;
....
1202.        vartype_T v_type = lp->ll_tv->v_type;
```

Use of Zero Initialized Pointer\Path 33:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=77
Status New

The variable declared in ht_stack at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5400 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5400.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	5406	5433
Object	ht_stack	ht_stack

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
5406.       ht_stack_T    *ht_stack = NULL;
....
5433.        ht_stack = ht_stack->prev;
```

Use of Zero Initialized Pointer\Path 34:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=78

Status	New

The variable declared in ht_stack at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5400 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5400.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	5406	5431
Object	ht_stack	ht_stack

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
5406.      ht_stack_T   *ht_stack = NULL;
....
5431.       cur_ht = ht_stack->ht;
```

Use of Zero Initialized Pointer\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=79
Status	New

The variable declared in list_stack at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5480 is not initialized when it is used by list_stack at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5480.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	5485	5504
Object	list_stack	list_stack

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
5485.      list_stack_T *list_stack = NULL;
....
5504.       list_stack = list_stack->prev;
```

Use of Zero Initialized Pointer\Path 36:

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

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=80
Status	New

The variable declared in list_stack at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5480 is not initialized when it is used by list_stack at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5480.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	5485	5502
Object	list_stack	list_stack

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
5485.       list_stack_T *list_stack = NULL;
....
5502.        cur_l = list_stack->list;
```

Use of Zero Initialized Pointer\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=81
Status	New

The variable declared in tf at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5865 is not initialized when it is used by r at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5865.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	6102	5905
Object	tf	r

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method echo_string_core(

```
....
6102.                    char_u *tf = NULL;
....
5905.              r = *tofree;
```

Use of Zero Initialized Pointer\Path 38:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=82
Status	New

The variable declared in tf at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5865 is not initialized when it is used by r at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5865.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	6102	5932
Object	tf	r

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method echo_string_core(

```
....
6102.                    char_u *tf = NULL;
....
5932.                  r = *tofree;
```

Use of Zero Initialized Pointer\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=83
Status	New

The variable declared in tf at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5865 is not initialized when it is used by r at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5865.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	6102	5988
Object	tf	r

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method echo_string_core(

```
....
6102.                    char_u *tf = NULL;
....
5988.              r = *tofree;
```

Use of Zero Initialized Pointer\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=84
Status	New

The variable declared in tf at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5865 is not initialized when it is used by r at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5865.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	6102	5993
Object	tf	r

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method echo_string_core(

```
....
6102.                   char_u *tf = NULL;
....
5993.           r = blob2string(tv->vval.v_blob, tofree, numbuf);
```

Use of Zero Initialized Pointer\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=85
Status	New

The variable declared in tf at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5865 is not initialized when it is used by r at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5865.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	6102	6017
Object	tf	r

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method echo_string_core(

```
....
6102.                    char_u *tf = NULL;
....
6017.              r = *tofree;
```

Use of Zero Initialized Pointer\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=86
Status	New

The variable declared in tf at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5865 is not initialized when it is used by r at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5865.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	6102	6042
Object	tf	r

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method echo_string_core(

```
....
6102.                    char_u *tf = NULL;
....
6042.              r = *tofree;
```

Use of Zero Initialized Pointer\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=87
Status	New

The variable declared in Pointer at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5865 is not initialized when it is used by r at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5865.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	6057	6063
Object	Pointer	r

Code Snippet

File Name	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method	echo_string_core(

```
....
6057.            *tofree = NULL;
....
6063.              r = *tofree;
```

Use of Zero Initialized Pointer\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=88
Status	New

The variable declared in tf at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5865 is not initialized when it is used by r at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5865.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	6102	6063
Object	tf	r

Code Snippet	
File Name	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method	echo_string_core(

```
....
6102.                    char_u *tf = NULL;
....
6063.              r = *tofree;
```

Use of Zero Initialized Pointer\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=89
Status	New

The variable declared in tf at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5865 is not initialized when it is used by r at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 5865.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	6102	6077
Object	tf	r

Code Snippet	
File Name	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method	echo_string_core(

```
....
6102.                    char_u *tf = NULL;
....
6077.              r = *tofree = alloc(len);
```

Use of Zero Initialized Pointer\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=90
Status	New

The variable declared in == at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 1012 is not initialized when it is used by ll_tv at vim@@vim-v9.0.1376-CVE-2021-4136-FP.c in line 1012.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	1391	1444
Object	==	ll_tv

Code Snippet	
File Name	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method	get_lval(

```
....
1391.                            && var_wrong_func_name(key, lp->ll_di ==
NULL))
....
1444.            lp->ll_tv = &lp->ll_di->di_tv;
```

Use of Zero Initialized Pointer\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=91
Status	New

The variable declared in == at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 1012 is not initialized when it is used by ll_tv at vim@@vim-v9.0.1376-CVE-2022-1420-FP.c in line 1012.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c

Line	1391	1444
Object	==	ll_tv

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method get_lval(

```
....
1391.                           && var_wrong_func_name(key, lp->ll_di ==
NULL))
....
1444.           lp->ll_tv = &lp->ll_di->di_tv;
```

Use of Zero Initialized Pointer\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=92
Status	New

The variable declared in retval at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 613 is not initialized when it is used by retval at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 6346.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	630	6366
Object	retval	retval

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method eval_to_string_eap(

```
....
630.        retval = NULL;
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method make_expanded_name(

```
....
6366.       retval = alloc(STRLEN(temp_result) + (expr_start - in_start)
```

Use of Zero Initialized Pointer\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50

	075&pathid=93
Status	New

The variable declared in Pointer at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 6246 is not initialized when it is used by retval at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 6346.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	6260	6366
Object	Pointer	retval

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method find_name_end(

```
....
6260.       *expr_start = NULL;
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method make_expanded_name(

```
....
6366.       retval = alloc(STRLEN(temp_result) + (expr_start - in_start)
```

Use of Zero Initialized Pointer\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=94
Status	New

The variable declared in v_string at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 2225 is not initialized when it is used by s at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 4747.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	2277	4796
Object	v_string	s

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method eval_func(

```
....
2277.       rettv->vval.v_string = NULL;
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method eval_index_inner(

```
....
4796.             char_u      *s = tv_get_string(rettv);
```

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=1050086&projectid=50075&pathid=13
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 2908 of vim@@vim-v9.0.1000-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-v9.0.1000-CVE-2023-1170-TP.c	vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Line	3028	3028
Object	AssignExpr	AssignExpr

Code Snippet

File Name vim@@vim-v9.0.1000-CVE-2023-1170-TP.c

Method str_to_reg(

```
....
3028.           extra += i;
```

Integer Overflow\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50

	075&pathid=14
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1526 of vim@@vim-v9.0.1000-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-v9.0.1000-CVE-2023-1170-TP.c	vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Line	1910	1910
Object	AssignExpr	AssignExpr

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Method do_put(

```
....
1910.           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=1050086&projectid=50075&pathid=15
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1526 of vim@@vim-v9.0.1000-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-v9.0.1000-CVE-2023-1170-TP.c	vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Line	2055	2055
Object	AssignExpr	AssignExpr

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Method do_put(

```
....
2055.             totlen = count * yanklen;
```

Integer Overflow\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50

	075&pathid=16
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1526 of vim@@vim-v9.0.1000-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-v9.0.1000-CVE-2023-1170-TP.c	vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Line	1886	1886
Object	AssignExpr	AssignExpr

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Method do_put(

```
....
1886.              spaces = y_width + 1;
```

Integer Overflow\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=17
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 2908 of vim@@vim-v9.0.1000-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-v9.0.1000-CVE-2023-1175-TP.c	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Line	3028	3028
Object	AssignExpr	AssignExpr

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Method str_to_reg(

```
....
3028.            extra += i;
```

Integer Overflow\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=18

Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1526 of vim@@vim-v9.0.1000-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-v9.0.1000-CVE-2023-1175-TP.c	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Line	1910	1910
Object	AssignExpr	AssignExpr

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Method do_put(

```
....
1910.           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=1050086&projectid=50075&pathid=19
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1526 of vim@@vim-v9.0.1000-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-v9.0.1000-CVE-2023-1175-TP.c	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Line	2055	2055
Object	AssignExpr	AssignExpr

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Method do_put(

```
....
2055.             totlen = count * yanklen;
```

Integer Overflow\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=20

Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1526 of vim@@vim-v9.0.1000-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-v9.0.1000-CVE-2023-1175-TP.c	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Line	1886	1886
Object	AssignExpr	AssignExpr

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Method do_put(

```
....
1886.              spaces = y_width + 1;
```

Integer Overflow\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=21
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 2908 of vim@@vim-v9.0.1000-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-v9.0.1000-CVE-2023-2609-TP.c	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Line	3028	3028
Object	AssignExpr	AssignExpr

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Method str_to_reg(

```
....
3028.            extra += i;
```

Integer Overflow\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=22
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1526 of vim@@vim-v9.0.1000-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-v9.0.1000-CVE-2023-2609-TP.c	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Line	1910	1910
Object	AssignExpr	AssignExpr

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Method do_put(

```
....
1910.           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=1050086&projectid=50075&pathid=23
Status New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1526 of vim@@vim-v9.0.1000-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-v9.0.1000-CVE-2023-2609-TP.c	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Line	2055	2055
Object	AssignExpr	AssignExpr

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Method do_put(

```
....
2055.             totlen = count * yanklen;
```

Integer Overflow\Path 12:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=24
Status New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1526 of vim@@vim-v9.0.1000-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-v9.0.1000-CVE-2023-2609-TP.c	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Line	1886	1886
Object	AssignExpr	AssignExpr

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Method do_put(

```
....
1886.             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=1050086&projectid=50075&pathid=5
Status New

Calling free() (line 5261) on a variable that was not dynamically allocated (line 5261) in file vim@@vim-v9.0.1000-CVE-2022-4293-FP.c may result with a crash.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5295	5295
Object	tempitem	tempitem

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
5295.       free(tempitem);
```

MemoryFree on StackVariable\Path 2:

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

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=6
Status	New

Calling free() (line 5341) on a variable that was not dynamically allocated (line 5341) in file vim@@vim-v9.0.1000-CVE-2022-4293-FP.c may result with a crash.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5366	5366
Object	tempitem	tempitem

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
5366.        free(tempitem);
```

MemoryFree on StackVariable\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=7
Status	New

Calling free() (line 5261) on a variable that was not dynamically allocated (line 5261) in file vim@@vim-v9.0.1000-CVE-2023-0054-TP.c may result with a crash.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	5295	5295
Object	tempitem	tempitem

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
5295.        free(tempitem);
```

MemoryFree on StackVariable\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50

	075&pathid=8
Status	New

Calling free() (line 5341) on a variable that was not dynamically allocated (line 5341) in file vim@@vim-v9.0.1000-CVE-2023-0054-TP.c may result with a crash.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	5366	5366
Object	tempitem	tempitem

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
5366.       free(tempitem);
```

MemoryFree on StackVariable\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=9
Status	New

Calling free() (line 5400) on a variable that was not dynamically allocated (line 5400) in file vim@@vim-v9.0.1376-CVE-2021-4136-FP.c may result with a crash.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	5434	5434
Object	tempitem	tempitem

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
5434.       free(tempitem);
```

MemoryFree on StackVariable\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=10

Status	New

Calling free() (line 5480) on a variable that was not dynamically allocated (line 5480) in file vim@@vim-v9.0.1376-CVE-2021-4136-FP.c may result with a crash.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	5505	5505
Object	tempitem	tempitem

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
5505.       free(tempitem);
```

MemoryFree on StackVariable\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=11
Status	New

Calling free() (line 5400) on a variable that was not dynamically allocated (line 5400) in file vim@@vim-v9.0.1376-CVE-2022-1420-FP.c may result with a crash.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	5434	5434
Object	tempitem	tempitem

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method set_ref_in_ht(hashtab_T *ht, int copyID, list_stack_T **list_stack)

```
....
5434.       free(tempitem);
```

MemoryFree on StackVariable\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=12
Status	New

Calling free() (line 5480) on a variable that was not dynamically allocated (line 5480) in file vim@@vim-v9.0.1376-CVE-2022-1420-FP.c may result with a crash.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	5505	5505
Object	tempitem	tempitem

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method set_ref_in_list_items(list_T *l, int copyID, ht_stack_T **ht_stack)

```
....
5505.        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=1050086&projectid=50075&pathid=40
Status	New

The variable declared in wp at vim@@vim-v9.0.1000-CVE-2023-0433-TP.c in line 824 is not initialized when it is used by w_old_cursor_lnum at vim@@vim-v9.0.1000-CVE-2023-0433-TP.c in line 824.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Line	878	886
Object	wp	w_old_cursor_lnum

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Method op_format(

```
....
878.        win_T *wp;
....
886.              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=1050086&projectid=50075&pathid=41
Status	New

The variable declared in wp at vim@@vim-v9.0.1000-CVE-2023-0433-TP.c in line 824 is not initialized when it is used by w_old_cursor_lnum at vim@@vim-v9.0.1000-CVE-2023-0433-TP.c in line 824.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Line	878	882
Object	wp	w_old_cursor_lnum

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Method op_format(

```
....
878.        win_T *wp;
....
882.            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=1050086&projectid=50075&pathid=42
Status	New

The variable declared in wp at vim@@vim-v9.0.1000-CVE-2023-0433-TP.c in line 824 is not initialized when it is used by w_old_visual_lnum at vim@@vim-v9.0.1000-CVE-2023-0433-TP.c in line 824.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Line	878	886
Object	wp	w_old_visual_lnum

Code Snippet

File Name	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Method	op_format(

```
....
878.        win_T *wp;
....
886.              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=1050086&projectid=50075&pathid=43
Status	New

The variable declared in wp at vim@@vim-v9.0.1000-CVE-2023-0433-TP.c in line 824 is not initialized when it is used by w_old_cursor_lnum at vim@@vim-v9.0.1000-CVE-2023-0433-TP.c in line 824.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Line	878	887
Object	wp	w_old_cursor_lnum

Code Snippet

File Name	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Method	op_format(

```
....
878.        win_T *wp;
....
887.                    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=1050086&projectid=50075&pathid=44
Status	New

The variable declared in wp at vim@@vim-v9.0.1000-CVE-2023-0433-TP.c in line 824 is not initialized when it is used by wp at vim@@vim-v9.0.1000-CVE-2023-0433-TP.c in line 824.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Line	878	889
Object	wp	wp

Code Snippet	
File Name	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Method	op_format(

```
....
878.        win_T *wp;
....
889.                    wp->w_old_visual_lnum += old_line_count;
```

Divide By Zero

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

Description

Divide By Zero\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=1
Status	New

The application performs an illegal operation in num_divide, in vim@@vim-v9.0.1000-CVE-2022-4293-FP.c. In line 50, the program attempts to divide by n2, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input n2 in num_divide of vim@@vim-v9.0.1000-CVE-2022-4293-FP.c, at line 50.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	76	76
Object	n2	n2

Code Snippet	
File Name	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method	num_divide(varnumber_T n1, varnumber_T n2, int *failed)

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

Divide By Zero\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=2
Status	New

The application performs an illegal operation in num_divide, in vim@@vim-v9.0.1000-CVE-2023-0054-TP.c. In line 50, the program attempts to divide by n2, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input n2 in num_divide of vim@@vim-v9.0.1000-CVE-2023-0054-TP.c, at line 50.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	76	76
Object	n2	n2

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method num_divide(varnumber_T n1, varnumber_T n2, int *failed)

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

Divide By Zero\Path 3:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=3
Status New

The application performs an illegal operation in num_divide, in vim@@vim-v9.0.1376-CVE-2021-4136-FP.c. In line 50, the program attempts to divide by n2, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input n2 in num_divide of vim@@vim-v9.0.1376-CVE-2021-4136-FP.c, at line 50.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	76	76
Object	n2	n2

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method num_divide(varnumber_T n1, varnumber_T n2, int *failed)

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

Divide By Zero\Path 4:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=4
Status New

The application performs an illegal operation in num_divide, in vim@@vim-v9.0.1376-CVE-2022-1420-FP.c. In line 50, the program attempts to divide by n2, which might be evaluate to 0 (zero) at time of division. This

value could be a hard-coded zero value, or received from external, untrusted input n2 in num_divide of vim@@vim-v9.0.1376-CVE-2022-1420-FP.c, at line 50.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	76	76
Object	n2	n2

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method num_divide(varnumber_T n1, varnumber_T n2, int *failed)

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

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=1050086&projectid=50075&pathid=36
Status	New

The dangerous function, atoi, was found in use at line 2828 in vim@@vim-v9.0.1000-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-v9.0.1000-CVE-2023-1170-TP.c	vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Line	2857	2857
Object	atoi	atoi

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Method write_reg_contents_ex(

```
....
2857.            int     num = atoi((char *)str);
```

Dangerous Functions\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=37
Status	New

The dangerous function, atoi, was found in use at line 2828 in vim@@vim-v9.0.1000-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-v9.0.1000-CVE-2023-1175-TP.c	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Line	2857	2857
Object	atoi	atoi

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Method write_reg_contents_ex(

```
....
2857.           int     num = atoi((char *)str);
```

Dangerous Functions\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=38
Status	New

The dangerous function, atoi, was found in use at line 2828 in vim@@vim-v9.0.1000-CVE-2023-2609-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-v9.0.1000-CVE-2023-2609-TP.c	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Line	2857	2857
Object	atoi	atoi

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Method write_reg_contents_ex(

```
....
2857.           int     num = atoi((char *)str);
```

Memory Leak

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

Categories

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

Description

Memory Leak\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=39
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c	vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Line	650	650
Object	neW	neW

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0433-TP.c
Method auto_format(

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

NULL Pointer Dereference

Query Path:
CPP\Cx\CPP Low Visibility\NULL Pointer Dereference Version:1

Categories

NIST SP 800-53: SC-5 Denial of Service Protection (P1)
OWASP Top 10 2017: A1-Injection

Description

NULL Pointer Dereference\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=195
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 258 is not initialized when it is used by eap at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 193.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Line	326	201
Object	null	eap

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method eval_expr_typval(

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

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method eval1_emsg(char_u **arg, typval_T *rettv, exarg_T *eap)

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

NULL Pointer Dereference\Path 2:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=196
Status New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 482 is not initialized when it is used by gap at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 482.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	539	539
Object	null	gap

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method skip_expr_concatenate(

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

NULL Pointer Dereference\Path 3:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=197
Status New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 642 is not initialized when it is used by eap at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 613.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	647	624
Object	null	eap

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method eval_to_string(

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

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method eval_to_string_eap(

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

NULL Pointer Dereference\Path 4:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=198
Status New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5081	5425
Object	null	ht_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
5081.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=199
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5086	5425
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method garbage_collect(int testing)

```
....
5086.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=200
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5091	5425
Object	null	ht_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
5091.                   &aucmd_win[i].auc_win->w_winvar.di_tv, copyID,
NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 7:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=201
Status New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5095	5425
Object	null	ht_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
5095.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=202
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5099	5425
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method garbage_collect(int testing)

```
....
5099.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=203
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5105	5425
Object	null	ht_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
5105.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
....
5425.                 newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 10:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=204
Status New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5376 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5385	5425
Object	null	ht_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_callback(callback_T *cb, int copyID)

```
....
5385.      return set_ref_in_item(&tv, copyID, NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
....
5425.                 newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=205
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5081	5425
Object	null	newitem

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
5081.                                                 NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
....
5425.                 newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=206
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Line	5095	5425
Object	null	newitem

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
5095.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=207
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5091	5425
Object	null	newitem

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
5091.                   &aucmd_win[i].auc_win->w_winvar.di_tv, copyID,
NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
....
5425.                 newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=208
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5376 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5385	5425
Object	null	newitem

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_callback(callback_T *cb, int copyID)

```
....
5385.      return set_ref_in_item(&tv, copyID, NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
....
5425.                 newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=209
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Line	5099	5425
Object	null	newitem

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
5099.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=210
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5105	5425
Object	null	newitem

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
5105.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=211
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5086	5425
Object	null	newitem

Code Snippet

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method garbage_collect(int testing)

```
....
5086.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=212
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by list_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5081	5452
Object	null	list_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
5081.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
....
5452.                   newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=213
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by list_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5086	5452
Object	null	list_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
5086.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
....
5452.                   newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 20:

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

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=214
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by list_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5091	5452
Object	null	list_stack

Code Snippet

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method garbage_collect(int testing)

```
....
5091.                   &aucmd_win[i].auc_win->w_winvar.di_tv, copyID,
NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
....
5452.                   newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=215
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by list_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5095	5452
Object	null	list_stack

Code Snippet

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method garbage_collect(int testing)

```
....
5095.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
....
5452.                   newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=216
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by list_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5099	5452
Object	null	list_stack

Code Snippet

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method garbage_collect(int testing)

```
....
5099.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
....
5452.                   newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=217
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by list_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5105	5452
Object	null	list_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
5105.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
....
5452.                   newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 24:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=218
Status New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5376 is not initialized when it is used by list_stack at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5385	5452
Object	null	list_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_callback(callback_T *cb, int copyID)

```
....
5385.       return set_ref_in_item(&tv, copyID, NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
....
5452.                   newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=219
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5081	5452
Object	null	newitem

Code Snippet

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method garbage_collect(int testing)

```
....
5081.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
....
5452.                   newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=220
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5086	5452
Object	null	newitem

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
5086.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
....
5452.                   newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 27:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=221
Status New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5091	5452
Object	null	newitem

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
5091.                   &aucmd_win[i].auc_win->w_winvar.di_tv, copyID,
NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method	set_ref_in_item(

```
....
5452.                   newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=222
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5095	5452
Object	null	newitem

Code Snippet

File Name	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method	garbage_collect(int testing)

```
....
5095.                                                   NULL, NULL);
```

File Name	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method	set_ref_in_item(

```
....
5452.                   newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=223
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-	vim@@vim-v9.0.1000-CVE-2022-4293-

	FP.c	FP.c
Line	5105	5452
Object	null	newitem

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method garbage_collect(int testing)

```
....
5105.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
....
5452.                 newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 30:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=224
Status New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5376 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5385	5452
Object	null	newitem

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_callback(callback_T *cb, int copyID)

```
....
5385.      return set_ref_in_item(&tv, copyID, NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method set_ref_in_item(

```
....
5452.                 newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=225
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5020 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2022-4293-FP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	5099	5452
Object	null	newitem

Code Snippet

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method garbage_collect(int testing)

```
....
5099.                                                 NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c

Method set_ref_in_item(

```
....
5452.                 newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=226
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 258 is not initialized when it is used by eap at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 193.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c

Line	326	201
Object	null	eap

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method eval_expr_typval(

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

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method eval1_emsg(char_u **arg, typval_T *rettv, exarg_T *eap)

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

NULL Pointer Dereference\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=227
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 482 is not initialized when it is used by gap at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 482.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	539	539
Object	null	gap

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method skip_expr_concatenate(

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

NULL Pointer Dereference\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=228
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 642 is not initialized when it is used by eap at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 613.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	647	624
Object	null	eap

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method eval_to_string(

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

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method eval_to_string_eap(

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

NULL Pointer Dereference\Path 35:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=229
Status New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5020 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	5081	5425
Object	null	ht_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
5081.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c

Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=230
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5020 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	5086	5425
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c

Method garbage_collect(int testing)

```
....
5086.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c

Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=231
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5020 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	5091	5425
Object	null	ht_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
5091.                   &aucmd_win[i].auc_win->w_winvar.di_tv, copyID,
NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=232
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5020 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	5095	5425
Object	null	ht_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
5095.                                                       NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c

Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=233
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5020 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	5099	5425
Object	null	ht_stack

Code Snippet

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c

Method garbage_collect(int testing)

```
....
5099.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c

Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=234
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5020 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	5105	5425
Object	null	ht_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
5105.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 41:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=235
Status New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5376 is not initialized when it is used by ht_stack at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	5385	5425
Object	null	ht_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method set_ref_in_callback(callback_T *cb, int copyID)

```
....
5385.       return set_ref_in_item(&tv, copyID, NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
....
5425.                 newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=236
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5020 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	5081	5425
Object	null	newitem

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
5081.                                                 NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
....
5425.                 newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=237
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5020 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c

Line	5095	5425
Object	null	newitem

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
5095.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=238
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5020 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	5091	5425
Object	null	newitem

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
5091.                   &aucmd_win[i].auc_win->w_winvar.di_tv, copyID,
NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 45:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=239
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5376 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	5385	5425
Object	null	newitem

Code Snippet

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c

Method set_ref_in_callback(callback_T *cb, int copyID)

```
....
5385.       return set_ref_in_item(&tv, copyID, NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c

Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=240
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5020 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c

Line	5099	5425
Object	null	newitem

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
5099.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=241
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5020 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	5105	5425
Object	null	newitem

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
5105.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 48:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=242
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5020 is not initialized when it is used by newitem at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	5086	5425
Object	null	newitem

Code Snippet

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c

Method garbage_collect(int testing)

```
....
5086.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c

Method set_ref_in_item(

```
....
5425.                   newitem->prev = *ht_stack;
```

NULL Pointer Dereference\Path 49:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=243
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5020 is not initialized when it is used by list_stack at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	5081	5452
Object	null	list_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
5081.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c

Method set_ref_in_item(

```
....
5452.                   newitem->prev = *list_stack;
```

NULL Pointer Dereference\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=244
Status	New

The variable declared in null at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5020 is not initialized when it is used by list_stack at vim@@vim-v9.0.1000-CVE-2023-0054-TP.c in line 5396.

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	5086	5452
Object	null	list_stack

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method garbage_collect(int testing)

```
....
5086.                                                   NULL, NULL);
```

File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c

Method set_ref_in_item(

```
....
5452.                   newitem->prev = *list_stack;
```

Use of Sizeof On a Pointer Type

Query Path:
CPP\Cx\CPP Low Visibility\Use of Sizeof On a Pointer Type Version:1

Description

Use of Sizeof On a Pointer Type\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=162
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	397	397
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method init_evalarg(evalarg_T *evalarg)

```
....
397.      ga_init2(&evalarg->eval_tofree_ga, sizeof(char_u *), 20);
```

Use of Sizeof On a Pointer Type\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=163
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	500	500
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method skip_expr_concatenate(

```
....
500.        ga_init2(gap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=164
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	504	504
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method skip_expr_concatenate(

```
....
504.        ga_init2(freegap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 4:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=165
Status New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c	vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Line	3711	3711
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2022-4293-FP.c
Method eval8(

```
....
3711.       ga_init2(&type_list, sizeof(type_T *), 10);
```

Use of Sizeof On a Pointer Type\Path 5:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=166
Status New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	397	397

Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method init_evalarg(evalarg_T *evalarg)

```
....
397.      ga_init2(&evalarg->eval_tofree_ga, sizeof(char_u *), 20);
```

Use of Sizeof On a Pointer Type\Path 6:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=167
Status New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	500	500
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method skip_expr_concatenate(

```
....
500.        ga_init2(gap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 7:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=168
Status New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	504	504
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method skip_expr_concatenate(

```
....
504.        ga_init2(freegap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=169
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c	vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Line	3711	3711
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-0054-TP.c
Method eval8(

```
....
3711.       ga_init2(&type_list, sizeof(type_T *), 10);
```

Use of Sizeof On a Pointer Type\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=170
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-1170-TP.c	vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Line	1187	1187
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Method op_yank(oparg_T *oap, int deleting, int mess)

```
....
1187.       y_current->y_array = lalloc_clear(sizeof(char_u *) *
yanklines, TRUE);
```

Use of Sizeof On a Pointer Type\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=171
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-1170-TP.c	vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Line	1502	1502
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Method copy_yank_reg(yankreg_T *reg)

```
....
1502.                                   sizeof(char_u *) * y_current->y_size,
TRUE);
```

Use of Sizeof On a Pointer Type\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=172
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-1170-TP.c	vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Line	2972	2972
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Method str_to_reg(

```
....
2972.       pp = lalloc_clear((y_ptr->y_size + newlines) * sizeof(char_u
*), 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=1050086&projectid=50075&pathid=173

Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Line	1187	1187
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Method op_yank(oparg_T *oap, int deleting, int mess)

```
....
1187.       y_current->y_array = lalloc_clear(sizeof(char_u *) *
yanklines, 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=1050086&projectid=50075&pathid=174
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Line	1502	1502
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Method copy_yank_reg(yankreg_T *reg)

```
....
1502.                               sizeof(char_u *) * y_current->y_size,
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=1050086&projectid=50075&pathid=175
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-1175-	vim@@vim-v9.0.1000-CVE-2023-1175-

	TP.c	TP.c
Line	2972	2972
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Method str_to_reg(

```
....
2972.       pp = lalloc_clear((y_ptr->y_size + newlines) * sizeof(char_u
*), TRUE);
```

Use of Sizeof On a Pointer Type\Path 15:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=176
Status New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Line	1187	1187
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Method op_yank(oparg_T *oap, int deleting, int mess)

```
....
1187.       y_current->y_array = lalloc_clear(sizeof(char_u *) *
yanklines, TRUE);
```

Use of Sizeof On a Pointer Type\Path 16:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=177
Status New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Line	1502	1502
Object	sizeof	sizeof

Code Snippet	
File Name	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Method	copy_yank_reg(yankreg_T *reg)

```
....
1502.                                sizeof(char_u *) * y_current->y_size,
TRUE);
```

Use of Sizeof On a Pointer Type\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=178
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Line	2972	2972
Object	sizeof	sizeof

Code Snippet	
File Name	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Method	str_to_reg(

```
....
2972.      pp = lalloc_clear((y_ptr->y_size + newlines) * sizeof(char_u
*), TRUE);
```

Use of Sizeof On a Pointer Type\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=179
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Line	1397	1397
Object	sizeof	sizeof

Code Snippet	
File Name	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Method	cleanup_subexpr(void)

```
....
1397.           vim_memset(rex.reg_startp, 0, sizeof(char_u *) *
NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=180
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Line	1398	1398
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Method cleanup_subexpr(void)

```
....
1398.           vim_memset(rex.reg_endp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=181
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Line	1418	1418
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Method cleanup_zsubexpr(void)

```
....
1418.           vim_memset(reg_startzp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=182
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Line	1419	1419
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Method cleanup_zsubexpr(void)

```
....
1419.           vim_memset(reg_endzp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=183
Status	New

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	398	398
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method init_evalarg(evalarg_T *evalarg)

```
....
398.      ga_init2(&evalarg->eval_tofree_ga, sizeof(char_u *), 20);
```

Use of Sizeof On a Pointer Type\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=184
Status	New

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	501	501
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method skip_expr_concatenate(

```
....
501.        ga_init2(gap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 24:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=185
Status New

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	505	505
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method skip_expr_concatenate(

```
....
505.        ga_init2(freegap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 25:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=186
Status New

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c	vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Line	3819	3819

Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4136-FP.c
Method eval8(

```
....
3819.       ga_init2(&type_list, sizeof(type_T *), 10);
```

Use of Sizeof On a Pointer Type\Path 26:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=187
Status New

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Line	1398	1398
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Method cleanup_subexpr(void)

```
....
1398.       vim_memset(rex.reg_startp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 27:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=188
Status New

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Line	1399	1399
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Method cleanup_subexpr(void)

```
....
1399.       vim_memset(rex.reg_endp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=189
Status	New

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Line	1419	1419
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Method cleanup_zsubexpr(void)

```
....
1419.       vim_memset(reg_startzp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=190
Status	New

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Line	1420	1420
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Method cleanup_zsubexpr(void)

```
....
1420.       vim_memset(reg_endzp, 0, sizeof(char_u *) * NSUBEXP);
```

Use of Sizeof On a Pointer Type\Path 30:

Severity	Low

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=191
Status	New

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	398	398
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method init_evalarg(evalarg_T *evalarg)

```
....
398.      ga_init2(&evalarg->eval_tofree_ga, sizeof(char_u *), 20);
```

Use of Sizeof On a Pointer Type\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=192
Status	New

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	501	501
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method skip_expr_concatenate(

```
....
501.        ga_init2(gap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=193
Status	New

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	505	505
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method skip_expr_concatenate(

```
....
505.        ga_init2(freegap, sizeof(char_u *), 10);
```

Use of Sizeof On a Pointer Type\Path 33:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=194
Status New

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c	vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Line	3819	3819
Object	sizeof	sizeof

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2022-1420-FP.c
Method eval8(

```
....
3819.       ga_init2(&type_list, sizeof(type_T *), 10);
```

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=1050086&projectid=50075&pathid=329
Status New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-1170-TP.c	vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Line	1467	1467
Object	y_idx	y_idx

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Method yank_copy_line(struct block_def *bd, long y_idx, int exclude_trailing_space)

```
....
1467.       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=1050086&projectid=50075&pathid=330
Status New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-1170-TP.c	vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Line	2721	2721
Object	len	len

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Method get_reg_contents(int regname, int flags)

```
....
2721.       retval[len] = NUL;
```

Unchecked Array Index\Path 3:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=331
Status New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Line	1467	1467

Object	y_idx	y_idx

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Method yank_copy_line(struct block_def *bd, long y_idx, int exclude_trailing_space)

```
....
1467.       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=1050086&projectid=50075&pathid=332
Status New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Line	2721	2721
Object	len	len

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Method get_reg_contents(int regname, int flags)

```
....
2721.       retval[len] = NUL;
```

Unchecked Array Index\Path 5:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=333
Status New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Line	1467	1467
Object	y_idx	y_idx

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Method yank_copy_line(struct block_def *bd, long y_idx, int exclude_trailing_space)

```
....
1467.       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=1050086&projectid=50075&pathid=334
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Line	2721	2721
Object	len	len

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Method get_reg_contents(int regname, int flags)

```
....
2721.        retval[len] = NUL;
```

Unchecked Array Index\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=335
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Line	2552	2552
Object	len	len

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Method reg_submatch(int no)

```
....
2552.                 retval[len] = '\n';
```

Unchecked Array Index\Path 8:

Severity	Low

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=336
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Line	2563	2563
Object	len	len

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Method reg_submatch(int no)

```
....
2563.                    retval[len] = '\n';
```

Unchecked Array Index\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=337
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Line	2571	2571
Object	len	len

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Method reg_submatch(int no)

```
....
2571.                    retval[len] = NUL;
```

Unchecked Array Index\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=338
Status	New

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Line	2552	2552
Object	len	len

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Method reg_submatch(int no)

```
....
2552.                   retval[len] = '\n';
```

Unchecked Array Index\Path 11:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=339
Status New

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Line	2563	2563
Object	len	len

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Method reg_submatch(int no)

```
....
2563.                     retval[len] = '\n';
```

Unchecked Array Index\Path 12:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=340
Status New

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Line	2571	2571

Object	len	len

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Method reg_submatch(int no)

```
....
2571.                   retval[len] = NUL;
```

Arithmenic Operation On Boolean

Query Path:
CPP\Cx\CPP Low Visibility\Arithmenic Operation On Boolean Version:1

Categories

FISMA 2014: Audit And Accountability
NIST SP 800-53: SC-5 Denial of Service Protection (P1)

Description

Arithmenic Operation On Boolean\Path 1:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=25
Status New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-1170-TP.c	vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Line	1773	1773
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Method do_put(

```
....
1773.           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=1050086&projectid=50075&pathid=26
Status New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-1170-	vim@@vim-v9.0.1000-CVE-2023-1170-

	TP.c	TP.c
Line	2213	2213
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Method do_put(

```
....
2213.           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=1050086&projectid=50075&pathid=27
Status New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-1170-TP.c	vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Line	2219	2219
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-1170-TP.c
Method do_put(

```
....
2219.             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=1050086&projectid=50075&pathid=28
Status New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Line	1773	1773
Object	BinaryExpr	BinaryExpr

Code Snippet	
File Name	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Method	do_put(

```
....
1773.           coladvance_force(getviscol() + (dir == FORWARD));
```

Arithmenic Operation On Boolean\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=29
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Line	2213	2213
Object	BinaryExpr	BinaryExpr

Code Snippet	
File Name	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Method	do_put(

```
....
2213.           if (curbuf->b_op_start.lnum + (y_type == MCHAR) - 1 +
nr_lines
```

Arithmenic Operation On Boolean\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=30
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Line	2219	2219
Object	BinaryExpr	BinaryExpr

Code Snippet	
File Name	vim@@vim-v9.0.1000-CVE-2023-1175-TP.c
Method	do_put(

```
....
2219.              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=1050086&projectid=50075&pathid=31
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Line	1773	1773
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Method do_put(

```
....
1773.           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=1050086&projectid=50075&pathid=32
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Line	2213	2213
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Method do_put(

```
....
2213.           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=1050086&projectid=50075&pathid=33
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c	vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Line	2219	2219
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2609-TP.c
Method do_put(

```
....
2219.             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=1050086&projectid=50075&pathid=34
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Line	1351	1351
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Method reg_match_visual(void)

```
....
1351.       if (cols < start || cols > end - (*p_sel == 'e'))
```

Arithmenic Operation On Boolean\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=35

Status	New

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Line	1351	1351
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Method reg_match_visual(void)

```
....
1351.       if (cols < start || cols > end - (*p_sel == 'e'))
```

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=1050086&projectid=50075&pathid=156
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Line	2793	2793
Object	fprintf	fprintf

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....
2793.             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=1050086&projectid=50075&pathid=157
Status	New

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Line	2793	2793
Object	fprintf	fprintf

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....
2793.             fprintf(f, "Syntax error in \"%s\"\n", expr);
```

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=1050086&projectid=50075&pathid=158
Status	New

	Source	Destination
File	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Line	2790	2790
Object	f	f

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....
2790.           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=1050086&projectid=50075&pathid=159
Status	New

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Line	2790	2790
Object	f	f

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....
2790.           f = fopen(BT_REGEXP_DEBUG_LOG_NAME, "a");
```

TOCTOU

Query Path:
CPP\Cx\CPP Low Visibility\TOCTOU Version:1

Description

TOCTOU\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1050086&projectid=50075&pathid=160
Status	New

The vim_regcomp method in vim@@vim-v9.0.1000-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-v9.0.1000-CVE-2023-2610-TP.c	vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Line	2790	2790
Object	fopen	fopen

Code Snippet
File Name vim@@vim-v9.0.1000-CVE-2023-2610-TP.c
Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....
2790.           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=1050086&projectid=50075&pathid=161
Status	New

The vim_regcomp method in vim@@vim-v9.0.1376-CVE-2021-4192-FP.c file utilizes fopen that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c	vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Line	2790	2790
Object	fopen	fopen

Code Snippet
File Name vim@@vim-v9.0.1376-CVE-2021-4192-FP.c
Method vim_regcomp(char_u *expr_arg, int re_flags)

```
....
2790.           f = fopen(BT_REGEXP_DEBUG_LOG_NAME, "a");
```

Divide By Zero

Risk

What might happen

When a program divides a number by zero, an exception will be raised. If this exception is not handled by the application, unexpected results may occur, including crashing the application. This can be considered a DoS (Denial of Service) attack, if an external user has control of the value of the denominator or can cause this error to occur.

Cause

How does it happen

The program receives an unexpected value, and uses it for division without filtering, validation, or verifying that the value is not zero. The application does not explicitly handle this error or prevent division by zero from 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

```
public float getAverage(HttpServletRequest req) {
      int total = Integer.parseInt(req.getParameter("total"));
      int count = Integer.parseInt(req.getParameter("count"));

      return total / count;
}
```

Checked Division

```
public float getAverage(HttpServletRequest req) {
      int total = Integer.parseInt(req.getParameter("total"));
      int count = Integer.parseInt(req.getParameter("count"));

      if (count > 0)
            return total / count;
      else
            return 0;
}
```

MemoryFree on StackVariable

Risk

What might happen

Undefined Behavior may result with a crash. Crashes may give an attacker valuable information about the system and the program internals. Furthermore, it may leave unprotected files (e.g memory) that may be exploited.

Cause

How does it happen

Calling free() on a variable that was not dynamically allocated (e.g. malloc) will result with an Undefined Behavior.

General Recommendations

How to avoid it

Use free() only on dynamically allocated variables in order to prevent unexpected behavior from the compiler.

Source Code Examples

CPP

Bad - Calling free() on a static variable

```
void clean_up(){
  char temp[256];
  do_something();
  free(tmp);
  return;
}
```

Good - Calling free() only on variables that were dynamically allocated

```
void clean_up(){
  char *buff;
  buff = (char*) malloc(1024);
  free(buff);
  return;
}
```

Integer Overflow

Risk

What might happen

Assigning large data types into smaller data types, without proper checks and explicit casting, will lead to undefined behavior and unintentional effects, such as data corruption (e.g. value wraparound, wherein maximum values become minimum values); system crashes; infinite loops; logic errors, such as bypassing of security mechanisms; or even buffer overflows leading to arbitrary code execution.

Cause

How does it happen

This flaw can occur when implicitly casting numerical data types of a larger size, into a variable with a data type of a smaller size. This forces the program to discard some bits of information from the number. Depending on how the numerical data types are stored in memory, this is often the bits with the highest value, causing substantial corruption of the stored number. Alternatively, the sign bit of a signed integer could be lost, completely reversing the intention of the number.

General Recommendations

How to avoid it

- Avoid casting larger data types to smaller types.
- Prefer promoting the target variable to a large enough data type.
- If downcasting is necessary, always check that values are valid and in range of the target type, before casting

---

Source Code Examples

CPP

Unsafe Downsize Casting

```
int unsafe_addition(short op1, int op2) {

    // op2 gets forced from int into a short
    short total = op1 + op2;

    return total;
}
```

Safer Use of Proper Data Types

```
int safe_addition(short op1, int op2) {

    // total variable is of type int, the largest type that is needed
    int total = 0;

    // check if total will overflow available integer size
    if (INT_MAX - abs(op2) > op1)
```

```
    {
        total = op1 + op2;
    }
    else
    {
        // instead of overflow, saturate (but this is not always a good thing)
        total = INT_MAX
    }

    return total;
}
```

Dangerous Functions

Risk

What might happen

Use of dangerous functions may expose varying risks associated with each particular function, with potential impact of improper usage of these functions varying significantly. The presence of such functions indicates a flaw in code maintenance policies and adherence to secure coding practices, in a way that has allowed introducing known dangerous code into the application.

Cause

How does it happen

A dangerous function has been identified within the code. Functions are often deemed dangerous to use for numerous reasons, as there are different sets of vulnerabilities associated with usage of such functions. For example, some string copy and concatenation functions are vulnerable to Buffer Overflow, Memory Disclosure, Denial of Service and more. Use of these functions is not recommended.

General Recommendations

How to avoid it

- Deploy a secure and recommended alternative to any functions that were identified as dangerous.
    - If no secure alternative is found, conduct further researching and testing to identify whether current usage successfully sanitizes and verifies values, and thus successfully avoids the use-cases for whom the function is indeed dangerous
- Conduct a periodical review of methods that are in use, to ensure that all external libraries and built-in functions are up-to-date and whose use has not been excluded from best secure coding practices.

Source Code Examples

CPP

Buffer Overflow in gets()

```
int main()
{
	char buf[10];

	printf("Please enter your name: ");
	gets(buf); // veryveryverylongname
	if (buf == ACCEPTED_NAME)
	{
		// Do something
	}
	return 0;
}
```

Safe reading from user

```
int main()
{
	char buf[10];

	printf("Please enter your name: ");
	fgets(buf, sizeof(buf), stdin); //setting the amount of bytes to read
	if (buf == ACCEPTED_NAME)
	{
		//Do something
	}
	return 0;
}
```

Unsafe function for string copy

```
int main(int argc, char* argv[])
{
	char buf[10];
	strcpy(buf, argv[1]); // overflow occurs when len(argv[1]) > 10 bytes

	return 0;
}
```

Safe string copy

```
int main(int argc, char* argv[])
{
	char buf[10];
	strncpy(buf, argv[1], sizeof(buf));
	buf[9]= '\0'; //strncpy doesn't NULL terminates

	return 0;
}
```

Unsafe format string

```
int main(int argc, char* argv[])
{
	printf(argv[1]); // If argv[1] contains a format token, such as %s,%x or %d, will cause
an access violation
	return 0;
}
```

Safe format string

```
int main(int argc, char* argv[])
{
      printf("%s", argv[1]); // Second parameter is not a formattable string
      return 0;
}
```

Failure to Release Memory Before Removing Last Reference ('Memory Leak')

Weakness ID: 401 *(Weakness Base)* **Status:** Draft

Description

Description Summary

The software does not sufficiently track and release allocated memory after it has been used, which slowly consumes remaining memory.

Extended Description

This is often triggered by improper handling of malformed data or unexpectedly interrupted sessions.

Terminology Notes

"memory leak" has sometimes been used to describe other kinds of issues, e.g. for information leaks in which the contents of memory are inadvertently leaked (CVE-2003-0400 is one such example of this terminology conflict).

Time of Introduction

- Architecture and Design
- Implementation

Applicable Platforms

Languages

C

C++

Modes of Introduction

Memory leaks have two common and sometimes overlapping causes:

- Error conditions and other exceptional circumstances
- Confusion over which part of the program is responsible for freeing the memory

Common Consequences

Scope	Effect
Availability	Most memory leaks result in general software reliability problems, but if an attacker can intentionally trigger a memory leak, the attacker might be able to launch a denial of service attack (by crashing or hanging the program) or take advantage of other unexpected program behavior resulting from a low memory condition.

Likelihood of Exploit

Medium

Demonstrative Examples

Example 1

The following C function leaks a block of allocated memory if the call to read() fails to return the expected number of bytes:

(Bad Code)

Example Language: **C**

```
char* getBlock(int fd) {
char* buf = (char*) malloc(BLOCK_SIZE);
if (!buf) {
return NULL;
}
if (read(fd, buf, BLOCK_SIZE) != BLOCK_SIZE) {

return NULL;
}
```

```
return buf;
}
```

Example 2

Here the problem is that every time a connection is made, more memory is allocated. So if one just opened up more and more connections, eventually the machine would run out of memory.

(Bad Code)

Example Language: **C**

```
bar connection(){
foo = malloc(1024);
return foo;
}
endConnection(bar foo) {

free(foo);
}
int main() {

while(1) //thread 1
//On a connection
foo=connection(); //thread 2
//When the connection ends
endConnection(foo)
}
```

Observed Examples

Reference	Description
CVE-2005-3119	Memory leak because function does not free() an element of a data structure.
CVE-2004-0427	Memory leak when counter variable is not decremented.
CVE-2002-0574	Memory leak when counter variable is not decremented.
CVE-2005-3181	Kernel uses wrong function to release a data structure, preventing data from being properly tracked by other code.
CVE-2004-0222	Memory leak via unknown manipulations as part of protocol test suite.
CVE-2001-0136	Memory leak via a series of the same command.

Potential Mitigations

Pre-design: Use a language or compiler that performs automatic bounds checking.

Phase: Architecture and Design

Use an abstraction library to abstract away risky APIs. Not a complete solution.

Pre-design through Build: The Boehm-Demers-Weiser Garbage Collector or valgrind can be used to detect leaks in code. This is not a complete solution as it is not 100% effective.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	398	Indicator of Poor Code Quality	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Category	399	Resource Management Errors	**Development Concepts (primary)699**
ChildOf	Category	633	Weaknesses that Affect Memory	**Resource-specific Weaknesses (primary)631**
ChildOf	Category	730	OWASP Top Ten 2004 Category A9 - Denial of Service	**Weaknesses in OWASP Top Ten (2004) (primary)711**
ChildOf	Weakness Base	772	Missing Release of Resource after Effective	**Research Concepts (primary)1000**

			Lifetime	
MemberOf	View	630	Weaknesses Examined by SAMATE	**Weaknesses Examined by SAMATE (primary)630**
CanFollow	Weakness Class	390	Detection of Error Condition Without Action	Research Concepts1000

Relationship Notes

This is often a resultant weakness due to improper handling of malformed data or early termination of sessions.

Affected Resources

- Memory

Functional Areas

- Memory management

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
PLOVER			Memory leak
7 Pernicious Kingdoms			Memory Leak
CLASP			Failure to deallocate data
OWASP Top Ten 2004	A9	CWE More Specific	Denial of Service

White Box Definitions

A weakness where the code path has:

1. start statement that allocates dynamically allocated memory resource

2. end statement that loses identity of the dynamically allocated memory resource creating situation where dynamically allocated memory resource is never relinquished

Where "loses" is defined through the following scenarios:

1. identity of the dynamic allocated memory resource never obtained

2. the statement assigns another value to the data element that stored the identity of the dynamically allocated memory resource and there are no aliases of that data element

3. identity of the dynamic allocated memory resource obtained but never passed on to function for memory resource release

4. the data element that stored the identity of the dynamically allocated resource has reached the end of its scope at the statement and there are no aliases of that data element

References

J. Whittaker and H. Thompson. "How to Break Software Security". Addison Wesley. 2003.

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	PLOVER		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-08-15		Veracode	External
	Suggested OWASP Top Ten 2004 mapping		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Relationships, Other Notes, References, Relationship Notes, Taxonomy Mappings, Terminology Notes		
2008-10-14	CWE Content Team	MITRE	Internal
	updated Description		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Other Notes		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Name		
2009-07-17	KDM Analytics		External
	Improved the White Box Definition		

2009-07-27	CWE Content Team	MITRE	Internal
	updated White Box Definitions		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Modes of Introduction, Other Notes		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		

Previous Entry Names

Change Date	Previous Entry Name
2008-04-11	Memory Leak
2009-05-27	Failure to Release Memory Before Removing Last Reference (aka 'Memory Leak')

BACK TO TOP

Use of Uninitialized Pointer

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
- Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
- Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.

Source Code Examples

Use of Zero Initialized Pointer

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
- Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
- Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.

Source Code Examples

CPP

Explicit NULL Dereference

```
char * input = NULL;
printf("%s", input);
```

Implicit NULL Dereference

```
char * input;
printf("%s", input);
```

Java

Explicit Null Dereference

```
Object o = null;
out.println(o.getClass());
```

Indicator of Poor Code Quality

Weakness ID: 398 *(Weakness Class)* **Status:** Draft

Description

Description Summary

The code has features that do not directly introduce a weakness or vulnerability, but indicate that the product has not been carefully developed or maintained.

Extended Description

Programs are more likely to be secure when good development practices are followed. If a program is complex, difficult to maintain, not portable, or shows evidence of neglect, then there is a higher likelihood that weaknesses are buried in the code.

Time of Introduction

- Architecture and Design
- Implementation

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	18	Source Code	**Development Concepts (primary)699**
ChildOf	Weakness Class	710	Coding Standards Violation	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	107	Struts: Unused Validation Form	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	110	Struts: Validator Without Form Field	**Research Concepts (primary)1000**
ParentOf	Category	399	Resource Management Errors	**Development Concepts (primary)699**
ParentOf	Weakness Base	401	Failure to Release Memory Before Removing Last Reference ('Memory Leak')	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	404	Improper Resource Shutdown or Release	Development Concepts699 **Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Variant	415	Double Free	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	416	Use After Free	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Variant	457	Use of Uninitialized Variable	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	474	Use of Function with Inconsistent Implementations	**Development Concepts (primary)699 Seven Pernicious Kingdoms (primary)700 Research Concepts (primary)1000**
ParentOf	Weakness Base	475	Undefined Behavior for Input to API	**Development Concepts (primary)699 Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	476	NULL Pointer Dereference	**Development Concepts**

				(primary)699 Seven Pernicious Kingdoms (primary)700 Research Concepts (primary)1000
ParentOf	Weakness Base	477	Use of Obsolete Functions	**Development Concepts (primary)699 Seven Pernicious Kingdoms (primary)700 Research Concepts (primary)1000**
ParentOf	Weakness Variant	478	Missing Default Case in Switch Statement	**Development Concepts (primary)699**
ParentOf	Weakness Variant	479	Unsafe Function Call from a Signal Handler	**Development Concepts (primary)699**
ParentOf	Weakness Variant	483	Incorrect Block Delimitation	**Development Concepts (primary)699**
ParentOf	Weakness Base	484	Omitted Break Statement in Switch	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Variant	546	Suspicious Comment	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Weakness Variant	547	Use of Hard-coded, Security-relevant Constants	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Weakness Variant	561	Dead Code	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Weakness Base	562	Return of Stack Variable Address	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Variant	563	Unused Variable	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Category	569	Expression Issues	**Development Concepts (primary)699**
ParentOf	Weakness Variant	585	Empty Synchronized Block	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Weakness Variant	586	Explicit Call to Finalize()	**Development Concepts (primary)699**
ParentOf	Weakness Variant	617	Reachable Assertion	**Development Concepts (primary)699**
ParentOf	Weakness Base	676	Use of Potentially Dangerous Function	**Development Concepts (primary)699 Research Concepts (primary)1000**
MemberOf	View	700	Seven Pernicious Kingdoms	**Seven Pernicious Kingdoms (primary)700**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name

7 Pernicious Kingdoms			Code Quality

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	7 Pernicious Kingdoms		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Eric Dalci	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**

```
sub DisplayPrivateMessage {
my($id) = @_;
my $Message = LookupMessageObject($id);
print "From: " . encodeHTML($Message->{from}) . "<br>\n";
print "Subject: " . encodeHTML($Message->{subject}) . "\n";
print "<hr>\n";
print "Body: " . encodeHTML($Message->{body}) . "\n";
}

my $q = new CGI;
# For purposes of this example, assume that CWE-309 and
# CWE-523 do not apply.
if (! AuthenticateUser($q->param('username'), $q->param('password'))) {
ExitError("invalid username or password");
}

my $id = $q->param('id');
DisplayPrivateMessage($id);
```

While the program properly exits if authentication fails, it does not ensure that the message is addressed to the user. As a result, an authenticated attacker could provide any arbitrary identifier and read private messages that were intended for other users.

One way to avoid this problem would be to ensure that the "to" field in the message object matches the username of the authenticated user.

Observed Examples

Reference	Description
CVE-2009-3168	Web application does not restrict access to admin scripts, allowing authenticated users to reset administrative passwords.

CVE-2009-2960	Web application does not restrict access to admin scripts, allowing authenticated users to modify passwords of other users.
CVE-2009-3597	Web application stores database file under the web root with insufficient access control (CWE-219), allowing direct request.
CVE-2009-2282	Terminal server does not check authorization for guest access.
CVE-2009-3230	Database server does not use appropriate privileges for certain sensitive operations.
CVE-2009-2213	Gateway uses default "Allow" configuration for its authorization settings.
CVE-2009-0034	Chain: product does not properly interpret a configuration option for a system group, allowing users to gain privileges.
CVE-2008-6123	Chain: SNMP product does not properly parse a configuration option for which hosts are allowed to connect, allowing unauthorized IP addresses to connect.
CVE-2008-5027	System monitoring software allows users to bypass authorization by creating custom forms.
CVE-2008-7109	Chain: reliance on client-side security (CWE-602) allows attackers to bypass authorization using a custom client.
CVE-2008-3424	Chain: product does not properly handle wildcards in an authorization policy list, allowing unintended access.
CVE-2009-3781	Content management system does not check access permissions for private files, allowing others to view those files.
CVE-2008-4577	ACL-based protection mechanism treats negative access rights as if they are positive, allowing bypass of intended restrictions.
CVE-2008-6548	Product does not check the ACL of a page accessed using an "include" directive, allowing attackers to read unauthorized files.
CVE-2007-2925	Default ACL list for a DNS server does not set certain ACLs, allowing unauthorized DNS queries.
CVE-2006-6679	Product relies on the X-Forwarded-For HTTP header for authorization, allowing unintended access by spoofing the header.
CVE-2005-3623	OS kernel does not check for a certain privilege before setting ACLs for files.
CVE-2005-2801	Chain: file-system code performs an incorrect comparison (CWE-697), preventing defauls ACLs from being properly applied.
CVE-2001-1155	Chain: product does not properly check the result of a reverse DNS lookup because of operator precedence (CWE-783), allowing bypass of DNS-based access restrictions.

Potential Mitigations

Phase: Architecture and Design

Divide your application into anonymous, normal, privileged, and administrative areas. Reduce the attack surface by carefully mapping roles with data and functionality. Use role-based access control (RBAC) to enforce the roles at the appropriate boundaries.

Note that this approach may not protect against horizontal authorization, i.e., it will not protect a user from attacking others with the same role.

Phase: Architecture and Design

Ensure that you perform access control checks related to your business logic. These checks may be different than the access control checks that you apply to more generic resources such as files, connections, processes, memory, and database records. For example, a database may restrict access for medical records to a specific database user, but each record might only be intended to be accessible to the patient and the patient's doctor.

Phase: Architecture and Design

Strategy: Libraries or Frameworks

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness

easier to avoid.

For example, consider using authorization frameworks such as the JAAS Authorization Framework and the OWASP ESAPI Access Control feature.

Phase: Architecture and Design

For web applications, make sure that the access control mechanism is enforced correctly at the server side on every page. Users should not be able to access any unauthorized functionality or information by simply requesting direct access to that page.

One way to do this is to ensure that all pages containing sensitive information are not cached, and that all such pages restrict access to requests that are accompanied by an active and authenticated session token associated with a user who has the required permissions to access that page.

Phases: System Configuration; Installation

Use the access control capabilities of your operating system and server environment and define your access control lists accordingly. Use a "default deny" policy when defining these ACLs.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	254	Security Features	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Weakness Class	284	Access Control (Authorization) Issues	**Development Concepts (primary)699 Research Concepts (primary)1000**
ChildOf	Category	721	OWASP Top Ten 2007 Category A10 - Failure to Restrict URL Access	**Weaknesses in OWASP Top Ten (2007) (primary)629**
ChildOf	Category	723	OWASP Top Ten 2004 Category A2 - Broken Access Control	**Weaknesses in OWASP Top Ten (2004) (primary)711**
ChildOf	Category	753	2009 Top 25 - Porous Defenses	**Weaknesses in the 2009 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)750**
ChildOf	Category	803	2010 Top 25 - Porous Defenses	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
ParentOf	Weakness Variant	219	Sensitive Data Under Web Root	**Research Concepts (primary)1000**
ParentOf	Weakness Base	551	Incorrect Behavior Order: Authorization Before Parsing and Canonicalization	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Class	638	Failure to Use Complete Mediation	Research Concepts1000
ParentOf	Weakness Base	804	Guessable CAPTCHA	**Development Concepts (primary)699 Research Concepts (primary)1000**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
7 Pernicious Kingdoms			Missing Access Control
OWASP Top Ten 2007	A10	CWE More Specific	Failure to Restrict URL Access
OWASP Top Ten 2004	A2	CWE More Specific	Broken Access Control

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
1	Accessing Functionality Not Properly Constrained by ACLs	
13	Subverting Environment Variable Values	

17	Accessing, Modifying or Executing Executable Files
87	Forceful Browsing
39	Manipulating Opaque Client-based Data Tokens
45	Buffer Overflow via Symbolic Links
51	Poison Web Service Registry
59	Session Credential Falsification through Prediction
60	Reusing Session IDs (aka Session Replay)
77	Manipulating User-Controlled Variables
76	Manipulating Input to File System Calls
104	Cross Zone Scripting

References

NIST. "Role Based Access Control and Role Based Security". <http://csrc.nist.gov/groups/SNS/rbac/>.

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 4, "Authorization" Page 114; Chapter 6, "Determining Appropriate Access Control" Page 171. 2nd Edition. Microsoft. 2002.

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	7 Pernicious Kingdoms		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-15		Veracode	External
	Suggested OWASP Top Ten 2004 mapping		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Relationships, Other Notes, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences, Description, Likelihood of Exploit, Name, Other Notes, Potential Mitigations, References, Relationships		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Potential Mitigations		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Description, Related Attack Patterns		
2009-07-27	CWE Content Team	MITRE	Internal
	updated Relationships		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Type		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Demonstrative Examples, Detection Factors, Modes of Introduction, Observed Examples, Relationships		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Alternate Terms, Detection Factors, Potential Mitigations, References, Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Potential Mitigations		
Previous Entry Names			
Change Date	**Previous Entry Name**		
2009-01-12	Missing or Inconsistent Access Control		

BACK TO TOP

Incorrect Permission Assignment for Critical Resource

Weakness ID: 732 *(Weakness Class)* **Status:** Draft

Description

Description Summary

The software specifies permissions for a security-critical resource in a way that allows that resource to be read or modified by unintended actors.

Extended Description

When a resource is given a permissions setting that provides access to a wider range of actors than required, it could lead to the disclosure of sensitive information, or the modification of that resource by unintended parties. This is especially dangerous when the resource is related to program configuration, execution or sensitive user data.

Time of Introduction

- Architecture and Design
- Implementation
- Installation
- Operation

Applicable Platforms

Languages

Language-independent

Modes of Introduction

The developer may set loose permissions in order to minimize problems when the user first runs the program, then create documentation stating that permissions should be tightened. Since system administrators and users do not always read the documentation, this can result in insecure permissions being left unchanged.

The developer might make certain assumptions about the environment in which the software runs - e.g., that the software is running on a single-user system, or the software is only accessible to trusted administrators. When the software is running in a different environment, the permissions become a problem.

Common Consequences

Scope	Effect
Confidentiality	An attacker may be able to read sensitive information from the associated resource, such as credentials or configuration information stored in a file.
Integrity	An attacker may be able to modify critical properties of the associated resource to gain privileges, such as replacing a world-writable executable with a Trojan horse.
Availability	An attacker may be able to destroy or corrupt critical data in the associated resource, such as deletion of records from a database.

Likelihood of Exploit

Medium to High

Detection Methods

Automated Static Analysis

Automated static analysis may be effective in detecting permission problems for system resources such as files, directories, shared memory, device interfaces, etc. Automated techniques may be able to detect the use of library functions that modify permissions, then analyze function calls for arguments that contain potentially insecure values.

However, since the software's intended security policy might allow loose permissions for certain operations (such as publishing a file on a web server), automated static analysis may produce some false positives - i.e., warnings that do not have any security consequences or require any code changes.

When custom permissions models are used - such as defining who can read messages in a particular forum in a bulletin board system - these can be difficult to detect using automated static analysis. It may be possible to define custom signatures that

identify any custom functions that implement the permission checks and assignments.

Automated Dynamic Analysis

Automated dynamic analysis may be effective in detecting permission problems for system resources such as files, directories, shared memory, device interfaces, etc.

However, since the software's intended security policy might allow loose permissions for certain operations (such as publishing a file on a web server), automated dynamic analysis may produce some false positives - i.e., warnings that do not have any security consequences or require any code changes.

When custom permissions models are used - such as defining who can read messages in a particular forum in a bulletin board system - these can be difficult to detect using automated dynamic analysis. It may be possible to define custom signatures that identify any custom functions that implement the permission checks and assignments.

Manual Static Analysis

Manual static analysis may be effective in detecting the use of custom permissions models and functions. The code could then be examined to identifying usage of the related functions. Then the human analyst could evaluate permission assignments in the context of the intended security model of the software.

Manual Dynamic Analysis

Manual dynamic analysis may be effective in detecting the use of custom permissions models and functions. The program could then be executed with a focus on exercising code paths that are related to the custom permissions. Then the human analyst could evaluate permission assignments in the context of the intended security model of the software.

Fuzzing

Fuzzing is not effective in detecting this weakness.

Demonstrative Examples

Example 1

The following code sets the umask of the process to 0 before creating a file and writing "Hello world" into the file.

(Bad Code)

Example Language: **C**

```
#define OUTFILE "hello.out"

umask(0);
FILE *out;
/* Ignore CWE-59 (link following) for brevity */
out = fopen(OUTFILE, "w");
if (out) {
fprintf(out, "hello world!\n");
fclose(out);
}
```

After running this program on a UNIX system, running the "ls -l" command might return the following output:

(Result)

```
-rw-rw-rw- 1 username 13 Nov 24 17:58 hello.out
```

The "rw-rw-rw-" string indicates that the owner, group, and world (all users) can read the file and write to it.

Example 2

The following code snippet might be used as a monitor to periodically record whether a web site is alive. To ensure that the file can always be modified, the code uses chmod() to make the file world-writable.

(Bad Code)

Example Language: **Perl**

```
$fileName = "secretFile.out";

if (-e $fileName) {
chmod 0777, $fileName;
}
```

```
my $outFH;
if (! open($outFH, ">>$fileName")) {
ExitError("Couldn't append to $fileName: $!");
}
my $dateString = FormatCurrentTime();
my $status = IsHostAlive("cwe.mitre.org");
print $outFH "$dateString cwe status: $status!\n";
close($outFH);
```

The first time the program runs, it might create a new file that inherits the permissions from its environment. A file listing might look like:

(Result)

```
-rw-r--r-- 1 username 13 Nov 24 17:58 secretFile.out
```

This listing might occur when the user has a default umask of 022, which is a common setting. Depending on the nature of the file, the user might not have intended to make it readable by everyone on the system.

The next time the program runs, however - and all subsequent executions - the chmod will set the file's permissions so that the owner, group, and world (all users) can read the file and write to it:

(Result)

```
-rw-rw-rw- 1 username 13 Nov 24 17:58 secretFile.out
```

Perhaps the programmer tried to do this because a different process uses different permissions that might prevent the file from being updated.

Example 3

The following command recursively sets world-readable permissions for a directory and all of its children:

(Bad Code)

Example Language: **Shell**

```
chmod -R ugo+r DIRNAME
```

If this command is run from a program, the person calling the program might not expect that all the files under the directory will be world-readable. If the directory is expected to contain private data, this could become a security problem.

Observed Examples

Reference	Description
CVE-2009-3482	Anti-virus product sets insecure "Everyone: Full Control" permissions for files under the "Program Files" folder, allowing attackers to replace executables with Trojan horses.
CVE-2009-3897	Product creates directories with 0777 permissions at installation, allowing users to gain privileges and access a socket used for authentication.
CVE-2009-3489	Photo editor installs a service with an insecure security descriptor, allowing users to stop or start the service, or execute commands as SYSTEM.
CVE-2009-3289	Library function copies a file to a new target and uses the source file's permissions for the target, which is incorrect when the source file is a symbolic link, which typically has 0777 permissions.
CVE-2009-0115	Device driver uses world-writable permissions for a socket file, allowing attackers to inject arbitrary commands.
CVE-2009-1073	LDAP server stores a cleartext password in a world-readable file.
CVE-2009-0141	Terminal emulator creates TTY devices with world-writable permissions, allowing an attacker to write to the terminals of other users.

CVE-2008-0662	VPN product stores user credentials in a registry key with "Everyone: Full Control" permissions, allowing attackers to steal the credentials.
CVE-2008-0322	Driver installs its device interface with "Everyone: Write" permissions.
CVE-2009-3939	Driver installs a file with world-writable permissions.
CVE-2009-3611	Product changes permissions to 0777 before deleting a backup; the permissions stay insecure for subsequent backups.
CVE-2007-6033	Product creates a share with "Everyone: Full Control" permissions, allowing arbitrary program execution.
CVE-2007-5544	Product uses "Everyone: Full Control" permissions for memory-mapped files (shared memory) in inter-process communication, allowing attackers to tamper with a session.
CVE-2005-4868	Database product uses read/write permissions for everyone for its shared memory, allowing theft of credentials.
CVE-2004-1714	Security product uses "Everyone: Full Control" permissions for its configuration files.
CVE-2001-0006	"Everyone: Full Control" permissions assigned to a mutex allows users to disable network connectivity.
CVE-2002-0969	Chain: database product contains buffer overflow that is only reachable through a .ini configuration file - which has "Everyone: Full Control" permissions.

Potential Mitigations

Phase: Implementation

When using a critical resource such as a configuration file, check to see if the resource has insecure permissions (such as being modifiable by any regular user), and generate an error or even exit the software if there is a possibility that the resource could have been modified by an unauthorized party.

Phase: Architecture and Design

Divide your application into anonymous, normal, privileged, and administrative areas. Reduce the attack surface by carefully defining distinct user groups, privileges, and/or roles. Map these against data, functionality, and the related resources. Then set the permissions accordingly. This will allow you to maintain more fine-grained control over your resources.

Phases: Implementation; Installation

During program startup, explicitly set the default permissions or umask to the most restrictive setting possible. Also set the appropriate permissions during program installation. This will prevent you from inheriting insecure permissions from any user who installs or runs the program.

Phase: System Configuration

For all configuration files, executables, and libraries, make sure that they are only readable and writable by the software's administrator.

Phase: Documentation

Do not suggest insecure configuration changes in your documentation, especially if those configurations can extend to resources and other software that are outside the scope of your own software.

Phase: Installation

Do not assume that the system administrator will manually change the configuration to the settings that you recommend in the manual.

Phase: Testing

Use tools and techniques that require manual (human) analysis, such as penetration testing, threat modeling, and interactive tools that allow the tester to record and modify an active session. These may be more effective than strictly automated techniques. This is especially the case with weaknesses that are related to design and business rules.

Phase: Testing

Use monitoring tools that examine the software's process as it interacts with the operating system and the network. This technique is useful in cases when source code is unavailable, if the software was not developed by you, or if you want to verify that the build phase did not introduce any new weaknesses. Examples include debuggers that directly attach to the running process; system-call tracing utilities such as truss (Solaris) and strace (Linux); system activity monitors such as FileMon, RegMon, Process Monitor, and other Sysinternals utilities (Windows); and sniffers and protocol analyzers that monitor network traffic.

Attach the monitor to the process and watch for library functions or system calls on OS resources such as files, directories, and shared memory. Examine the arguments to these calls to infer which permissions are being used.

Note that this technique is only useful for permissions issues related to system resources. It is not likely to detect application-level business rules that are related to permissions, such as if a user of a blog system marks a post as "private," but the blog system inadvertently marks it as "public."

Phases: Testing; System Configuration

Ensure that your software runs properly under the Federal Desktop Core Configuration (FDCC) or an equivalent hardening configuration guide, which many organizations use to limit the attack surface and potential risk of deployed software.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	275	Permission Issues	**Development Concepts (primary)699**
ChildOf	Weakness Class	668	Exposure of Resource to Wrong Sphere	**Research Concepts (primary)1000**
ChildOf	Category	753	2009 Top 25 - Porous Defenses	**Weaknesses in the 2009 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)750**
ChildOf	Category	803	2010 Top 25 - Porous Defenses	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
RequiredBy	Compound Element: Composite	689	Permission Race Condition During Resource Copy	Research Concepts1000
ParentOf	Weakness Variant	276	Incorrect Default Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	277	Insecure Inherited Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	278	Insecure Preserved Inherited Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	279	Incorrect Execution-Assigned Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Base	281	Improper Preservation of Permissions	**Research Concepts (primary)1000**

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
232	Exploitation of Privilege/Trust	
1	Accessing Functionality Not Properly Constrained by ACLs	
17	Accessing, Modifying or Executing Executable Files	
60	Reusing Session IDs (aka Session Replay)	
61	Session Fixation	
62	Cross Site Request Forgery (aka Session Riding)	
122	Exploitation of Authorization	
180	Exploiting Incorrectly Configured Access Control Security Levels	
234	Hijacking a privileged process	

References

Mark Dowd, John McDonald and Justin Schuh. "The Art of Software Security Assessment". Chapter 9, "File Permissions." Page 495.. 1st Edition. Addison Wesley. 2006.

John Viega and Gary McGraw. "Building Secure Software". Chapter 8, "Access Control." Page 194.. 1st Edition. Addison-Wesley. 2002.

Maintenance Notes

The relationships between privileges, permissions, and actors (e.g. users and groups) need further refinement within the Research view. One complication is that these concepts apply to two different pillars, related to control of resources (CWE-664) and protection mechanism failures (CWE-396).

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
2008-09-08			Internal CWE Team
	new weakness-focused entry for Research view.		
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2009-01-12	CWE Content Team	MITRE	Internal
	updated Description, Likelihood of Exploit, Name, Potential Mitigations, Relationships		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Potential Mitigations, Related Attack Patterns		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Name		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Demonstrative Examples, Detection Factors, Modes of Introduction, Observed Examples, Potential Mitigations, References		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Potential Mitigations, Related Attack Patterns		
Previous Entry Names			
Change Date	**Previous Entry Name**		
2009-01-12	Insecure Permission Assignment for Resource		
2009-05-27	Insecure Permission Assignment for Critical Resource		

BACK TO TOP

TOCTOU

Risk

What might happen

At best, a Race Condition may cause errors in accuracy, overidden values or unexpected behavior that may result in denial-of-service. At worst, it may allow attackers to retrieve data or bypass security processes by replaying a controllable Race Condition until it plays out in their favor.

Cause

How does it happen

Race Conditions occur when a public, single instance of a resource is used by multiple concurrent logical processes. If the these logical processes attempt to retrieve and update the resource without a timely management system, such as a lock, a Race Condition will occur.

An example for when a Race Condition occurs is a resource that may return a certain value to a process for further editing, and then updated by a second process, resulting in the original process' data no longer being valid. Once the original process edits and updates the incorrect value back into the resource, the second process' update has been overwritten and lost.

General Recommendations

How to avoid it

When sharing resources between concurrent processes across the application ensure that these resources are either thread-safe, or implement a locking mechanism to ensure expected concurrent activity.

Source Code Examples

Java

Different Threads Increment and Decrement The Same Counter Repeatedly, Resulting in a Race Condition

```
	public static int counter = 0;
	public static void start() throws InterruptedException {
		incrementCounter ic;
		decrementCounter dc;
		while(counter == 0) {
			counter = 0;
			ic = new incrementCounter();
			dc = new decrementCounter();
			ic.start();
			dc.start();
			ic.join();
			dc.join();
		}
		System.out.println(counter); //Will stop and return either -1 or 1 due to race
condition over counter
	}

	public static class incrementCounter extends Thread {
	    public void run() {
		counter++;
	    }
```

```
    }

    public static class decrementCounter extends Thread {
        public void run() {
           counter--;
        }
    }
```

Different Threads Increment and Decrement The Same Thread-Safe Counter Repeatedly, Never Resulting in a Race Condition

```
    public static int counter = 0;
    public static Object lock = new Object();

    public static void start() throws InterruptedException {
           incrementCounter ic;
           decrementCounter dc;
           while(counter == 0) { // because of proper locking, this condition is never false
                  counter = 0;
                  ic = new incrementCounter();
                  dc = new decrementCounter();
                  ic.start();
                  dc.start();
                  ic.join();
                  dc.join();
           }
           System.out.println(counter); // Never reached
    }

    public static class incrementCounter extends Thread {
        public void run() {
           synchronized (lock) {
                  counter++;
           }
        }
    }

    public static class decrementCounter extends Thread {
        public void run() {
           synchronized (lock) {
                  counter--;
           }
        }
    }
```

Use of sizeof() on a Pointer Type

Weakness ID: 467 *(Weakness Variant)* **Status:** Draft

Description

Description Summary

The code calls sizeof() on a malloced pointer type, which always returns the wordsize/8. This can produce an unexpected result if the programmer intended to determine how much memory has been allocated.

Time of Introduction

- Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

Scope	Effect
Integrity	This error can often cause one to allocate a buffer that is much smaller than what is needed, leading to resultant weaknesses such as buffer overflows.

Likelihood of Exploit

High

Demonstrative Examples

Example 1

Care should be taken to ensure sizeof returns the size of the data structure itself, and not the size of the pointer to the data structure.

In this example, sizeof(foo) returns the size of the pointer.

(Bad Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(foo));
```

In this example, sizeof(*foo) returns the size of the data structure and not the size of the pointer.

(Good Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(*foo));
```

Example 2

This example defines a fixed username and password. The AuthenticateUser() function is intended to accept a username and a password from an untrusted user, and check to ensure that it matches the username and password. If the username and password match, AuthenticateUser() is intended to indicate that authentication succeeded.

(Bad Code)

```
/* Ignore CWE-259 (hard-coded password) and CWE-309 (use of password system for authentication) for this example. */

char *username = "admin";
char *pass = "password";

int AuthenticateUser(char *inUser, char *inPass) {
```

```
printf("Sizeof username = %d\n", sizeof(username));
printf("Sizeof pass = %d\n", sizeof(pass));

if (strncmp(username, inUser, sizeof(username))) {
printf("Auth failure of username using sizeof\n");
return(AUTH_FAIL);
}
/* Because of CWE-467, the sizeof returns 4 on many platforms and architectures. */
if (! strncmp(pass, inPass, sizeof(pass))) {
printf("Auth success of password using sizeof\n");
return(AUTH_SUCCESS);
}
else {
printf("Auth fail of password using sizeof\n");
return(AUTH_FAIL);
}
}

int main (int argc, char **argv)
{
int authResult;

if (argc < 3) {
ExitError("Usage: Provide a username and password");
}
authResult = AuthenticateUser(argv[1], argv[2]);
if (authResult != AUTH_SUCCESS) {
ExitError("Authentication failed");
}
else {
DoAuthenticatedTask(argv[1]);
}
}
```

In AuthenticateUser(), because sizeof() is applied to a parameter with an array type, the sizeof() call might return 4 on many modern architectures. As a result, the strncmp() call only checks the first four characters of the input password, resulting in a partial comparison (CWE-187), leading to improper authentication (CWE-287).

Because of the partial comparison, any of these passwords would still cause authentication to succeed for the "admin" user:

(Attack)

```
pass5
passABCDEFGH
passWORD
```

Because only 4 characters are checked, this significantly reduces the search space for an attacker, making brute force attacks more feasible.

The same problem also applies to the username, so values such as "adminXYZ" and "administrator" will succeed for the username.

Potential Mitigations

Phase: Implementation

Use expressions such as "sizeof(*pointer)" instead of "sizeof(pointer)", unless you intend to run sizeof() on a pointer type to gain some platform independence or if you are allocating a variable on the stack.

Other Notes

The use of sizeof() on a pointer can sometimes generate useful information. An obvious case is to find out the wordsize on a platform. More often than not, the appearance of sizeof(pointer) indicates a bug.

Weakness Ordinalities

Ordinality	Description
Primary	*(where the weakness exists independent of other weaknesses)*

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	465	Pointer Issues	**Development Concepts (primary)699**
ChildOf	Weakness Class	682	Incorrect Calculation	**Research Concepts (primary)1000**
ChildOf	Category	737	CERT C Secure Coding Section 03 - Expressions (EXP)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
ChildOf	Category	740	CERT C Secure Coding Section 06 - Arrays (ARR)	Weaknesses Addressed by the CERT C Secure Coding Standard734
CanPrecede	Weakness Base	131	Incorrect Calculation of Buffer Size	Research Concepts1000

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Use of sizeof() on a pointer type
CERT C Secure Coding	ARR01-C		Do not apply the sizeof operator to a pointer when taking the size of an array
CERT C Secure Coding	EXP01-C		Do not take the size of a pointer to determine the size of the pointed-to type

White Box Definitions

A weakness where code path has:

1. end statement that passes an identity of a dynamically allocated memory resource to a sizeof operator
2. start statement that allocates the dynamically allocated memory resource

References

Robert Seacord. "EXP01-A. Do not take the sizeof a pointer to determine the size of a type". <https://www.securecoding.cert.org/confluence/display/seccode/EXP01-A.+Do+not+take+the+sizeof+a+pointer+to+determine+the+size+of+a+type>.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	CLASP		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		

BACK TO TOP

NULL Pointer Dereference

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
- Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
- Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.

Source Code Examples

Improper Validation of Array Index

Weakness ID: 129 *(Weakness Base)* **Status:** Draft

Description

Description Summary

The product uses untrusted input when calculating or using an array index, but the product does not validate or incorrectly validates the index to ensure the index references a valid position within the array.

Alternate Terms

out-of-bounds array index

index-out-of-range

array index underflow

Time of Introduction

- Implementation

Applicable Platforms

Languages

C: *(Often)*

C++: *(Often)*

Language-independent

Common Consequences

Scope	Effect
Integrity Availability	Unchecked array indexing will very likely result in the corruption of relevant memory and perhaps instructions, leading to a crash, if the values are outside of the valid memory area.
Integrity	If the memory corrupted is data, rather than instructions, the system will continue to function with improper values.
Confidentiality Integrity	Unchecked array indexing can also trigger out-of-bounds read or write operations, or operations on the wrong objects; i.e., "buffer overflows" are not always the result. This may result in the exposure or modification of sensitive data.
Integrity	If the memory accessible by the attacker can be effectively controlled, it may be possible to execute arbitrary code, as with a standard buffer overflow and possibly without the use of large inputs if a precise index can be controlled.
Integrity Availability Confidentiality	A single fault could allow either an overflow (CWE-788) or underflow (CWE-786) of the array index. What happens next will depend on the type of operation being performed out of bounds, but can expose sensitive information, cause a system crash, or possibly lead to arbitrary code execution.

Likelihood of Exploit

High

Detection Methods

Automated Static Analysis

This weakness can often be detected using automated static analysis tools. Many modern tools use data flow analysis or constraint-based techniques to minimize the number of false positives.

Automated static analysis generally does not account for environmental considerations when reporting out-of-bounds memory operations. This can make it difficult for users to determine which warnings should be investigated first. For example, an analysis tool might report array index errors that originate from command line arguments in a program that is not expected to run with setuid or other special privileges.

Effectiveness: High

This is not a perfect solution, since 100% accuracy and coverage are not feasible.

Automated Dynamic Analysis

This weakness can be detected using dynamic tools and techniques that interact with the software using large test suites with many diverse inputs, such as fuzz testing (fuzzing), robustness testing, and fault injection. The software's operation may slow down, but it should not become unstable, crash, or generate incorrect results.

Black Box

Black box methods might not get the needed code coverage within limited time constraints, and a dynamic test might not produce any noticeable side effects even if it is successful.

Demonstrative Examples

Example 1

The following C/C++ example retrieves the sizes of messages for a pop3 mail server. The message sizes are retrieved from a socket that returns in a buffer the message number and the message size, the message number (num) and size (size) are extracted from the buffer and the message size is placed into an array using the message number for the array index.

(Bad Code)

Example Language: **C**

```
/* capture the sizes of all messages */
int getsizes(int sock, int count, int *sizes) {
...
char buf[BUFFER_SIZE];
int ok;
int num, size;

// read values from socket and added to sizes array
while ((ok = gen_recv(sock, buf, sizeof(buf))) == 0)
{

// continue read from socket until buf only contains '.'
if (DOTLINE(buf))
break;
else if (sscanf(buf, "%d %d", &num, &size) == 2)
sizes[num - 1] = size;
}
...
}
```

In this example the message number retrieved from the buffer could be a value that is outside the allowable range of indices for the array and could possibly be a negative number. Without proper validation of the value to be used for the array index an array overflow could occur and could potentially lead to unauthorized access to memory addresses and system crashes. The value of the array index should be validated to ensure that it is within the allowable range of indices for the array as in the following code.

(Good Code)

Example Language: **C**

```
/* capture the sizes of all messages */
int getsizes(int sock, int count, int *sizes) {
...
char buf[BUFFER_SIZE];
int ok;
int num, size;

// read values from socket and added to sizes array
while ((ok = gen_recv(sock, buf, sizeof(buf))) == 0)
{

// continue read from socket until buf only contains '.'
if (DOTLINE(buf))
```

```
break;
else if (sscanf(buf, "%d %d", &num, &size) == 2) {
if (num > 0 && num <= (unsigned)count)
sizes[num - 1] = size;
else
/* warn about possible attempt to induce buffer overflow */
report(stderr, "Warning: ignoring bogus data for message sizes returned by server.\n");
}
}
...
}
```

Example 2

In the code snippet below, an unchecked integer value is used to reference an object in an array.

(Bad Code)

Example Language: **Java**

```
public String getValue(int index) {
return array[index];
}
```

If index is outside of the range of the array, this may result in an ArrayIndexOutOfBounds Exception being raised.

Example 3

In the following Java example the method displayProductSummary is called from a Web service servlet to retrieve product summary information for display to the user. The servlet obtains the integer value of the product number from the user and passes it to the displayProductSummary method. The displayProductSummary method passes the integer value of the product number to the getProductSummary method which obtains the product summary from the array object containing the project summaries using the integer value of the product number as the array index.

(Bad Code)

Example Language: **Java**

```
// Method called from servlet to obtain product information
public String displayProductSummary(int index) {

String productSummary = new String("");

try {
String productSummary = getProductSummary(index);

} catch (Exception ex) {...}

return productSummary;
}

public String getProductSummary(int index) {
return products[index];
}
```

In this example the integer value used as the array index that is provided by the user may be outside the allowable range of indices for the array which may provide unexpected results or may comes the application to fail. The integer value used for the array index should be validated to ensure that it is within the allowable range of indices for the array as in the following code.

(Good Code)

Example Language: **Java**

```
// Method called from servlet to obtain product information
public String displayProductSummary(int index) {

String productSummary = new String("");
```

```
try {
String productSummary = getProductSummary(index);

} catch (Exception ex) {...}

return productSummary;
}

public String getProductSummary(int index) {
String productSummary = "";

if ((index >= 0) && (index < MAX_PRODUCTS)) {
productSummary = products[index];
}
else {
System.err.println("index is out of bounds");
throw new IndexOutOfBoundsException();
}

return productSummary;
}
```

An alternative in Java would be to use one of the collection objects such as ArrayList that will automatically generate an exception if an attempt is made to access an array index that is out of bounds.

(Good Code)

Example Language: **Java**

```
ArrayList productArray = new ArrayList(MAX_PRODUCTS);
...
try {
productSummary = (String) productArray.get(index);
} catch (IndexOutOfBoundsException ex) {...}
```

Observed Examples

Reference	Description
CVE-2005-0369	large ID in packet used as array index
CVE-2001-1009	negative array index as argument to POP LIST command
CVE-2003-0721	Integer signedness error leads to negative array index
CVE-2004-1189	product does not properly track a count and a maximum number, which can lead to resultant array index overflow.
CVE-2007-5756	chain: device driver for packet-capturing software allows access to an unintended IOCTL with resultant array index error.

Potential Mitigations

Phase: Architecture and Design

Strategies: Input Validation; Libraries or Frameworks

Use an input validation framework such as Struts or the OWASP ESAPI Validation API. If you use Struts, be mindful of weaknesses covered by the CWE-101 category.

Phase: Architecture and Design

For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side, in order to avoid CWE-602. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the server.

Even though client-side checks provide minimal benefits with respect to server-side security, they are still useful. First, they can support intrusion detection. If the server receives input that should have been rejected by the client, then it may be an indication of an attack. Second, client-side error-checking can provide helpful feedback to the user about the expectations for valid input. Third, there may be a reduction in server-side processing time for accidental input errors, although this is typically a small savings.

Phase: Requirements

Strategy: Language Selection

Use a language with features that can automatically mitigate or eliminate out-of-bounds indexing errors.

For example, Ada allows the programmer to constrain the values of a variable and languages such as Java and Ruby will allow the programmer to handle exceptions when an out-of-bounds index is accessed.

Phase: Implementation

Strategy: Input Validation

Assume all input is malicious. Use an "accept known good" input validation strategy (i.e., use a whitelist). Reject any input that does not strictly conform to specifications, or transform it into something that does. Use a blacklist to reject any unexpected inputs and detect potential attacks.

When accessing a user-controlled array index, use a stringent range of values that are within the target array. Make sure that you do not allow negative values to be used. That is, verify the minimum as well as the maximum of the range of acceptable values.

Phase: Implementation

Be especially careful to validate your input when you invoke code that crosses language boundaries, such as from an interpreted language to native code. This could create an unexpected interaction between the language boundaries. Ensure that you are not violating any of the expectations of the language with which you are interfacing. For example, even though Java may not be susceptible to buffer overflows, providing a large argument in a call to native code might trigger an overflow.

Weakness Ordinalities

Ordinality	Description
Resultant	The most common condition situation leading to unchecked array indexing is the use of loop index variables as buffer indexes. If the end condition for the loop is subject to a flaw, the index can grow or shrink unbounded, therefore causing a buffer overflow or underflow. Another common situation leading to this condition is the use of a function's return value, or the resulting value of a calculation directly as an index in to a buffer.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	20	Improper Input Validation	**Development Concepts (primary)699 Research Concepts (primary)1000**
ChildOf	Category	189	Numeric Errors	Development Concepts699
ChildOf	Category	633	Weaknesses that Affect Memory	**Resource-specific Weaknesses (primary)631**
ChildOf	Category	738	CERT C Secure Coding Section 04 - Integers (INT)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
ChildOf	Category	740	CERT C Secure Coding Section 06 - Arrays (ARR)	Weaknesses Addressed by the CERT C Secure Coding Standard734
ChildOf	Category	802	2010 Top 25 - Risky Resource Management	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
CanPrecede	Weakness Class	119	Failure to Constrain Operations within the Bounds of a Memory Buffer	Research Concepts1000
CanPrecede	Weakness Variant	789	Uncontrolled Memory Allocation	Research Concepts1000
PeerOf	Weakness Base	124	Buffer Underwrite ('Buffer Underflow')	Research Concepts1000

Theoretical Notes

An improperly validated array index might lead directly to the always-incorrect behavior of "access of array using out-of-bounds index."

Affected Resources

- Memory

f Causal Nature

Explicit

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Unchecked array indexing
PLOVER			INDEX - Array index overflow
CERT C Secure Coding	ARR00-C		Understand how arrays work
CERT C Secure Coding	ARR30-C		Guarantee that array indices are within the valid range
CERT C Secure Coding	ARR38-C		Do not add or subtract an integer to a pointer if the resulting value does not refer to a valid array element
CERT C Secure Coding	INT32-C		Ensure that operations on signed integers do not result in overflow

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
100	Overflow Buffers	

References

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 5, "Array Indexing Errors" Page 144. 2nd Edition. Microsoft. 2002.

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	CLASP		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Sean Eidemiller	Cigital	External
	added/updated demonstrative examples		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Alternate Terms, Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Description, Name, Relationships		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Observed Examples, Other Notes, Potential Mitigations, Theoretical Notes, Weakness Ordinalities		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Demonstrative Examples, Detection Factors, Likelihood of Exploit, Potential Mitigations, References, Related Attack Patterns, Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Related Attack Patterns		
Previous Entry Names			
Change Date	**Previous Entry Name**		
2009-10-29	Unchecked Array Indexing		

BACK TO TOP

Scanned Languages

Language	Hash Number	Change Date
CPP	4541647240435660	1/6/2025
Common	0105849645654507	1/6/2025