wireshark Scan Report

Project Name	wireshark
Scan Start	Friday, June 21, 2024 6:10:37 PM
Preset	Checkmarx Default
Scan Time	01h:00m:07s
Lines Of Code Scanned	290233
Files Scanned	27
Report Creation Time	Friday, June 21, 2024 7:21:40 PM
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037
Team	CxServer
Checkmarx Version	8.7.0
Scan Type	Full
Source Origin	LocalPath
Density	5/10000 (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

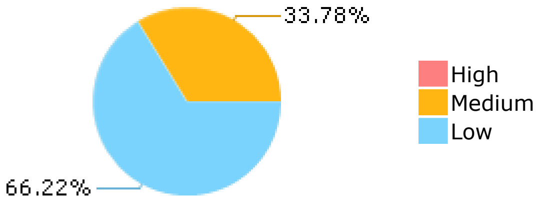
Results Limit

Results limit per query was set to 50

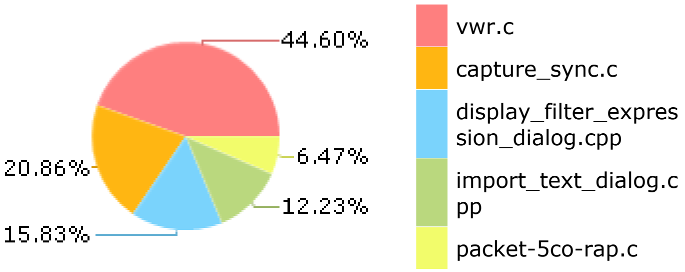
Selected Queries

Selected queries are listed in Result Summary

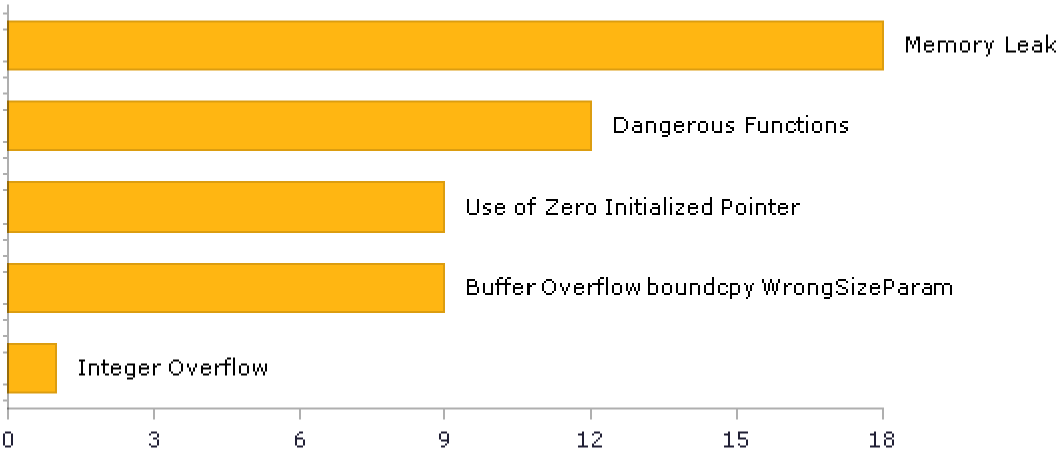
Result Summary



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	12	12
A2-Broken Authentication	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	3	3
A3-Sensitive Data Exposure	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	App. Specific	0	0
A4-XML External Entities (XXE)	App. Specific	AVERAGE	COMMON	EASY	SEVERE	App. Specific	0	0
A5-Broken Access Control*	App. Specific	AVERAGE	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A6-Security Misconfiguration	App. Specific	EASY	WIDESPREAD	EASY	MODERATE	App. Specific	0	0
A7-Cross-Site Scripting (XSS)	App. Specific	EASY	WIDESPREAD	EASY	MODERATE	App. Specific	0	0
A8-Insecure Deserialization	App. Specific	DIFFICULT	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A9-Using Components with Known Vulnerabilities*	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	MODERATE	App. Specific	12	12
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	12	12
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	10	10
PCI DSS (3.2) - 6.5.3 - Insecure cryptographic storage	0	0
PCI DSS (3.2) - 6.5.4 - Insecure communications	0	0
PCI DSS (3.2) - 6.5.5 - Improper error handling*	0	0
PCI DSS (3.2) - 6.5.7 - Cross-site scripting (XSS)	0	0
PCI DSS (3.2) - 6.5.8 - Improper access control	0	0
PCI DSS (3.2) - 6.5.9 - Cross-site request forgery	0	0
PCI DSS (3.2) - 6.5.10 - Broken authentication and session management	0	0

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

Scan Summary - FISMA 2014

Category	Description	Issues Found	Best Fix Locations
Access Control	Organizations must limit information system access to authorized users, processes acting on behalf of authorized users, or devices (including other information systems) and to the types of transactions and functions that authorized users are permitted to exercise.	3	3
Audit And Accountability*	Organizations must: (i) create, protect, and retain information system audit records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful, unauthorized, or inappropriate information system activity; and (ii) ensure that the actions of individual information system users can be uniquely traced to those users so they can be held accountable for their actions.	0	0
Configuration Management	Organizations must: (i) establish and maintain baseline configurations and inventories of organizational information systems (including hardware, software, firmware, and documentation) throughout the respective system development life cycles; and (ii) establish and enforce security configuration settings for information technology products employed in organizational information systems.	0	0
Identification And Authentication*	Organizations must identify information system users, processes acting on behalf of users, or devices and authenticate (or verify) the identities of those users, processes, or devices, as a prerequisite to allowing access to organizational information systems.	1	1
Media Protection	Organizations must: (i) protect information system media, both paper and digital; (ii) limit access to information on information system media to authorized users; and (iii) sanitize or destroy information system media before disposal or release for reuse.	0	0
System And Communications Protection	Organizations must: (i) monitor, control, and protect organizational communications (i.e., information transmitted or received by organizational information systems) at the external boundaries and key internal boundaries of the information systems; and (ii) employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational information systems.	0	0
System And Information Integrity	Organizations must: (i) identify, report, and correct information and information system flaws in a timely manner; (ii) provide protection from malicious code at appropriate locations within organizational information systems; and (iii) monitor information system security alerts and advisories and take appropriate actions in response.	1	1

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

Scan Summary - NIST SP 800-53

Category	Issues Found	Best Fix Locations
AC-12 Session Termination (P2)	0	0
AC-3 Access Enforcement (P1)	3	3
AC-4 Information Flow Enforcement (P1)	0	0
AC-6 Least Privilege (P1)	0	0
AU-9 Protection of Audit Information (P1)	0	0
CM-6 Configuration Settings (P2)	0	0
IA-5 Authenticator Management (P1)	0	0
IA-6 Authenticator Feedback (P2)	0	0
IA-8 Identification and Authentication (Non-Organizational Users) (P1)	0	0
SC-12 Cryptographic Key Establishment and Management (P1)	0	0
SC-13 Cryptographic Protection (P1)	0	0
SC-17 Public Key Infrastructure Certificates (P1)	0	0
SC-18 Mobile Code (P2)	0	0
SC-23 Session Authenticity (P1)*	0	0
SC-28 Protection of Information at Rest (P1)	1	1
SC-4 Information in Shared Resources (P1)	0	0
SC-5 Denial of Service Protection (P1)*	32	30
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*	52	52
SI-11 Error Handling (P2)*	29	29
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

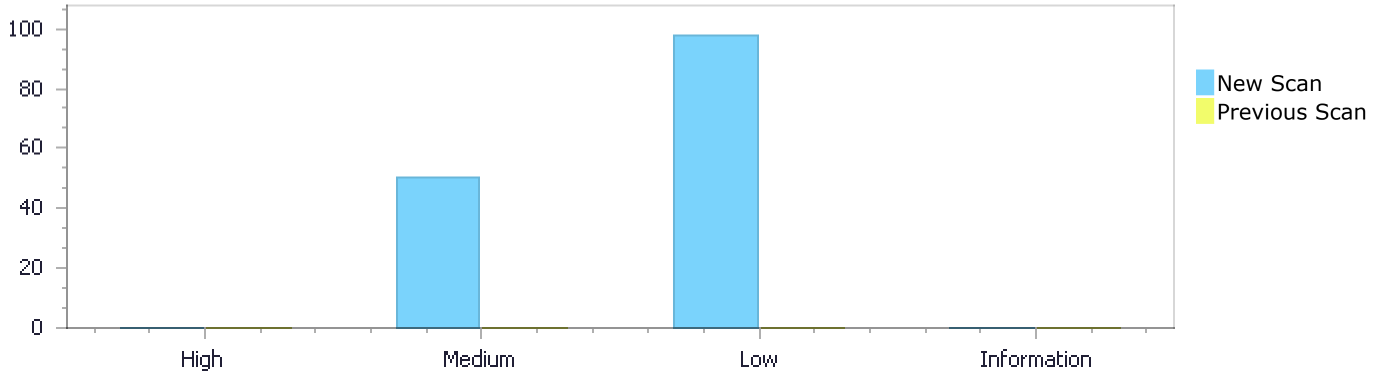
Category	Issues Found	Best Fix Locations
Must audit	0	0
Check	0	0
Optional	0	0

Results Distribution By Status

First scan of the project

	High	Medium	Low	Information	Total
New Issues	0	50	98	0	148
Recurrent Issues	0	0	0	0	0
Total	0	50	98	0	148

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	50	98	0	148
Urgent	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0
Total	0	50	98	0	148

Result Summary

Vulnerability Type	Occurrences	Severity
Memory Leak	18	Medium
Dangerous Functions	12	Medium
Buffer Overflow boundcpy WrongSizeParam	9	Medium
Use of Zero Initialized Pointer	9	Medium
Integer Overflow	1	Medium

Use After Free	1	Medium
Unchecked Array Index	51	Low
Unchecked Return Value	29	Low
Sizeof Pointer Argument	4	Low
Incorrect Permission Assignment For Critical Resources	3	Low
TOCTOU	3	Low
Use of Sizeof On a Pointer Type	3	Low
Improper Resource Shutdown or Release	2	Low
NULL Pointer Dereference	2	Low
Information Exposure Through Comments	1	Low

10 Most Vulnerable Files

High and Medium Vulnerabilities

File Name	Issues Found
wireshark-1/display_filter_expression_dialog.cpp	21
wireshark-1/vwr.c	11
wireshark-1/import_text_dialog.cpp	7
wireshark-1/wslua_struct.c	5
wireshark-1/capture_sync.c	4
wireshark-1/unicode-utils.c	2

Scan Results Details

Memory Leak

Query Path:

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

Categories

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

Description

Memory Leak\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=114
Status	New

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	153	153
Object	QListWidgetItem	QListWidgetItem

Code Snippet

File Name	wireshark-1/display_filter_expression_dialog.cpp
Method	DisplayFilterExpressionDialog::DisplayFilterExpressionDialog(QWidget *parent) :

```
....
153.      new QListWidgetItem("is present", ui->relationListWidget,
present_op_);
```

Memory Leak\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=115
Status	New

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	154	154
Object	QListWidgetItem	QListWidgetItem

Code Snippet

File Name	wireshark-1/display_filter_expression_dialog.cpp

Method DisplayFilterExpressionDialog::DisplayFilterExpressionDialog(QWidget *parent) :

```
....
154.        new QListWidgetItem("==", ui->relationListWidget, any_eq_op_);
```

Memory Leak\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=116
Status	New

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	155	155
Object	QListWidgetItem	QListWidgetItem

Code Snippet
File Name wireshark-1/display_filter_expression_dialog.cpp
Method DisplayFilterExpressionDialog::DisplayFilterExpressionDialog(QWidget *parent) :

```
....
155.        new QListWidgetItem("!=", ui->relationListWidget, all_ne_op_);
```

Memory Leak\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=117
Status	New

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	156	156
Object	QListWidgetItem	QListWidgetItem

Code Snippet
File Name wireshark-1/display_filter_expression_dialog.cpp
Method DisplayFilterExpressionDialog::DisplayFilterExpressionDialog(QWidget *parent) :

```
....
156.        new QListWidgetItem("===", ui->relationListWidget,
all_eq_op_);
```

Memory Leak\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=118
Status	New

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	157	157
Object	QListWidgetItem	QListWidgetItem

Code Snippet
File Name wireshark-1/display_filter_expression_dialog.cpp
Method DisplayFilterExpressionDialog::DisplayFilterExpressionDialog(QWidget *parent) :

```
....
157.        new QListWidgetItem("!==", ui->relationListWidget,
any_ne_op_);
```

Memory Leak\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=119
Status	New

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	158	158
Object	QListWidgetItem	QListWidgetItem

Code Snippet
File Name wireshark-1/display_filter_expression_dialog.cpp
Method DisplayFilterExpressionDialog::DisplayFilterExpressionDialog(QWidget *parent) :

```
....
158.        new QListWidgetItem(">", ui->relationListWidget, gt_op_);
```

Memory Leak\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=120

Status	New

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	159	159
Object	QListWidgetItem	QListWidgetItem

Code Snippet
File Name wireshark-1/display_filter_expression_dialog.cpp
Method DisplayFilterExpressionDialog::DisplayFilterExpressionDialog(QWidget *parent) :

```
....
159.        new QListWidgetItem("<", ui->relationListWidget, lt_op_);
```

Memory Leak\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=121
Status	New

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	160	160
Object	QListWidgetItem	QListWidgetItem

Code Snippet
File Name wireshark-1/display_filter_expression_dialog.cpp
Method DisplayFilterExpressionDialog::DisplayFilterExpressionDialog(QWidget *parent) :

```
....
160.        new QListWidgetItem(">=", ui->relationListWidget, ge_op_);
```

Memory Leak\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=122
Status	New

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp

Line	161	161
Object	QListWidgetItem	QListWidgetItem

Code Snippet
File Name wireshark-1/display_filter_expression_dialog.cpp
Method DisplayFilterExpressionDialog::DisplayFilterExpressionDialog(QWidget *parent) :

```
....
161.        new QListWidgetItem("<=", ui->relationListWidget, le_op_);
```

Memory Leak\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=123
Status	New

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	162	162
Object	QListWidgetItem	QListWidgetItem

Code Snippet
File Name wireshark-1/display_filter_expression_dialog.cpp
Method DisplayFilterExpressionDialog::DisplayFilterExpressionDialog(QWidget *parent) :

```
....
162.        new QListWidgetItem("contains", ui->relationListWidget,
contains_op_);
```

Memory Leak\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=124
Status	New

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	163	163
Object	QListWidgetItem	QListWidgetItem

Code Snippet
File Name wireshark-1/display_filter_expression_dialog.cpp

Method	DisplayFilterExpressionDialog::DisplayFilterExpressionDialog(QWidget *parent) :

```
....
163.        new QListWidgetItem("matches", ui->relationListWidget,
matches_op_);
```

Memory Leak\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=125
Status	New

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	164	164
Object	QListWidgetItem	QListWidgetItem

Code Snippet

File Name	wireshark-1/display_filter_expression_dialog.cpp
Method	DisplayFilterExpressionDialog::DisplayFilterExpressionDialog(QWidget *parent) :

```
....
164.        new QListWidgetItem("in", ui->relationListWidget, in_op_);
```

Memory Leak\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=126
Status	New

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	79	79
Object	proto_ti	proto_ti

Code Snippet

File Name	wireshark-1/display_filter_expression_dialog.cpp
Method	static void generateProtocolTreeItems(QPromise<QTreeWidgetItem *> &promise)

```
....
79.            QTreeWidgetItem *proto_ti = new
QTreeWidgetItem(proto_type_);
```

Memory Leak\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=127
Status	New

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	320	320
Object	eli	eli

Code Snippet

File Name wireshark-1/display_filter_expression_dialog.cpp

Method void DisplayFilterExpressionDialog::fillEnumBooleanValues(const true_false_string *tfs)

```
....
320.       QListWidgetItem *eli = new QListWidgetItem(tfs->true_string,
ui->enumListWidget);
```

Memory Leak\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=128
Status	New

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	358	358
Object	eli	eli

Code Snippet

File Name wireshark-1/display_filter_expression_dialog.cpp

Method void DisplayFilterExpressionDialog::fillEnumRangeValues(const _range_string *rvals)

```
....
358.          QListWidgetItem *eli = new QListWidgetItem(range_text, ui-
>enumListWidget);
```

Memory Leak\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=129
Status	New

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	126	126
Object	watcher	watcher

Code Snippet

File Name	wireshark-1/display_filter_expression_dialog.cpp
Method	DisplayFilterExpressionDialog::DisplayFilterExpressionDialog(QWidget *parent) :

```
....
126.      watcher(new QFutureWatcher<QTreeWidgetItem *>(nullptr)),
```

Memory Leak\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=130
Status	New

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	322	322
Object	eli	eli

Code Snippet

File Name	wireshark-1/display_filter_expression_dialog.cpp
Method	void DisplayFilterExpressionDialog::fillEnumBooleanValues(const true_false_string *tfs)

```
....
322.      eli = new QListWidgetItem(tfs->false_string, ui-
>enumListWidget);
```

Memory Leak\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=131
Status	New

	Source	Destination
File	wireshark-1/import_text_dialog.cpp	wireshark-1/import_text_dialog.cpp
Line	103	103
Object	encap_buttons	encap_buttons

Code Snippet
File Name wireshark-1/import_text_dialog.cpp
Method ImportTextDialog::ImportTextDialog(QWidget *parent) :

```
....
103.        encap_buttons = new QButtonGroup(this);
```

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-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=102
Status	New

The dangerous function, memcpy, was found in use at line 1579 in wireshark-1/capture_sync.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	1627	1627
Object	memcpy	memcpy

Code Snippet
File Name wireshark-1/capture_sync.c
Method pipe_read_block(GIOChannel *pipe_io, char *indicator, int len, char *msg,

```
....
1627.          memcpy(msg, header, sizeof(header));
```

Dangerous Functions\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=103
Status	New

The dangerous function, memcpy, was found in use at line 1144 in wireshark-1/vwr.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1415	1415
Object	memcpy	memcpy

Code Snippet	
File Name	wireshark-1/vwr.c
Method	static gboolean vwr_read_s1_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1415.       memcpy(&data_ptr[bytes_written], &rec[plcp_hdr_len],
actual_octets);
```

Dangerous Functions\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=104
Status	New

The dangerous function, memcpy, was found in use at line 1421 in wireshark-1/vwr.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1832	1832
Object	memcpy	memcpy

Code Snippet	
File Name	wireshark-1/vwr.c
Method	static gboolean vwr_read_s2_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1832.       memcpy(&data_ptr[bytes_written], &rec[vwr->MPDU_OFF],
actual_octets);
```

Dangerous Functions\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=105
Status	New

The dangerous function, memcpy, was found in use at line 1837 in wireshark-1/vwr.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2504	2504
Object	memcpy	memcpy

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2504.           memcpy(&data_ptr[bytes_written], &rec[stats_offset+16],
16);
```

Dangerous Functions\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=106
Status	New

The dangerous function, memcpy, was found in use at line 1837 in wireshark-1/vwr.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2546	2546
Object	memcpy	memcpy

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2546.          memcpy(&data_ptr[bytes_written], &rec[stats_offset+(vwr-
>MPDU_OFF)], actual_octets);
```

Dangerous Functions\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=107
Status	New

The dangerous function, memcpy, was found in use at line 2556 in wireshark-1/vwr.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2817	2817
Object	memcpy	memcpy

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_rec_data_ethernet(vwr_t *vwr, wtap_rec *record,

```
....
2817.      memcpy(&data_ptr[bytes_written], m_ptr, actual_octets);
```

Dangerous Functions\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=108
Status	New

The dangerous function, memcpy, was found in use at line 441 in wireshark-1/wslua_struct.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wireshark-1/wslua_struct.c	wireshark-1/wslua_struct.c
Line	490	490
Object	memcpy	memcpy

Code Snippet
File Name wireshark-1/wslua_struct.c
Method WSLUA_CONSTRUCTOR Struct_unpack (lua_State *L) {

```
....
490.            memcpy(&f, data+pos, size);
```

Dangerous Functions\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=109
Status	New

The dangerous function, memcpy, was found in use at line 441 in wireshark-1/wslua_struct.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wireshark-1/wslua_struct.c	wireshark-1/wslua_struct.c
Line	497	497
Object	memcpy	memcpy

Code Snippet

File Name	wireshark-1/wslua_struct.c
Method	WSLUA_CONSTRUCTOR Struct_unpack (lua_State *L) {

```
....
497.            memcpy(&d, data+pos, size);
```

Dangerous Functions\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=110
Status	New

The dangerous function, strlen, was found in use at line 611 in wireshark-1/capture_sync.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	783	783
Object	strlen	strlen

Code Snippet

File Name	wireshark-1/capture_sync.c
Method	sync_pipe_start(capture_options *capture_opts, GPtrArray *capture_comments,

```
....
783.            if (interface_opts->cfilter != NULL &&
strlen(interface_opts->cfilter) != 0) {
```

Dangerous Functions\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=111
Status	New

The dangerous function, strlen, was found in use at line 405 in wireshark-1/import_text_dialog.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wireshark-1/import_text_dialog.cpp	wireshark-1/import_text_dialog.cpp
Line	429	429
Object	strlen	strlen

Code Snippet

File Name	wireshark-1/import_text_dialog.cpp
Method	int ImportTextDialog::exec() {

```
....
429.        if (strlen(import_info_.timestamp_format) == 0) {
```

Dangerous Functions\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=112
Status	New

The dangerous function, MultiByteToWideChar, was found in use at line 252 in wireshark-1/unicode-utils.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wireshark-1/unicode-utils.c	wireshark-1/unicode-utils.c
Line	271	271
Object	MultiByteToWideChar	MultiByteToWideChar

Code Snippet

File Name	wireshark-1/unicode-utils.c
Method	utf_8to16(const char *utf8str)

```
....
271.        while (MultiByteToWideChar(CP_UTF8, 0, utf8str, -1, NULL, 0)
>= utf16buf_len[idx]) {
```

Dangerous Functions\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=113
Status	New

The dangerous function, MultiByteToWideChar, was found in use at line 252 in wireshark-1/unicode-utils.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wireshark-1/unicode-utils.c	wireshark-1/unicode-utils.c
Line	282	282
Object	MultiByteToWideChar	MultiByteToWideChar

Code Snippet

File Name	wireshark-1/unicode-utils.c
Method	utf_8to16(const char *utf8str)

```
....
282.        if (MultiByteToWideChar(CP_UTF8, 0, utf8str, -1,
utf16buf[idx], utf16buf_len[idx]) == 0)
```

Buffer Overflow boundcpy WrongSizeParam

Query Path:
CPP\Cx\CPP Buffer Overflow\Buffer Overflow boundcpy WrongSizeParam Version:1

Categories

PCI DSS v3.2: PCI DSS (3.2) - 6.5.2 - Buffer overflows
OWASP Top 10 2017: A1-Injection

Description

Buffer Overflow boundcpy WrongSizeParam\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=33
Status	New

The size of the buffer used by pipe_read_block in header, at line 1579 of wireshark-1/capture_sync.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that pipe_read_block passes to header, at line 1579 of wireshark-1/capture_sync.c, to overwrite the target buffer.

	Source	Destination

File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	1627	1627
Object	header	header

Code Snippet
File Name wireshark-1/capture_sync.c
Method pipe_read_block(GIOChannel *pipe_io, char *indicator, int len, char *msg,

```
....
1627.           memcpy(msg, header, sizeof(header));
```

Buffer Overflow boundcpy WrongSizeParam\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=34
Status	New

The size of the buffer used by ImportTextDialog::on_modeTabWidget_currentChanged in Namespace1485015897, at line 697 of wireshark-1/import_text_dialog.cpp, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ImportTextDialog::on_modeTabWidget_currentChanged passes to Namespace1485015897, at line 697 of wireshark-1/import_text_dialog.cpp, to overwrite the target buffer.

	Source	Destination
File	wireshark-1/import_text_dialog.cpp	wireshark-1/import_text_dialog.cpp
Line	704	704
Object	Namespace1485015897	Namespace1485015897

Code Snippet
File Name wireshark-1/import_text_dialog.cpp
Method void ImportTextDialog::on_modeTabWidget_currentChanged(int index) {

```
....
704.          memset(&import_info_.hexdump, 0, 
sizeof(import_info_.hexdump));
```

Buffer Overflow boundcpy WrongSizeParam\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=35
Status	New

The size of the buffer used by ImportTextDialog::on_modeTabWidget_currentChanged in Namespace1485015897, at line 697 of wireshark-1/import_text_dialog.cpp, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that

ImportTextDialog::on_modeTabWidget_currentChanged passes to Namespace1485015897, at line 697 of wireshark-1/import_text_dialog.cpp, to overwrite the target buffer.

	Source	Destination
File	wireshark-1/import_text_dialog.cpp	wireshark-1/import_text_dialog.cpp
Line	711	711
Object	Namespace1485015897	Namespace1485015897

Code Snippet
File Name wireshark-1/import_text_dialog.cpp
Method void ImportTextDialog::on_modeTabWidget_currentChanged(int index) {

```
....
711.            memset(&import_info_.regex, 0, 
sizeof(import_info_.regex));
```

Buffer Overflow boundcpy WrongSizeParam\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=36
Status	New

The size of the buffer used by vwr_read_s1_W_rec in actual_octets, at line 1144 of wireshark-1/vwr.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that vwr_read_s1_W_rec passes to actual_octets, at line 1144 of wireshark-1/vwr.c, to overwrite the target buffer.

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1415	1415
Object	actual_octets	actual_octets

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s1_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1415.       memcpy(&data_ptr[bytes_written], &rec[plcp_hdr_len], 
actual_octets);
```

Buffer Overflow boundcpy WrongSizeParam\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=37
Status	New

The size of the buffer used by vwr_read_s2_W_rec in actual_octets, at line 1421 of wireshark-1/vwr.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that vwr_read_s2_W_rec passes to actual_octets, at line 1421 of wireshark-1/vwr.c, to overwrite the target buffer.

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1832	1832
Object	actual_octets	actual_octets

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s2_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1832.       memcpy(&data_ptr[bytes_written], &rec[vwr->MPDU_OFF],
actual_octets);
```

Buffer Overflow boundcpy WrongSizeParam\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=38
Status	New

The size of the buffer used by vwr_read_s3_W_rec in actual_octets, at line 1837 of wireshark-1/vwr.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that vwr_read_s3_W_rec passes to actual_octets, at line 1837 of wireshark-1/vwr.c, to overwrite the target buffer.

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2546	2546
Object	actual_octets	actual_octets

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2546.           memcpy(&data_ptr[bytes_written], &rec[stats_offset+(vwr-
>MPDU_OFF)], actual_octets);
```

Buffer Overflow boundcpy WrongSizeParam\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=39
Status	New

The size of the buffer used by vwr_read_rec_data_ethernet in actual_octets, at line 2556 of wireshark-1/vwr.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that vwr_read_rec_data_ethernet passes to actual_octets, at line 2556 of wireshark-1/vwr.c, to overwrite the target buffer.

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2817	2817
Object	actual_octets	actual_octets

Code Snippet

File Name	wireshark-1/vwr.c
Method	static gboolean vwr_read_rec_data_ethernet(vwr_t *vwr, wtap_rec *record,

```
....
2817.      memcpy(&data_ptr[bytes_written], m_ptr, actual_octets);
```

Buffer Overflow boundcpy WrongSizeParam\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=40
Status	New

The size of the buffer used by Struct_unpack in size, at line 441 of wireshark-1/wslua_struct.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Struct_unpack passes to size, at line 441 of wireshark-1/wslua_struct.c, to overwrite the target buffer.

	Source	Destination
File	wireshark-1/wslua_struct.c	wireshark-1/wslua_struct.c
Line	490	490
Object	size	size

Code Snippet

File Name	wireshark-1/wslua_struct.c
Method	WSLUA_CONSTRUCTOR Struct_unpack (lua_State *L) {

```
....
490.          memcpy(&f, data+pos, size);
```

Buffer Overflow boundcpy WrongSizeParam\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=41
Status	New

The size of the buffer used by Struct_unpack in size, at line 441 of wireshark-1/wslua_struct.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that Struct_unpack passes to size, at line 441 of wireshark-1/wslua_struct.c, to overwrite the target buffer.

	Source	Destination
File	wireshark-1/wslua_struct.c	wireshark-1/wslua_struct.c
Line	497	497
Object	size	size

Code Snippet
File Name wireshark-1/wslua_struct.c
Method WSLUA_CONSTRUCTOR Struct_unpack (lua_State *L) {

```
....
497.           memcpy(&d, data+pos, size);
```

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-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=133
Status	New

The variable declared in s_start_ptr at wireshark-1/vwr.c in line 1837 is not initialized when it is used by s_start_ptr at wireshark-1/vwr.c in line 1837.

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1846	2487
Object	s_start_ptr	s_start_ptr

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1846.       const guint8      *s_start_ptr = NULL,*s_trail_ptr = NULL,
*plcp_ptr, *m_ptr; /* stats & MPDU ptr */
....
2487.           data_ptr[bytes_written] = s_start_ptr[2];    /*** For
Signal Bandwidth Mask ***/
```

Use of Zero Initialized Pointer\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=134
Status	New

The variable declared in s_start_ptr at wireshark-1/vwr.c in line 1837 is not initialized when it is used by s_start_ptr at wireshark-1/vwr.c in line 1837.

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1846	2489
Object	s_start_ptr	s_start_ptr

Code Snippet	
File Name	wireshark-1/vwr.c
Method	static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1846.       const guint8      *s_start_ptr = NULL,*s_trail_ptr = NULL,
*plcp_ptr, *m_ptr; /* stats & MPDU ptr */
....
2489.           data_ptr[bytes_written] = s_start_ptr[3];    /*** For
Antenna Port Energy Detect and MU_MASK ***/
```

Use of Zero Initialized Pointer\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=135
Status	New

The variable declared in Pointer at wireshark-1/capture_sync.c in line 224 is not initialized when it is used by args at wireshark-1/capture_sync.c in line 201.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	238	201
Object	Pointer	args

Code Snippet	
File Name	wireshark-1/capture_sync.c
Method	init_pipe_args(int *argc) {

```
....
238.        *argv = NULL;
```

▼

File Name wireshark-1/capture_sync.c

Method sync_pipe_add_arg(char **args, int *argc, const char *arg)

```
....
201.  sync_pipe_add_arg(char **args, int *argc, const char *arg)
```

Use of Zero Initialized Pointer\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=136
Status	New

The variable declared in field_ at wireshark-1/display_filter_expression_dialog.cpp in line 123 is not initialized when it is used by field_ at wireshark-1/display_filter_expression_dialog.cpp in line 229.

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	132	231
Object	field_	field_

Code Snippet

File Name wireshark-1/display_filter_expression_dialog.cpp

Method DisplayFilterExpressionDialog::DisplayFilterExpressionDialog(QWidget *parent) :

```
....
132.      field_(NULL)
```

▼

File Name wireshark-1/display_filter_expression_dialog.cpp

Method void DisplayFilterExpressionDialog::updateWidgets()

```
....
231.      bool rel_enable = field_ != NULL;
```

Use of Zero Initialized Pointer\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=137
Status	New

The variable declared in field_ at wireshark-1/display_filter_expression_dialog.cpp in line 363 is not initialized when it is used by field_ at wireshark-1/display_filter_expression_dialog.cpp in line 229.

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	366	231
Object	field_	field_

Code Snippet

File Name wireshark-1/display_filter_expression_dialog.cpp

Method void DisplayFilterExpressionDialog::on_fieldTreeWidget_itemSelectionChanged()

```
....
366.        field_ = NULL;
```

File Name wireshark-1/display_filter_expression_dialog.cpp

Method void DisplayFilterExpressionDialog::updateWidgets()

```
....
231.        bool rel_enable = field_ != NULL;
```

Use of Zero Initialized Pointer\Path 6:

Severity Medium

Result State To Verify

Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=138

Status New

The variable declared in field_ at wireshark-1/display_filter_expression_dialog.cpp in line 363 is not initialized when it is used by field_ at wireshark-1/display_filter_expression_dialog.cpp in line 363.

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	366	438
Object	field_	field_

Code Snippet

File Name wireshark-1/display_filter_expression_dialog.cpp

Method void DisplayFilterExpressionDialog::on_fieldTreeWidget_itemSelectionChanged()

```
....
366.        field_ = NULL;
....
438.        bool all_show = field_ != NULL;
```

Use of Zero Initialized Pointer\Path 7:

Severity Medium

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=139
Status	New

The variable declared in timestamp_format at wireshark-1/import_text_dialog.cpp in line 405 is not initialized when it is used by timestamp_format at wireshark-1/import_text_dialog.cpp in line 405.

	Source	Destination
File	wireshark-1/import_text_dialog.cpp	wireshark-1/import_text_dialog.cpp
Line	431	559
Object	timestamp_format	timestamp_format

Code Snippet

File Name	wireshark-1/import_text_dialog.cpp
Method	int ImportTextDialog::exec() {

```
....
431.            import_info_.timestamp_format = NULL;
....
559.        g_free((gpointer) import_info_.timestamp_format);
```

Use of Zero Initialized Pointer\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=140
Status	New

The variable declared in in_indication at wireshark-1/import_text_dialog.cpp in line 405 is not initialized when it is used by in_indication at wireshark-1/import_text_dialog.cpp in line 405.

	Source	Destination
File	wireshark-1/import_text_dialog.cpp	wireshark-1/import_text_dialog.cpp
Line	467	553
Object	in_indication	in_indication

Code Snippet

File Name	wireshark-1/import_text_dialog.cpp
Method	int ImportTextDialog::exec() {

```
....
467.                import_info_.regex.in_indication = NULL;
....
553.            g_free((gpointer) import_info_.regex.in_indication);
```

Use of Zero Initialized Pointer\Path 9:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=141
Status	New

The variable declared in out_indication at wireshark-1/import_text_dialog.cpp in line 405 is not initialized when it is used by out_indication at wireshark-1/import_text_dialog.cpp in line 405.

	Source	Destination
File	wireshark-1/import_text_dialog.cpp	wireshark-1/import_text_dialog.cpp
Line	468	554
Object	out_indication	out_indication

Code Snippet
File Name wireshark-1/import_text_dialog.cpp
Method int ImportTextDialog::exec() {

```
....
468.              import_info_.regex.out_indication = NULL;
....
554.          g_free((gpointer) import_info_.regex.out_indication);
```

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-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=44
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 328 of wireshark-1/wslua_struct.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	wireshark-1/wslua_struct.c	wireshark-1/wslua_struct.c
Line	396	396
Object	AssignExpr	AssignExpr

Code Snippet
File Name wireshark-1/wslua_struct.c

Method WSLUA_CONSTRUCTOR Struct_pack (lua_State *L) {

```
....
396.            posBuf[poscnt++] = (int)totalsize + 1;
```

Use After Free

Query Path:
CPP\Cx\CPP Medium Threat\Use After Free Version:1

Categories

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

Description

Use After Free\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=132
Status	New

The pointer proto_ti at wireshark-1/display_filter_expression_dialog.cpp in line 63 is being used after it has been freed.

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	92	98
Object	proto_ti	proto_ti

Code Snippet
File Name wireshark-1/display_filter_expression_dialog.cpp
Method static void generateProtocolTreeItems(QPromise<QTreeWidgetItem *> &promise)

```
....
92.                 delete proto_ti;
....
98.             int proto_id = proto_ti->data(0, Qt::UserRole).toInt();
```

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-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=51
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1363	1363
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s1_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1363.       data_ptr[bytes_written] = vVW510021_W_PLCP_LEGACY; /* pre-HT
*/
```

Unchecked Array Index\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=52
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1365	1365
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s1_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1365.       data_ptr[bytes_written] = rate_index;
```

Unchecked Array Index\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=53
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c

Line	1367	1367
Object	bytes_written	bytes_written

Code Snippet

File Name wireshark-1/vwr.c

Method static gboolean vwr_read_s1_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1367.      data_ptr[bytes_written] = 1; /* pre-VHT, so NSS = 1 */
```

Unchecked Array Index\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=54
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1369	1369
Object	bytes_written	bytes_written

Code Snippet

File Name wireshark-1/vwr.c

Method static gboolean vwr_read_s1_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1369.      data_ptr[bytes_written] = rssi;
```

Unchecked Array Index\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=55
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1372	1372
Object	bytes_written	bytes_written

Code Snippet

File Name wireshark-1/vwr.c

Method static gboolean vwr_read_s1_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1372.       data_ptr[bytes_written] = 100;
```

Unchecked Array Index\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=56
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1374	1374
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s1_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1374.       data_ptr[bytes_written] = 100;
```

Unchecked Array Index\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=57
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1376	1376
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s1_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1376.       data_ptr[bytes_written] = 100;
```

Unchecked Array Index\Path 8:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=58
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1379	1379
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s1_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1379.       data_ptr[bytes_written] = 0;
```

Unchecked Array Index\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=59
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1781	1781
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s2_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1781.       data_ptr[bytes_written] = plcp_type;
```

Unchecked Array Index\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=60
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1784	1784

Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s2_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1784.      data_ptr[bytes_written] = rate_mcs_index;
```

Unchecked Array Index\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=61
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1787	1787
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s2_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1787.      data_ptr[bytes_written] = nss;
```

Unchecked Array Index\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=62
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1789	1789
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s2_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1789.       data_ptr[bytes_written] = rssi[0];
```

Unchecked Array Index\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=63
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1791	1791
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s2_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1791.       data_ptr[bytes_written] = rssi[1];
```

Unchecked Array Index\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=64
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1793	1793
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s2_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1793.       data_ptr[bytes_written] = rssi[2];
```

Unchecked Array Index\Path 15:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=65
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1795	1795
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s2_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1795.       data_ptr[bytes_written] = rssi[3];
```

Unchecked Array Index\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=66
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	1798	1798
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s2_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
1798.       data_ptr[bytes_written] = 0;
```

Unchecked Array Index\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=67
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2279	2279

Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2279.           data_ptr[bytes_written] = 0;
```

Unchecked Array Index\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=68
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2281	2281
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2281.           data_ptr[bytes_written] = 0;
```

Unchecked Array Index\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=69
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2283	2283
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2283.           data_ptr[bytes_written] = 0;
```

Unchecked Array Index\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=70
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2293	2293
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2293.                     data_ptr[bytes_written] =
rf_ptr[RF_PORT_1_NOISE_OFF+i*RF_INTER_PORT_GAP_OFF];
```

Unchecked Array Index\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=71
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2295	2295
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2295.                     data_ptr[bytes_written] =
rf_ptr[RF_PORT_1_NOISE_OFF+1+i*RF_INTER_PORT_GAP_OFF];
```

Unchecked Array Index\Path 22:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=72
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2307	2307
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2307.                       data_ptr[bytes_written] =
rf_ptr[RF_PORT_1_SNR_OFF+i*RF_INTER_PORT_GAP_OFF];
```

Unchecked Array Index\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=73
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2309	2309
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2309.                       data_ptr[bytes_written] =
rf_ptr[RF_PORT_1_SNR_OFF+1+i*RF_INTER_PORT_GAP_OFF];
```

Unchecked Array Index\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=74
Status	New

	Source	Destination

File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2321	2321
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2321.                    data_ptr[bytes_written] =
rf_ptr[RF_PORT_1_PFE_OFF+i*RF_INTER_PORT_GAP_OFF];
```

Unchecked Array Index\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=75
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2323	2323
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2323.                    data_ptr[bytes_written] =
rf_ptr[RF_PORT_1_PFE_OFF+1+i*RF_INTER_PORT_GAP_OFF];
```

Unchecked Array Index\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=76
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2335	2335
Object	bytes_written	bytes_written

Code Snippet

File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2335.                        data_ptr[bytes_written] =
rf_ptr[RF_PORT_1_EVM_SD_SIG_OFF+i*RF_INTER_PORT_GAP_OFF];
```

Unchecked Array Index\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=77
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2337	2337
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2337.                        data_ptr[bytes_written] =
rf_ptr[RF_PORT_1_EVM_SD_SIG_OFF+1+i*RF_INTER_PORT_GAP_OFF];
```

Unchecked Array Index\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=78
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2349	2349
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2349.                        data_ptr[bytes_written] =
rf_ptr[RF_PORT_1_EVM_SP_SIG_OFF+i*RF_INTER_PORT_GAP_OFF];
```

Unchecked Array Index\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=79
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2351	2351
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2351.                   data_ptr[bytes_written] =
rf_ptr[RF_PORT_1_EVM_SP_SIG_OFF+1+i*RF_INTER_PORT_GAP_OFF];
```

Unchecked Array Index\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=80
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2363	2363
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2363.                   data_ptr[bytes_written] =
rf_ptr[RF_PORT_1_EVM_SD_DATA_OFF+i*RF_INTER_PORT_GAP_OFF];
```

Unchecked Array Index\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=81

Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2365	2365
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2365.                      data_ptr[bytes_written] =
rf_ptr[RF_PORT_1_EVM_SD_DATA_OFF+1+i*RF_INTER_PORT_GAP_OFF];
```

Unchecked Array Index\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=82
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2377	2377
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2377.                      data_ptr[bytes_written] =
rf_ptr[RF_PORT_1_EVM_SP_DATA_OFF+i*RF_INTER_PORT_GAP_OFF];
```

Unchecked Array Index\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=83
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2379	2379

Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2379.                     data_ptr[bytes_written] =
rf_ptr[RF_PORT_1_EVM_SP_DATA_OFF+1+i*RF_INTER_PORT_GAP_OFF];
```

Unchecked Array Index\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=84
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2391	2391
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2391.                     data_ptr[bytes_written] =
rf_ptr[RF_PORT_1_DSYMBOL_IDX_OFF+i*RF_INTER_PORT_GAP_OFF];
```

Unchecked Array Index\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=85
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2393	2393
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2393.                       data_ptr[bytes_written] =
rf_ptr[RF_PORT_1_DSYMBOL_IDX_OFF+1+i*RF_INTER_PORT_GAP_OFF];
```

Unchecked Array Index\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=86
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2405	2405
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2405.                       data_ptr[bytes_written] =
rf_ptr[RF_PORT_1_CONTEXT_OFF+i*RF_INTER_PORT_GAP_OFF];
```

Unchecked Array Index\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=87
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2407	2407
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2407.                       data_ptr[bytes_written] =
rf_ptr[RF_PORT_1_CONTEXT_OFF+1+i*RF_INTER_PORT_GAP_OFF];
```

Unchecked Array Index\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=88
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2464	2464
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2464.           data_ptr[bytes_written] = l1p_1;
```

Unchecked Array Index\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=89
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2467	2467
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2467.           data_ptr[bytes_written] = (nss << 4) | IS_TX;
```

Unchecked Array Index\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=90
Status	New

	Source	Destination

File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2473	2473
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2473.           data_ptr[bytes_written] = l1p_2;
```

Unchecked Array Index\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=91
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2476	2476
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2476.           data_ptr[bytes_written] = rssi[0];
```

Unchecked Array Index\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=92
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2478	2478
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2478.           data_ptr[bytes_written] = rssi[1];
```

Unchecked Array Index\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=93
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2480	2480
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2480.           data_ptr[bytes_written] = rssi[2];
```

Unchecked Array Index\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=94
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2482	2482
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2482.           data_ptr[bytes_written] = rssi[3];
```

Unchecked Array Index\Path 45:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=95
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2487	2487
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2487.           data_ptr[bytes_written] = s_start_ptr[2];    /*** For
Signal Bandwidth Mask ***/
```

Unchecked Array Index\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=96
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2489	2489
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2489.           data_ptr[bytes_written] = s_start_ptr[3];    /*** For
Antenna Port Energy Detect and MU_MASK ***/
```

Unchecked Array Index\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=97
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c

Line	2493	2493
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2493.                  data_ptr[bytes_written] = L1InfoC;   /*** For Other
plcp type = VHT ***/
```

Unchecked Array Index\Path 48:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=98
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2495	2495
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2495.                  data_ptr[bytes_written] = 0;     /*** For Other plcp
type, this offset is set to 0***/
```

Unchecked Array Index\Path 49:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=99
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2514	2514
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2514.          data_ptr[bytes_written] = flow_seq;
```

Unchecked Array Index\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=100
Status	New

	Source	Destination
File	wireshark-1/vwr.c	wireshark-1/vwr.c
Line	2527	2527
Object	bytes_written	bytes_written

Code Snippet
File Name wireshark-1/vwr.c
Method static gboolean vwr_read_s3_W_rec(vwr_t *vwr, wtap_rec *record,

```
....
2527.              data_ptr[bytes_written] = info_2nd;
```

Unchecked Return Value

Query Path:
CPP\Cx\CPP Low Visibility\Unchecked Return Value Version:1

Categories

NIST SP 800-53: SI-11 Error Handling (P2)

Description

Unchecked Return Value\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=1
Status	New

The sync_pipe_open_command method calls the snprintf function, at line 277 of wireshark-1/capture_sync.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	389	389
Object	snprintf	snprintf

Code Snippet
File Name wireshark-1/capture_sync.c
Method sync_pipe_open_command(char* const argv[], int *data_read_fd,

```
....
389.            snprintf(control_id, ARGV_NUMBER_LEN, "%ld",
GetCurrentProcessId());
```

Unchecked Return Value\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=2
Status	New

The sync_pipe_start method calls the snprintf function, at line 611 of wireshark-1/capture_sync.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	670	670
Object	snprintf	snprintf

Code Snippet
File Name wireshark-1/capture_sync.c
Method sync_pipe_start(capture_options *capture_opts, GPtrArray *capture_comments,

```
....
670.                snprintf(sfilesize, ARGV_NUMBER_LEN,
"filesize:%u",capture_opts->autostop_filesize);
```

Unchecked Return Value\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=3
Status	New

The sync_pipe_start method calls the snprintf function, at line 611 of wireshark-1/capture_sync.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	677	677
Object	snprintf	snprintf

Code Snippet

File Name	wireshark-1/capture_sync.c
Method	sync_pipe_start(capture_options *capture_opts, GPtrArray *capture_comments,

```
....
677.                snprintf(sfile_duration, ARGV_NUMBER_LEN,
"duration:%f",capture_opts->file_duration);
```

Unchecked Return Value\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=4
Status	New

The sync_pipe_start method calls the snprintf function, at line 611 of wireshark-1/capture_sync.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	684	684
Object	snprintf	snprintf

Code Snippet

File Name	wireshark-1/capture_sync.c
Method	sync_pipe_start(capture_options *capture_opts, GPtrArray *capture_comments,

```
....
684.                snprintf(sfile_interval, ARGV_NUMBER_LEN,
"interval:%d",capture_opts->file_interval);
```

Unchecked Return Value\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=5
Status	New

The sync_pipe_start method calls the snprintf function, at line 611 of wireshark-1/capture_sync.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	691	691

Object	snprintf	snprintf

Code Snippet
File Name wireshark-1/capture_sync.c
Method sync_pipe_start(capture_options *capture_opts, GPtrArray *capture_comments,

```
....
691.                snprintf(sfile_packets, ARGV_NUMBER_LEN,
"packets:%d",capture_opts->file_packets);
```

Unchecked Return Value\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=6
Status	New

The sync_pipe_start method calls the snprintf function, at line 611 of wireshark-1/capture_sync.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	698	698
Object	snprintf	snprintf

Code Snippet
File Name wireshark-1/capture_sync.c
Method sync_pipe_start(capture_options *capture_opts, GPtrArray *capture_comments,

```
....
698.                snprintf(sring_num_files, ARGV_NUMBER_LEN,
"files:%d",capture_opts->ring_num_files);
```

Unchecked Return Value\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=7
Status	New

The sync_pipe_start method calls the snprintf function, at line 611 of wireshark-1/capture_sync.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c

Line	705	705
Object	snprintf	snprintf

Code Snippet
File Name   wireshark-1/capture_sync.c
Method   sync_pipe_start(capture_options *capture_opts, GPtrArray *capture_comments,

```
....
705.                snprintf(nametimenum, ARGV_NUMBER_LEN,
"nametimenum:2");
```

Unchecked Return Value\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=8
Status	New

The sync_pipe_start method calls the snprintf function, at line 611 of wireshark-1/capture_sync.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	712	712
Object	snprintf	snprintf

Code Snippet
File Name   wireshark-1/capture_sync.c
Method   sync_pipe_start(capture_options *capture_opts, GPtrArray *capture_comments,

```
....
712.                snprintf(sautostop_files, ARGV_NUMBER_LEN,
"files:%d",capture_opts->autostop_files);
```

Unchecked Return Value\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=9
Status	New

The sync_pipe_start method calls the snprintf function, at line 611 of wireshark-1/capture_sync.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination

File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	719	719
Object	snprintf	snprintf

Code Snippet
File Name wireshark-1/capture_sync.c
Method sync_pipe_start(capture_options *capture_opts, GPtrArray *capture_comments,

```
....
719.                snprintf(sautostop_filesize, ARGV_NUMBER_LEN,
"filesize:%u",capture_opts->autostop_filesize);
```

Unchecked Return Value\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=10
Status	New

The sync_pipe_start method calls the snprintf function, at line 611 of wireshark-1/capture_sync.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	727	727
Object	snprintf	snprintf

Code Snippet
File Name wireshark-1/capture_sync.c
Method sync_pipe_start(capture_options *capture_opts, GPtrArray *capture_comments,

```
....
727.            snprintf(scount, ARGV_NUMBER_LEN, "%d",capture_opts-
>autostop_packets);
```

Unchecked Return Value\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=11
Status	New

The sync_pipe_start method calls the snprintf function, at line 611 of wireshark-1/capture_sync.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	734	734
Object	snprintf	snprintf

Code Snippet

File Name	wireshark-1/capture_sync.c
Method	sync_pipe_start(capture_options *capture_opts, GPtrArray *capture_comments,

```
....
734.            snprintf(sautostop_duration, ARGV_NUMBER_LEN,
"duration:%f",capture_opts->autostop_duration);
```

Unchecked Return Value\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=12
Status	New

The sync_pipe_start method calls the snprintf function, at line 611 of wireshark-1/capture_sync.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	741	741
Object	snprintf	snprintf

Code Snippet

File Name	wireshark-1/capture_sync.c
Method	sync_pipe_start(capture_options *capture_opts, GPtrArray *capture_comments,

```
....
741.            snprintf(scount, ARGV_NUMBER_LEN,
"packets:%d",capture_opts->autostop_written_packets);
```

Unchecked Return Value\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=13
Status	New

The sync_pipe_start method calls the snprintf function, at line 611 of wireshark-1/capture_sync.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	752	752
Object	snprintf	snprintf

Code Snippet
File Name wireshark-1/capture_sync.c
Method sync_pipe_start(capture_options *capture_opts, GPtrArray *capture_comments,

```
....
752.            snprintf(scount, ARGV_NUMBER_LEN, "%d", capture_opts->update_interval);
```

Unchecked Return Value\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=14
Status	New

The sync_pipe_start method calls the snprintf function, at line 611 of wireshark-1/capture_sync.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	790	790
Object	snprintf	snprintf

Code Snippet
File Name wireshark-1/capture_sync.c
Method sync_pipe_start(capture_options *capture_opts, GPtrArray *capture_comments,

```
....
790.                snprintf(ssnap, ARGV_NUMBER_LEN, "%d", interface_opts->snaplen);
```

Unchecked Return Value\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=15
Status	New

The sync_pipe_start method calls the snprintf function, at line 611 of wireshark-1/capture_sync.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	813	813
Object	snprintf	snprintf

Code Snippet
File Name wireshark-1/capture_sync.c
Method sync_pipe_start(capture_options *capture_opts, GPtrArray *capture_comments,

```
....
813.              snprintf(buffer_size, ARGV_NUMBER_LEN, "%d",
interface_opts->buffer_size);
```

Unchecked Return Value\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=16
Status	New

The sync_pipe_start method calls the snprintf function, at line 611 of wireshark-1/capture_sync.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	834	834
Object	snprintf	snprintf

Code Snippet
File Name wireshark-1/capture_sync.c
Method sync_pipe_start(capture_options *capture_opts, GPtrArray *capture_comments,

```
....
834.              snprintf(sauth, sizeof(sauth), "%s:%s",
```

Unchecked Return Value\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=17
Status	New

The sync_pipe_start method calls the snprintf function, at line 611 of wireshark-1/capture_sync.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	845	845
Object	snprintf	snprintf

Code Snippet
File Name	wireshark-1/capture_sync.c
Method	sync_pipe_start(capture_options *capture_opts, GPtrArray *capture_comments,

```
....
845.                snprintf(ssampling, ARGV_NUMBER_LEN, "%s:%d",
```

Unchecked Return Value\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=18
Status	New

The sync_pipe_start method calls the snprintf function, at line 611 of wireshark-1/capture_sync.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	863	863
Object	snprintf	snprintf

Code Snippet
File Name	wireshark-1/capture_sync.c
Method	sync_pipe_start(capture_options *capture_opts, GPtrArray *capture_comments,

```
....
863.        snprintf(control_id, ARGV_NUMBER_LEN, "%ld",
GetCurrentProcessId());
```

Unchecked Return Value\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=19
Status	New

The sync_pipe_signame method calls the snprintf function, at line 1922 of wireshark-1/capture_sync.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	2007	2007
Object	snprintf	snprintf

Code Snippet
File Name	wireshark-1/capture_sync.c
Method	sync_pipe_signame(int sig)

```
....
2007.           snprintf(sigmsg_buf, sizeof sigmsg_buf, "Signal %d",
sig);
```

Unchecked Return Value\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=20
Status	New

The disp_timeout method calls the snprintf function, at line 945 of wireshark-1/packet-5co-rap.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/packet-5co-rap.c	wireshark-1/packet-5co-rap.c
Line	948	948
Object	snprintf	snprintf

Code Snippet
File Name	wireshark-1/packet-5co-rap.c
Method	static void disp_timeout(gchar *result, guint32 timeout)

```
....
948.            snprintf( result, 12, "%u%s",
```

Unchecked Return Value\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=21
Status	New

The disp_timeout method calls the snprintf function, at line 945 of wireshark-1/packet-5co-rap.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/packet-5co-rap.c	wireshark-1/packet-5co-rap.c
Line	951	951
Object	snprintf	snprintf

Code Snippet
File Name wireshark-1/packet-5co-rap.c
Method static void disp_timeout(gchar *result, guint32 timeout)

```
....
951.            snprintf( result, 8, "Disabled");
```

Unchecked Return Value\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=22
Status	New

The disp_type method calls the snprintf function, at line 890 of wireshark-1/packet-5co-rap.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/packet-5co-rap.c	wireshark-1/packet-5co-rap.c
Line	894	894
Object	snprintf	snprintf

Code Snippet
File Name wireshark-1/packet-5co-rap.c
Method disp_type(gchar *result, guint32 type)

```
....
894.        snprintf( result, 18, "%u.%u (%.4X.%.4X)", nValueH, nValueL,
nValueH, nValueL);
```

Unchecked Return Value\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=23
Status	New

The disp_version method calls the snprintf function, at line 898 of wireshark-1/packet-5co-rap.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/packet-5co-rap.c	wireshark-1/packet-5co-rap.c
Line	904	904
Object	snprintf	snprintf

Code Snippet
File Name wireshark-1/packet-5co-rap.c
Method disp_version(gchar *result, guint32 version)

```
....
904.            snprintf( result, 11, "FW: %u.%u", nValueH, nValueL);
```

Unchecked Return Value\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=24
Status	New

The disp_version method calls the snprintf function, at line 898 of wireshark-1/packet-5co-rap.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/packet-5co-rap.c	wireshark-1/packet-5co-rap.c
Line	912	912
Object	snprintf	snprintf

Code Snippet
File Name wireshark-1/packet-5co-rap.c
Method disp_version(gchar *result, guint32 version)

```
....
912.            snprintf( result, 25, "HW: %u.%u / FW: %u.%u", nHWHigh,
nHWLow, nFWHigh, nFWLow);
```

Unchecked Return Value\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=25
Status	New

The disp_voltage method calls the snprintf function, at line 916 of wireshark-1/packet-5co-rap.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/packet-5co-rap.c	wireshark-1/packet-5co-rap.c
Line	920	920
Object	snprintf	snprintf

Code Snippet

File Name	wireshark-1/packet-5co-rap.c
Method	static void disp_voltage(gchar *result, guint32 voltage)

```
....
920.       snprintf( result, 11, "%u.%u V", nValueH, nValueL);
```

Unchecked Return Value\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=26
Status	New

The disp_mac method calls the snprintf function, at line 923 of wireshark-1/packet-5co-rap.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/packet-5co-rap.c	wireshark-1/packet-5co-rap.c
Line	927	927
Object	snprintf	snprintf

Code Snippet

File Name	wireshark-1/packet-5co-rap.c
Method	static void disp_mac(gchar *result, guint64 mac)

```
....
927.       snprintf( result, 18, "%.2X-%.2X-%.2X-%.2X-%.2X-%.2X",
pData[5], pData[4], pData[3], pData[2],
```

Unchecked Return Value\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=27
Status	New

The disp_ip method calls the snprintf function, at line 931 of wireshark-1/packet-5co-rap.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/packet-5co-rap.c	wireshark-1/packet-5co-rap.c
Line	935	935
Object	snprintf	snprintf

Code Snippet
File Name wireshark-1/packet-5co-rap.c
Method static void disp_ip(gchar *result, guint32 ip)

```
....
935.       snprintf( result, 15, "%u.%u.%u.%u", pData[3], pData[2],
pData[1], pData[0]);
```

Unchecked Return Value\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=28
Status	New

The disp_mask method calls the snprintf function, at line 938 of wireshark-1/packet-5co-rap.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/packet-5co-rap.c	wireshark-1/packet-5co-rap.c
Line	942	942
Object	snprintf	snprintf

Code Snippet
File Name wireshark-1/packet-5co-rap.c
Method static void disp_mask(gchar *result, guint32 mask)

```
....
942.       snprintf( result, 15, "%u.%u.%u.%u", pData[3], pData[2],
pData[1], pData[0]);
```

Unchecked Return Value\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=29
Status	New

The DisplayFilterExpressionDialog::DisplayFilterExpressionDialog method calls the width function, at line 123 of wireshark-1/display_filter_expression_dialog.cpp. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	wireshark-1/display_filter_expression_dialog.cpp	wireshark-1/display_filter_expression_dialog.cpp
Line	135	135
Object	width	width

Code Snippet
File Name wireshark-1/display_filter_expression_dialog.cpp
Method DisplayFilterExpressionDialog::DisplayFilterExpressionDialog(QWidget *parent) :

```
....
135.      if (parent) loadGeometry(parent->width() * 2 / 3, parent->height());
```

Sizeof Pointer Argument

Query Path:
CPP\Cx\CPP Low Visibility\Sizeof Pointer Argument Version:0

Description

Sizeof Pointer Argument\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=47
Status	New

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	1627	1627
Object	header	sizeof

Code Snippet
File Name wireshark-1/capture_sync.c
Method pipe_read_block(GIOChannel *pipe_io, char *indicator, int len, char *msg,

```
....
1627.         memcpy(msg, header, sizeof(header));
```

Sizeof Pointer Argument\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=48
Status	New

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c

Line	1628	1628
Object	header	sizeof

Code Snippet
File Name wireshark-1/capture_sync.c
Method pipe_read_block(GIOChannel *pipe_io, char *indicator, int len, char *msg,

```
....
1628.           g_io_channel_read_chars(pipe_io, &msg[sizeof(header)],
len-sizeof(header), &bytes_read, &err);
```

Sizeof Pointer Argument\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=49
Status	New

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	1628	1628
Object	header	sizeof

Code Snippet
File Name wireshark-1/capture_sync.c
Method pipe_read_block(GIOChannel *pipe_io, char *indicator, int len, char *msg,

```
....
1628.           g_io_channel_read_chars(pipe_io, &msg[sizeof(header)],
len-sizeof(header), &bytes_read, &err);
```

Sizeof Pointer Argument\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=50
Status	New

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	1628	1628
Object	header	sizeof

Code Snippet
File Name wireshark-1/capture_sync.c
Method pipe_read_block(GIOChannel *pipe_io, char *indicator, int len, char *msg,

```
....
1628.           g_io_channel_read_chars(pipe_io, &msg[sizeof(header)],
len-sizeof(header), &bytes_read, &err);
```

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-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=30
Status	New

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	207	207
Object	sizeof	sizeof

Code Snippet
File Name wireshark-1/capture_sync.c
Method sync_pipe_add_arg(char **args, int *argc, const char *arg)

```
....
207.      args = (char **)g_realloc( (gpointer) args, (*argc + 2) *
sizeof (char *));
```

Use of Sizeof On a Pointer Type\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=31
Status	New

	Source	Destination
File	wireshark-1/capture_sync.c	wireshark-1/capture_sync.c
Line	237	237
Object	sizeof	sizeof

Code Snippet
File Name wireshark-1/capture_sync.c
Method init_pipe_args(int *argc) {

```
....
237.        argv = (char **)g_malloc(sizeof (char *));
```

Use of Sizeof On a Pointer Type\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=32
Status	New

	Source	Destination
File	wireshark-1/unicode-utils.c	wireshark-1/unicode-utils.c
Line	347	347
Object	sizeof	sizeof

Code Snippet
File Name wireshark-1/unicode-utils.c
Method arg_list_utf_16to8(int argc, wchar_t *wc_argv[]) {

```
....
347.        argv = (char **)g_malloc((argc + 1) * sizeof(char *));
```

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-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=142
Status	New

	Source	Destination
File	wireshark-1/import_text_dialog.cpp	wireshark-1/import_text_dialog.cpp
Line	195	195
Object	open	open

Code Snippet
File Name wireshark-1/import_text_dialog.cpp

Method	void ImportTextDialog::loadSettingsFile()

```
....
195.        if (loadFile.open(QIODevice::ReadOnly)) {
```

Incorrect Permission Assignment For Critical Resources\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=143
Status	New

	Source	Destination
File	wireshark-1/import_text_dialog.cpp	wireshark-1/import_text_dialog.cpp
Line	212	212
Object	open	open

Code Snippet

File Name	wireshark-1/import_text_dialog.cpp
Method	void ImportTextDialog::saveSettingsFile()

```
....
212.        if (saveFile.open(QIODevice::WriteOnly)) {
```

Incorrect Permission Assignment For Critical Resources\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=144
Status	New

	Source	Destination
File	wireshark-1/import_text_dialog.cpp	wireshark-1/import_text_dialog.cpp
Line	597	597
Object	open	open

Code Snippet

File Name	wireshark-1/import_text_dialog.cpp
Method	void ImportTextDialog::on_textFileLineEdit_textChanged(const QString &file_name)

```
....
597.        if (file_name.length() > 0 &&
text_file.open(QIODevice::ReadOnly)) {
```

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-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=145
Status	New

The ImportTextDialog::loadSettingsFile method in wireshark-1/import_text_dialog.cpp file utilizes open 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	wireshark-1/import_text_dialog.cpp	wireshark-1/import_text_dialog.cpp
Line	195	195
Object	open	open

Code Snippet
File Name wireshark-1/import_text_dialog.cpp
Method void ImportTextDialog::loadSettingsFile()

```
....
195.        if (loadFile.open(QIODevice::ReadOnly)) {
```

TOCTOU\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=146
Status	New

The ImportTextDialog::saveSettingsFile method in wireshark-1/import_text_dialog.cpp file utilizes open 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	wireshark-1/import_text_dialog.cpp	wireshark-1/import_text_dialog.cpp
Line	212	212
Object	open	open

Code Snippet
File Name wireshark-1/import_text_dialog.cpp
Method void ImportTextDialog::saveSettingsFile()

```
....
212.        if (saveFile.open(QIODevice::WriteOnly)) {
```

TOCTOU\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=147
Status	New

The ImportTextDialog::on_textFileLineEdit_textChanged method in wireshark-1/import_text_dialog.cpp file utilizes open 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	wireshark-1/import_text_dialog.cpp	wireshark-1/import_text_dialog.cpp
Line	597	597
Object	open	open

Code Snippet

File Name	wireshark-1/import_text_dialog.cpp
Method	void ImportTextDialog::on_textFileLineEdit_textChanged(const QString &file_name)

```
....
597.        if (file_name.length() > 0 &&
text_file.open(QIODevice::ReadOnly)) {
```

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-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=42
Status	New

The variable declared in null at wireshark-1/import_text_dialog.cpp in line 746 is not initialized when it is used by QString at wireshark-1/import_text_dialog.cpp in line 746.

	Source	Destination

File	wireshark-1/import_text_dialog.cpp	wireshark-1/import_text_dialog.cpp
Line	749	757
Object	null	QString

Code Snippet
File Name wireshark-1/import_text_dialog.cpp
Method void ImportTextDialog::on_regexTextEdit_textChanged()

```
....
749.       GError* gerror = NULL;
....
757.           ti_ui_->regexHintLabel->setText(QString(gerror-
>message).toHtmlEscaped());
```

NULL Pointer Dereference\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=43
Status	New

The variable declared in 0 at wireshark-1/import_text_dialog.cpp in line 952 is not initialized when it is used by import_info_ at wireshark-1/import_text_dialog.cpp in line 952.

	Source	Destination
File	wireshark-1/import_text_dialog.cpp	wireshark-1/import_text_dialog.cpp
Line	954	954
Object	0	import_info_

Code Snippet
File Name wireshark-1/import_text_dialog.cpp
Method void ImportTextDialog::on_ipVersionComboBox_currentIndexChanged(int index)

```
....
954.       import_info_.ipv6 = (index == 1) ? 1 : 0;
```

Improper Resource Shutdown or Release

Query Path:
CPP\Cx\CPP Low Visibility\Improper Resource Shutdown or Release Version:0

Categories

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

Description

Improper Resource Shutdown or Release\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=300

	37&pathid=45
Status	New

The application's ImportTextDialog::loadSettingsFile method in wireshark-1/import_text_dialog.cpp defines and initializes the open object at 186. This object encapsulates a limited computing resource, such as open file streams, database connections, or network streams. This resource is not properly closed and released in all situations.

	Source	Destination
File	wireshark-1/import_text_dialog.cpp	wireshark-1/import_text_dialog.cpp
Line	195	195
Object	open	open

Code Snippet
File Name wireshark-1/import_text_dialog.cpp
Method void ImportTextDialog::loadSettingsFile()

```
....
195.        if (loadFile.open(QIODevice::ReadOnly)) {
```

Improper Resource Shutdown or Release\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=46
Status	New

The application's ImportTextDialog::saveSettingsFile method in wireshark-1/import_text_dialog.cpp defines and initializes the open object at 203. This object encapsulates a limited computing resource, such as open file streams, database connections, or network streams. This resource is not properly closed and released in all situations.

	Source	Destination
File	wireshark-1/import_text_dialog.cpp	wireshark-1/import_text_dialog.cpp
Line	212	212
Object	open	open

Code Snippet
File Name wireshark-1/import_text_dialog.cpp
Method void ImportTextDialog::saveSettingsFile()

```
....
212.        if (saveFile.open(QIODevice::WriteOnly)) {
```

Information Exposure Through Comments

Query Path:
CPP\Cx\CPP Low Visibility\Information Exposure Through Comments Version:1

Categories

FISMA 2014: Identification And Authentication
NIST SP 800-53: SC-28 Protection of Information at Rest (P1)

Description

Information Exposure Through Comments\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030044&projectid=30037&pathid=148
Status	New

	Source	Destination
File	wireshark-1/packet-ntlmssp.c	wireshark-1/packet-ntlmssp.c
Line	521	521
Object	password (1	password (1

Code Snippet
File Name wireshark-1/packet-ntlmssp.c
Method /* Unable to calculate the session key without a valid password (128 chars or less) ......*/

```
....
521.      /* Unable to calculate the session key without a valid
password (128 chars or less) ......*/
```

Buffer Overflow boundcpy WrongSizeParam

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.
Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

CPP

Overflowing Buffers

```
const int BUFFER_SIZE = 10;
char buffer[BUFFER_SIZE];

void copyStringToBuffer(char* inputString)
{
    strcpy(buffer, inputString);
}
```

Checked Buffers

```
const int BUFFER_SIZE = 10;
const int MAX_INPUT_SIZE = 256;
char buffer[BUFFER_SIZE];

void copyStringToBuffer(char* inputString)
{
    if (strnlen(inputString, MAX_INPUT_SIZE) < sizeof(buffer))
    {
        strncpy(buffer, inputString, sizeof(buffer));
    }
}
```

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 After Free

Risk

What might happen

A use after free error will cause code to use an area of memory previously assigned with a specific value, which has since been freed and may have been overwritten by another value. This error will likely cause unexpected behavior, memory corruption and crash errors. In some cases where the freed and used section of memory is used to determine execution flow, and the error can be induced by an attacker, this may result in execution of malicious code.

Cause

How does it happen

Pointers to variables allow code to have an address with a set size to a dynamically allocated variable. Eventually, the pointer's destination may become free - either explicitly in code, such as when programmatically freeing this variable, or implicitly, such as when a local variable is returned - once it is returned, the variable's scope is released. Once freed, this memory will be re-used by the application, overwritten with new data. At this point, dereferencing this pointer will potentially resolve newly written and unexpected data.

General Recommendations

How to avoid it

- Do not return local variables or pointers
- Review code to ensure no flow allows use of a pointer after it has been explicitly freed

Source Code Examples

CPP

Use of Variable after It was Freed

```
free(input);
printf("%s", input);
```

Use of Pointer to Local Variable That Was Freed On Return

```
int* func1()
{
      int i;
      i = 1;
      return &i;
}

void func2()

{

      int j;
      j = 5;
}

//..
      int * i = func1();
      printf("%d\r\n", *i); // Output could be 1 or Segmentation Fault
      func2();
      printf("%d\r\n", *i); // Output is 5, which is j's value, as func2() overwrote data in
the stack
//..
```

Use of Zero Initialized Pointer

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
- Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
- Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.

Source Code Examples

CPP

Explicit NULL Dereference

```
char * input = NULL;
printf("%s", input);
```

Implicit NULL Dereference

```
char * input;
printf("%s", input);
```

Java

Explicit Null Dereference

```
Object o = null;
out.println(o.getClass());
```

Unchecked Return Value

Risk

What might happen

A program that does not check function return values could cause the application to enter an undefined state. This could lead to unexpected behavior and unintended consequences, including inconsistent data, system crashes or other error-based exploits.

Cause

How does it happen

The application calls a system function, but does not receive or check the result of this funciton. These functions often return error codes in the result, or share other status codes with it's caller. The application simply ignores this result value, losing this vital information.

General Recommendations

How to avoid it

- Always check the result of any called function that returns a value, and verify the result is an expected value.

- Ensure the calling function responds to all possible return values.

- Expect runtime errors and handle them gracefully. Explicitly define a mechanism for handling unexpected errors.

Source Code Examples

CPP

Unchecked Memory Allocation

```
buff = (char*) malloc(size);
strncpy(buff, source, size);
```

Safer Memory Allocation

```
buff = (char*) malloc(size+1);
if (buff==NULL) exit(1);

strncpy(buff, source, size);
buff[size] = '\0';
```

Use of sizeof() on a Pointer Type

Weakness ID: 467 *(Weakness Variant)* **Status:** Draft

Description

Description Summary

The code calls sizeof() on a malloced pointer type, which always returns the wordsize/8. This can produce an unexpected result if the programmer intended to determine how much memory has been allocated.

Time of Introduction

- Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

Scope	Effect
Integrity	This error can often cause one to allocate a buffer that is much smaller than what is needed, leading to resultant weaknesses such as buffer overflows.

Likelihood of Exploit

High

Demonstrative Examples

Example 1

Care should be taken to ensure sizeof returns the size of the data structure itself, and not the size of the pointer to the data structure.

In this example, sizeof(foo) returns the size of the pointer.

(Bad Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(foo));
```

In this example, sizeof(*foo) returns the size of the data structure and not the size of the pointer.

(Good Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(*foo));
```

Example 2

This example defines a fixed username and password. The AuthenticateUser() function is intended to accept a username and a password from an untrusted user, and check to ensure that it matches the username and password. If the username and password match, AuthenticateUser() is intended to indicate that authentication succeeded.

(Bad Code)

```
/* Ignore CWE-259 (hard-coded password) and CWE-309 (use of password system for authentication) for this example. */

char *username = "admin";
char *pass = "password";

int AuthenticateUser(char *inUser, char *inPass) {
```

```
printf("Sizeof username = %d\n", sizeof(username));
printf("Sizeof pass = %d\n", sizeof(pass));

if (strncmp(username, inUser, sizeof(username))) {
printf("Auth failure of username using sizeof\n");
return(AUTH_FAIL);
}
/* Because of CWE-467, the sizeof returns 4 on many platforms and architectures. */
if (! strncmp(pass, inPass, sizeof(pass))) {
printf("Auth success of password using sizeof\n");
return(AUTH_SUCCESS);
}
else {
printf("Auth fail of password using sizeof\n");
return(AUTH_FAIL);
}
}

int main (int argc, char **argv)
{
int authResult;

if (argc < 3) {
ExitError("Usage: Provide a username and password");
}
authResult = AuthenticateUser(argv[1], argv[2]);
if (authResult != AUTH_SUCCESS) {
ExitError("Authentication failed");
}
else {
DoAuthenticatedTask(argv[1]);
}
}
```

In AuthenticateUser(), because sizeof() is applied to a parameter with an array type, the sizeof() call might return 4 on many modern architectures. As a result, the strncmp() call only checks the first four characters of the input password, resulting in a partial comparison (CWE-187), leading to improper authentication (CWE-287).

Because of the partial comparison, any of these passwords would still cause authentication to succeed for the "admin" user:

(Attack)

```
pass5
passABCDEFGH
passWORD
```

Because only 4 characters are checked, this significantly reduces the search space for an attacker, making brute force attacks more feasible.

The same problem also applies to the username, so values such as "adminXYZ" and "administrator" will succeed for the username.

Potential Mitigations

Phase: Implementation

Use expressions such as "sizeof(*pointer)" instead of "sizeof(pointer)", unless you intend to run sizeof() on a pointer type to gain some platform independence or if you are allocating a variable on the stack.

Other Notes

The use of sizeof() on a pointer can sometimes generate useful information. An obvious case is to find out the wordsize on a platform. More often than not, the appearance of sizeof(pointer) indicates a bug.

Weakness Ordinalities

Ordinality	Description
Primary	*(where the weakness exists independent of other weaknesses)*

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	465	Pointer Issues	**Development Concepts (primary)699**
ChildOf	Weakness Class	682	Incorrect Calculation	**Research Concepts (primary)1000**
ChildOf	Category	737	CERT C Secure Coding Section 03 - Expressions (EXP)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
ChildOf	Category	740	CERT C Secure Coding Section 06 - Arrays (ARR)	Weaknesses Addressed by the CERT C Secure Coding Standard734
CanPrecede	Weakness Base	131	Incorrect Calculation of Buffer Size	Research Concepts1000

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Use of sizeof() on a pointer type
CERT C Secure Coding	ARR01-C		Do not apply the sizeof operator to a pointer when taking the size of an array
CERT C Secure Coding	EXP01-C		Do not take the size of a pointer to determine the size of the pointed-to type

White Box Definitions

A weakness where code path has:

1. end statement that passes an identity of a dynamically allocated memory resource to a sizeof operator
2. start statement that allocates the dynamically allocated memory resource

References

Robert Seacord. "EXP01-A. Do not take the sizeof a pointer to determine the size of a type". <https://www.securecoding.cert.org/confluence/display/seccode/EXP01-A.+Do+not+take+the+sizeof+a+pointer+to+determine+the+size+of+a+type>.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	CLASP		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		

BACK TO TOP

NULL Pointer Dereference

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
- Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
- Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.

Source Code Examples

Improper Resource Shutdown or Release

Risk

What might happen

Unreleased resources can cause a drain of those available for system use, eventually causing general reliability and availability problems, such as performance degradation, process bloat, and system instability. If a resource leak can be intentionally exploited by an attacker, it may be possible to cause a widespread DoS (Denial of Service) attack. This might even expose sensitive information between unprivileged users, if the resource continues to retain data or user id between subsequent allocations.

Cause

How does it happen

The application code allocates resource objects, but does not ensure these are always closed and released in a timely manner. This can include database connections, file handles, network sockets, or any other resource that needs to be released. In some cases, these might be released - but only if everything works as planned; if there is any runtime exception during the normal course of system operations, resources start to leak.

Note that even in managed-memory languages such as Java, these resources must be explicitly released. Many types of resource are not released even when the Garbage Collector runs; and even if the the object would eventually release the resource, we have no control over when the Garbage Collector does run.

General Recommendations

How to avoid it

- Always close and release all resources.
- Ensure resources are released (along with any other necessary cleanup) in a `finally { }` block. Do not close resources in a `catch { }` block, since this is not ensured to be called.
- Explicitly call .close() on any instance of a class that implements the `Closable` or `AutoClosable` interfaces.
- Alternatively, an even better solution is to use the try-with-resources idiom, in order to automatically close any defined `AutoClosable` instances.

Source Code Examples

Java

Unreleased Database Connection

```
private MyObject getDataFromDb(int id)  {
      MyObject data = null;
      Connection con = null;
      try {
            Connection con = DriverManager.getConnection(CONN_STRING);
            data = queryDb(con, id);
      }
      catch ( SQLException e ) {
            handleError(e);
      }
}
```

Explicit Release of Database Connection

```
private MyObject getDataFromDb(int id)  {
      MyObject data = null;
      Connection con = null;
      try {
            Connection con = DriverManager.getConnection(CONN_STRING);
            data = queryDb(con, id);
      }
      catch ( SQLException e ) {
            handleError(e);
      }
      finally {
            if ((con != null) && (!con.isClosed())) {
            con.close();
        }
    }
}
```

Automatic Implicit Release Using Try-With-Resources

```
private MyObject getDataFromDb(int id)  {
      MyObject data = null;
      Connection con = null;
      try (Connection con = DriverManager.getConnection(CONN_STRING)) {
            data = queryDb(con, id);
      }
      catch ( SQLException e ) {
            handleError(e);
      }
}
```

Use of sizeof() on a Pointer Type

Weakness ID: 467 *(Weakness Variant)* **Status:** Draft

Description

Description Summary

The code calls sizeof() on a malloced pointer type, which always returns the wordsize/8. This can produce an unexpected result if the programmer intended to determine how much memory has been allocated.

Time of Introduction

- Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

Scope	Effect
Integrity	This error can often cause one to allocate a buffer that is much smaller than what is needed, leading to resultant weaknesses such as buffer overflows.

Likelihood of Exploit

High

Demonstrative Examples

Example 1

Care should be taken to ensure sizeof returns the size of the data structure itself, and not the size of the pointer to the data structure.

In this example, sizeof(foo) returns the size of the pointer.

(Bad Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(foo));
```

In this example, sizeof(*foo) returns the size of the data structure and not the size of the pointer.

(Good Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(*foo));
```

Example 2

This example defines a fixed username and password. The AuthenticateUser() function is intended to accept a username and a password from an untrusted user, and check to ensure that it matches the username and password. If the username and password match, AuthenticateUser() is intended to indicate that authentication succeeded.

(Bad Code)

```
/* Ignore CWE-259 (hard-coded password) and CWE-309 (use of password system for authentication) for this example. */

char *username = "admin";
char *pass = "password";

int AuthenticateUser(char *inUser, char *inPass) {
```

```
printf("Sizeof username = %d\n", sizeof(username));
printf("Sizeof pass = %d\n", sizeof(pass));

if (strncmp(username, inUser, sizeof(username))) {
printf("Auth failure of username using sizeof\n");
return(AUTH_FAIL);
}
/* Because of CWE-467, the sizeof returns 4 on many platforms and architectures. */
if (! strncmp(pass, inPass, sizeof(pass))) {
printf("Auth success of password using sizeof\n");
return(AUTH_SUCCESS);
}
else {
printf("Auth fail of password using sizeof\n");
return(AUTH_FAIL);
}
}

int main (int argc, char **argv)
{
int authResult;

if (argc < 3) {
ExitError("Usage: Provide a username and password");
}
authResult = AuthenticateUser(argv[1], argv[2]);
if (authResult != AUTH_SUCCESS) {
ExitError("Authentication failed");
}
else {
DoAuthenticatedTask(argv[1]);
}
}
```

In AuthenticateUser(), because sizeof() is applied to a parameter with an array type, the sizeof() call might return 4 on many modern architectures. As a result, the strncmp() call only checks the first four characters of the input password, resulting in a partial comparison (CWE-187), leading to improper authentication (CWE-287).

Because of the partial comparison, any of these passwords would still cause authentication to succeed for the "admin" user:

(Attack)

```
pass5
passABCDEFGH
passWORD
```

Because only 4 characters are checked, this significantly reduces the search space for an attacker, making brute force attacks more feasible.

The same problem also applies to the username, so values such as "adminXYZ" and "administrator" will succeed for the username.

Potential Mitigations

Phase: Implementation

Use expressions such as "sizeof(*pointer)" instead of "sizeof(pointer)", unless you intend to run sizeof() on a pointer type to gain some platform independence or if you are allocating a variable on the stack.

Other Notes

The use of sizeof() on a pointer can sometimes generate useful information. An obvious case is to find out the wordsize on a platform. More often than not, the appearance of sizeof(pointer) indicates a bug.

Weakness Ordinalities

Ordinality	Description
Primary	*(where the weakness exists independent of other weaknesses)*

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	465	Pointer Issues	**Development Concepts (primary)699**
ChildOf	Weakness Class	682	Incorrect Calculation	**Research Concepts (primary)1000**
ChildOf	Category	737	CERT C Secure Coding Section 03 - Expressions (EXP)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
ChildOf	Category	740	CERT C Secure Coding Section 06 - Arrays (ARR)	Weaknesses Addressed by the CERT C Secure Coding Standard734
CanPrecede	Weakness Base	131	Incorrect Calculation of Buffer Size	Research Concepts1000

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Use of sizeof() on a pointer type
CERT C Secure Coding	ARR01-C		Do not apply the sizeof operator to a pointer when taking the size of an array
CERT C Secure Coding	EXP01-C		Do not take the size of a pointer to determine the size of the pointed-to type

White Box Definitions

A weakness where code path has:

1. end statement that passes an identity of a dynamically allocated memory resource to a sizeof operator
2. start statement that allocates the dynamically allocated memory resource

References

Robert Seacord. "EXP01-A. Do not take the sizeof a pointer to determine the size of a type". <https://www.securecoding.cert.org/confluence/display/seccode/EXP01-A.+Do+not+take+the+sizeof+a+pointer+to+determine+the+size+of+a+type>.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	CLASP		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		

BACK TO TOP

Improper Validation of Array Index

Weakness ID: 129 *(Weakness Base)* **Status:** Draft

Description

Description Summary

The product uses untrusted input when calculating or using an array index, but the product does not validate or incorrectly validates the index to ensure the index references a valid position within the array.

Alternate Terms

out-of-bounds array index

index-out-of-range

array index underflow

Time of Introduction

- Implementation

Applicable Platforms

Languages

C: *(Often)*

C++: *(Often)*

Language-independent

Common Consequences

Scope	Effect
Integrity Availability	Unchecked array indexing will very likely result in the corruption of relevant memory and perhaps instructions, leading to a crash, if the values are outside of the valid memory area.
Integrity	If the memory corrupted is data, rather than instructions, the system will continue to function with improper values.
Confidentiality Integrity	Unchecked array indexing can also trigger out-of-bounds read or write operations, or operations on the wrong objects; i.e., "buffer overflows" are not always the result. This may result in the exposure or modification of sensitive data.
Integrity	If the memory accessible by the attacker can be effectively controlled, it may be possible to execute arbitrary code, as with a standard buffer overflow and possibly without the use of large inputs if a precise index can be controlled.
Integrity Availability Confidentiality	A single fault could allow either an overflow (CWE-788) or underflow (CWE-786) of the array index. What happens next will depend on the type of operation being performed out of bounds, but can expose sensitive information, cause a system crash, or possibly lead to arbitrary code execution.

Likelihood of Exploit

High

Detection Methods

Automated Static Analysis

This weakness can often be detected using automated static analysis tools. Many modern tools use data flow analysis or constraint-based techniques to minimize the number of false positives.

Automated static analysis generally does not account for environmental considerations when reporting out-of-bounds memory operations. This can make it difficult for users to determine which warnings should be investigated first. For example, an analysis tool might report array index errors that originate from command line arguments in a program that is not expected to run with setuid or other special privileges.

Effectiveness: High

This is not a perfect solution, since 100% accuracy and coverage are not feasible.

Automated Dynamic Analysis

This weakness can be detected using dynamic tools and techniques that interact with the software using large test suites with many diverse inputs, such as fuzz testing (fuzzing), robustness testing, and fault injection. The software's operation may slow down, but it should not become unstable, crash, or generate incorrect results.

Black Box

Black box methods might not get the needed code coverage within limited time constraints, and a dynamic test might not produce any noticeable side effects even if it is successful.

Demonstrative Examples

Example 1

The following C/C++ example retrieves the sizes of messages for a pop3 mail server. The message sizes are retrieved from a socket that returns in a buffer the message number and the message size, the message number (num) and size (size) are extracted from the buffer and the message size is placed into an array using the message number for the array index.

(Bad Code)

Example Language: **C**

```
/* capture the sizes of all messages */
int getsizes(int sock, int count, int *sizes) {
...
char buf[BUFFER_SIZE];
int ok;
int num, size;

// read values from socket and added to sizes array
while ((ok = gen_recv(sock, buf, sizeof(buf))) == 0)
{

// continue read from socket until buf only contains '.'
if (DOTLINE(buf))
break;
else if (sscanf(buf, "%d %d", &num, &size) == 2)
sizes[num - 1] = size;
}
...
}
```

In this example the message number retrieved from the buffer could be a value that is outside the allowable range of indices for the array and could possibly be a negative number. Without proper validation of the value to be used for the array index an array overflow could occur and could potentially lead to unauthorized access to memory addresses and system crashes. The value of the array index should be validated to ensure that it is within the allowable range of indices for the array as in the following code.

(Good Code)

Example Language: **C**

```
/* capture the sizes of all messages */
int getsizes(int sock, int count, int *sizes) {
...
char buf[BUFFER_SIZE];
int ok;
int num, size;

// read values from socket and added to sizes array
while ((ok = gen_recv(sock, buf, sizeof(buf))) == 0)
{

// continue read from socket until buf only contains '.'
if (DOTLINE(buf))
break;
else if (sscanf(buf, "%d %d", &num, &size) == 2) {
if (num > 0 && num <= (unsigned)count)
sizes[num - 1] = size;
else
/* warn about possible attempt to induce buffer overflow */
report(stderr, "Warning: ignoring bogus data for message sizes returned by server.\n");
}
}
...
}
```

Example 2

In the code snippet below, an unchecked integer value is used to reference an object in an array.

(Bad Code)

Example Language: **Java**

```
public String getValue(int index) {
return array[index];
}
```

If index is outside of the range of the array, this may result in an ArrayIndexOutOfBounds Exception being raised.

Example 3

In the following Java example the method displayProductSummary is called from a Web service servlet to retrieve product summary information for display to the user. The servlet obtains the integer value of the product number from the user and passes it to the displayProductSummary method. The displayProductSummary method passes the integer value of the product number to the getProductSummary method which obtains the product summary from the array object containing the project summaries using the integer value of the product number as the array index.

(Bad Code)

Example Language: **Java**

```
// Method called from servlet to obtain product information
public String displayProductSummary(int index) {

String productSummary = new String("");

try {
String productSummary = getProductSummary(index);

} catch (Exception ex) {...}

return productSummary;
}

public String getProductSummary(int index) {
return products[index];
}
```

In this example the integer value used as the array index that is provided by the user may be outside the allowable range of indices for the array which may provide unexpected results or may comes the application to fail. The integer value used for the array index should be validated to ensure that it is within the allowable range of indices for the array as in the following code.

(Good Code)

Example Language: **Java**

```
// Method called from servlet to obtain product information
public String displayProductSummary(int index) {

String productSummary = new String("");

try {
String productSummary = getProductSummary(index);

} catch (Exception ex) {...}

return productSummary;
}

public String getProductSummary(int index) {
String productSummary = "";

if ((index >= 0) && (index < MAX_PRODUCTS)) {
productSummary = products[index];
}
else {
System.err.println("index is out of bounds");
throw new IndexOutOfBoundsException();
}

return productSummary;
}
```

An alternative in Java would be to use one of the collection objects such as ArrayList that will automatically generate an exception if an attempt is made to access an array index that is out of bounds.

(Good Code)

Example Language: **Java**

```
ArrayList productArray = new ArrayList(MAX_PRODUCTS);
...
try {
productSummary = (String) productArray.get(index);
} catch (IndexOutOfBoundsException ex) {...}
```

Observed Examples

Reference	Description
CVE-2005-0369	large ID in packet used as array index
CVE-2001-1009	negative array index as argument to POP LIST command
CVE-2003-0721	Integer signedness error leads to negative array index
CVE-2004-1189	product does not properly track a count and a maximum number, which can lead to resultant array index overflow.
CVE-2007-5756	chain: device driver for packet-capturing software allows access to an unintended IOCTL with resultant array index error.

Potential Mitigations

Phase: Architecture and Design

Strategies: Input Validation; Libraries or Frameworks

Use an input validation framework such as Struts or the OWASP ESAPI Validation API. If you use Struts, be mindful of weaknesses covered by the CWE-101 category.

Phase: Architecture and Design

For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side, in order to avoid CWE-602. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the server.

Even though client-side checks provide minimal benefits with respect to server-side security, they are still useful. First, they can support intrusion detection. If the server receives input that should have been rejected by the client, then it may be an indication of an attack. Second, client-side error-checking can provide helpful feedback to the user about the expectations for valid input. Third, there may be a reduction in server-side processing time for accidental input errors, although this is typically a small savings.

Phase: Requirements

Strategy: Language Selection

Use a language with features that can automatically mitigate or eliminate out-of-bounds indexing errors.

For example, Ada allows the programmer to constrain the values of a variable and languages such as Java and Ruby will allow the programmer to handle exceptions when an out-of-bounds index is accessed.

Phase: Implementation

Strategy: Input Validation

Assume all input is malicious. Use an "accept known good" input validation strategy (i.e., use a whitelist). Reject any input that does not strictly conform to specifications, or transform it into something that does. Use a blacklist to reject any unexpected inputs and detect potential attacks.

When accessing a user-controlled array index, use a stringent range of values that are within the target array. Make sure that you do not allow negative values to be used. That is, verify the minimum as well as the maximum of the range of acceptable values.

Phase: Implementation

Be especially careful to validate your input when you invoke code that crosses language boundaries, such as from an interpreted language to native code. This could create an unexpected interaction between the language boundaries. Ensure that you are not violating any of the expectations of the language with which you are interfacing. For example, even though Java may not be susceptible to buffer overflows, providing a large argument in a call to native code might trigger an overflow.

Weakness Ordinalities

Ordinality	Description
Resultant	The most common condition situation leading to unchecked array indexing is the use of loop index variables as buffer indexes. If the end condition for the loop is subject to a flaw, the index can grow or shrink unbounded, therefore causing a buffer overflow or underflow. Another common situation leading to this condition is the use of a function's return value, or the resulting value of a calculation directly as an index in to a buffer.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	20	Improper Input Validation	**Development Concepts (primary)699** **Research Concepts (primary)1000**
ChildOf	Category	189	Numeric Errors	Development Concepts699
ChildOf	Category	633	Weaknesses that Affect Memory	**Resource-specific Weaknesses (primary)631**
ChildOf	Category	738	CERT C Secure Coding Section 04 - Integers (INT)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
ChildOf	Category	740	CERT C Secure Coding Section 06 - Arrays (ARR)	Weaknesses Addressed by the CERT C Secure Coding Standard734
ChildOf	Category	802	2010 Top 25 - Risky Resource Management	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
CanPrecede	Weakness Class	119	Failure to Constrain Operations within the Bounds of a Memory Buffer	Research Concepts1000
CanPrecede	Weakness Variant	789	Uncontrolled Memory Allocation	Research Concepts1000
PeerOf	Weakness Base	124	Buffer Underwrite ('Buffer Underflow')	Research Concepts1000

Theoretical Notes

An improperly validated array index might lead directly to the always-incorrect behavior of "access of array using out-of-bounds index."

Affected Resources

- Memory

f Causal Nature

Explicit

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Unchecked array indexing
PLOVER			INDEX - Array index overflow
CERT C Secure Coding	ARR00-C		Understand how arrays work
CERT C Secure Coding	ARR30-C		Guarantee that array indices are within the valid range
CERT C Secure Coding	ARR38-C		Do not add or subtract an integer to a pointer if the resulting value does not refer to a valid array element
CERT C Secure Coding	INT32-C		Ensure that operations on signed integers do not result in overflow

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
100	Overflow Buffers	

References

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 5, "Array Indexing Errors" Page 144. 2nd Edition. Microsoft. 2002.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	CLASP		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Sean Eidemiller	Cigital	External
	added/updated demonstrative examples		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Alternate Terms, Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Description, Name, Relationships		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Observed Examples, Other Notes, Potential Mitigations, Theoretical Notes, Weakness Ordinalities		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Demonstrative Examples, Detection Factors, Likelihood of Exploit, Potential Mitigations, References, Related Attack Patterns, Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Related Attack Patterns		

Previous Entry Names

Change Date	Previous Entry Name
2009-10-29	Unchecked Array Indexing

BACK TO TOP

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--;
               }
            }
        }
```

Information Leak Through Comments

Weakness ID: 615 *(Weakness Variant)* **Status:** Incomplete

Description

Description Summary

While adding general comments is very useful, some programmers tend to leave important data, such as: filenames related to the web application, old links or links which were not meant to be browsed by users, old code fragments, etc.

Extended Description

An attacker who finds these comments can map the application's structure and files, expose hidden parts of the site, and study the fragments of code to reverse engineer the application, which may help develop further attacks against the site.

Time of Introduction

- Implementation

Demonstrative Examples

Example 1

The following comment, embedded in a JSP, will be displayed in the resulting HTML output.

(Bad Code)

Example Languages: **HTML and JSP**

```
<!-- FIXME: calling this with more than 30 args kills the JDBC server -->
```

Observed Examples

Reference	Description
CVE-2007-6197	Version numbers and internal hostnames leaked in HTML comments.
CVE-2007-4072	CMS places full pathname of server in HTML comment.
CVE-2009-2431	blog software leaks real username in HTML comment.

Potential Mitigations

Remove comments which have sensitive information about the design/implementation of the application. Some of the comments may be exposed to the user and affect the security posture of the application.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Variant	540	Information Leak Through Source Code	**Development Concepts (primary)699** **Research Concepts (primary)1000**

Content History

Submissions

Submission Date	Submitter	Organization	Source
	Anonymous Tool Vendor (under NDA)		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Sean Eidemiller	Cigital	External
	added/updated demonstrative examples		
2008-07-01	Eric Dalci	Cigital	External
	updated Potential Mitigations, Time of Introduction		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2008-10-14	CWE Content Team	MITRE	Internal
	updated Description		
2009-03-10	CWE Content Team	MITRE	Internal

	updated Demonstrative Examples		
2009-07-27	CWE Content Team	MITRE	Internal
	updated Observed Examples, Taxonomy Mappings		

BACK TO TOP

Scanned Languages

Language	Hash Number	Change Date
CPP	4541647240435660	6/19/2024
Common	0105849645654507	6/19/2024