vul_files_10 Scan Report

Project Name	vul_files_10
Scan Start	Monday, January 6, 2025 6:42:45 PM
Preset	Checkmarx Default
Scan Time	01h:26m:14s
Lines Of Code Scanned	298861
Files Scanned	183
Report Creation Time	Monday, January 6, 2025 8:11:53 PM
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12
Team	CxServer
Checkmarx Version	8.7.0
Scan Type	Full
Source Origin	LocalPath
Density	6/10000 (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

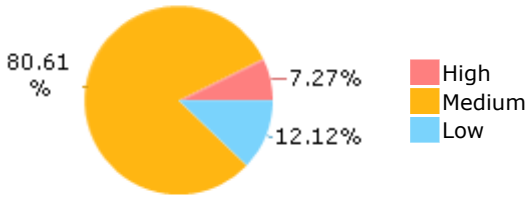
Results Limit

Results limit per query was set to 50

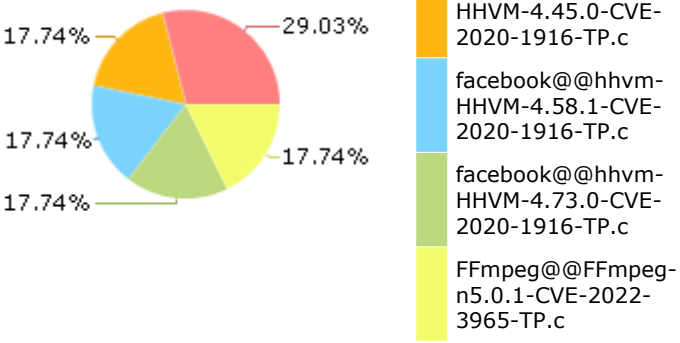
Selected Queries

Selected queries are listed in Result Summary

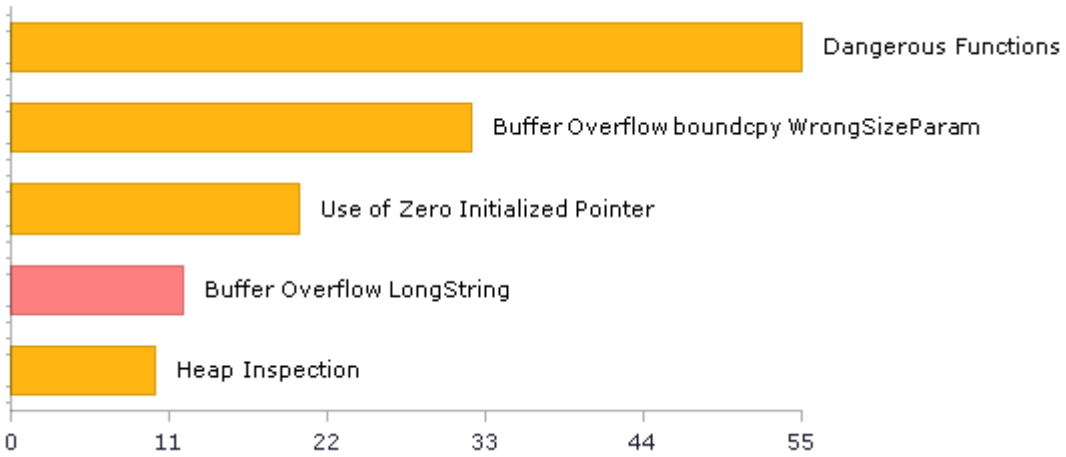
Result Summary



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	46	36
A2-Broken Authentication	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A3-Sensitive Data Exposure	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	App. Specific	10	10
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	55	55
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	10	10
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	55	55
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	50	41
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.	0	0
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.	1	1
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.	0	0
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.	10	10
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.	6	6

***** 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)	0	0
AC-4 Information Flow Enforcement (P1)	0	0
AC-6 Least Privilege (P1)	0	0
AU-9 Protection of Audit Information (P1)	0	0
CM-6 Configuration Settings (P2)	0	0
IA-5 Authenticator Management (P1)	0	0
IA-6 Authenticator Feedback (P2)	0	0
IA-8 Identification and Authentication (Non-Organizational Users) (P1)	0	0
SC-12 Cryptographic Key Establishment and Management (P1)	0	0
SC-13 Cryptographic Protection (P1)	0	0
SC-17 Public Key Infrastructure Certificates (P1)	0	0
SC-18 Mobile Code (P2)	0	0
SC-23 Session Authenticity (P1)*	0	0
SC-28 Protection of Information at Rest (P1)	0	0
SC-4 Information in Shared Resources (P1)	10	10
SC-5 Denial of Service Protection (P1)*	23	7
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*	24	15
SI-11 Error Handling (P2)*	6	6
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

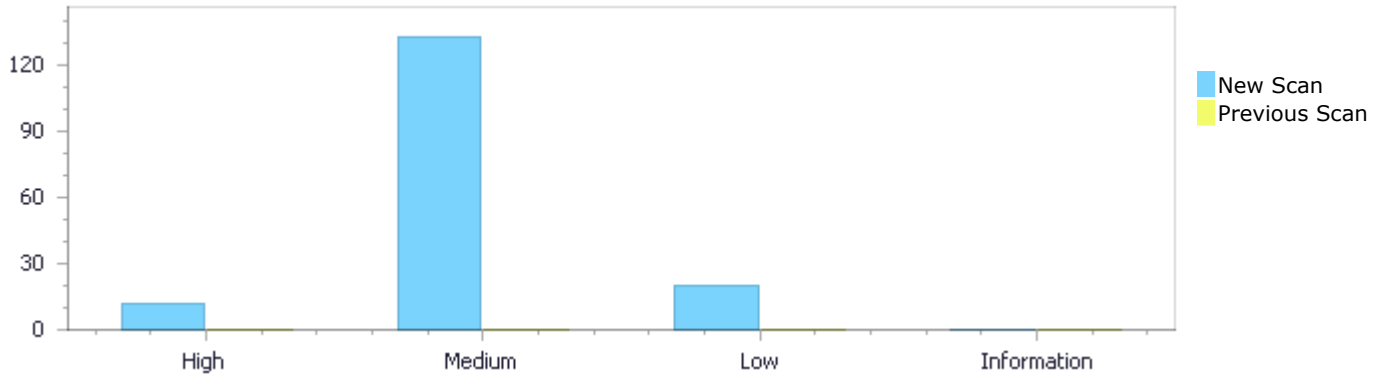
Scan Summary - Custom

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	12	133	20	0	165
Recurrent Issues	0	0	0	0	0
Total	12	133	20	0	165
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	12	133	20	0	165
Urgent	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0
Total	12	133	20	0	165

Result Summary

Vulnerability Type	Occurrences	Severity
Buffer Overflow LongString	12	High
Dangerous Functions	55	Medium
Buffer Overflow boundcpy WrongSizeParam	32	Medium
Use of Zero Initialized Pointer	20	Medium
Heap Inspection	10	Medium

MemoryFree on StackVariable	10	Medium
Integer Overflow	6	Medium
Unchecked Array Index	6	Low
Unchecked Return Value	6	Low
Sizeof Pointer Argument	3	Low
NULL Pointer Dereference	2	Low
Use of Sizeof On a Pointer Type	2	Low
Arithmenic Operation On Boolean	1	Low

10 Most Vulnerable Files

High and Medium Vulnerabilities

File Name	Issues Found
facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c	10
facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c	10
facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c	10
ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c	10
FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c	10
FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c	10
ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c	10
ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c	6
ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c	6
ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c	6

Scan Results Details

Buffer Overflow LongString

Query Path:
CPP\Cx\CPP Buffer Overflow\Buffer Overflow LongString Version:1

Categories

PCI DSS v3.2: PCI DSS (3.2) - 6.5.2 - Buffer overflows
NIST SP 800-53: SI-10 Information Input Validation (P1)
OWASP Top 10 2017: A1-Injection

Description

Buffer Overflow LongString\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=1
Status	New

The size of the buffer used by BF_set_key in tmp, at line 555 of facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *php_crypt_blowfish_rn passes to "8b \xd0\xc1\xd2\xcf\xcc\xd8", at line 826 of facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination
File	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Line	829	605
Object	"8b \xd0\xc1\xd2\xcf\xcc\xd8"	tmp

Code Snippet

File Name facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c

Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
829.    const char *test_key = "8b \xd0\xc1\xd2\xcf\xcc\xd8";
```

File Name facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c

Method static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

```
....
605.        tmp[0] |= (unsigned char)*ptr; /* correct */
```

Buffer Overflow LongString\Path 2:

Severity	High
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12

	&pathid=2
Status	New

The size of the buffer used by BF_set_key in tmp, at line 555 of facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *php_crypt_blowfish_rn passes to "8b \xd0\xc1\xd2\xcf\xcc\xd8", at line 826 of facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination
File	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Line	829	607
Object	"8b \xd0\xc1\xd2\xcf\xcc\xd8"	tmp

Code Snippet
File Name facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
829.    const char *test_key = "8b \xd0\xc1\xd2\xcf\xcc\xd8";
```

File Name facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Method static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

```
....
607.        tmp[1] |= (BF_word_signed)(signed char)*ptr; /* bug */
```

Buffer Overflow LongString\Path 3:

Severity	High
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=3
Status	New

The size of the buffer used by BF_set_key in tmp, at line 555 of facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *php_crypt_blowfish_rn passes to "\xff\xa3", at line 826 of facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination
File	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Line	868	605
Object	"\xff\xa3"	tmp

Code Snippet
File Name facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
868.      const char *k = "\xff\xa3" "34" "\xff\xff\xff\xa3" "345";
```

File Name facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c

Method static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

```
....
605.        tmp[0] |= (unsigned char)*ptr; /* correct */
```

Buffer Overflow LongString\Path 4:

Severity	High
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=4
Status	New

The size of the buffer used by BF_set_key in tmp, at line 555 of facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *php_crypt_blowfish_rn passes to "\xff\xa3", at line 826 of facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination
File	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Line	868	607
Object	"\xff\xa3"	tmp

Code Snippet

File Name facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c

Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
868.      const char *k = "\xff\xa3" "34" "\xff\xff\xff\xa3" "345";
```

File Name facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c

Method static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

```
....
607.        tmp[1] |= (BF_word_signed)(signed char)*ptr; /* bug */
```

Buffer Overflow LongString\Path 5:

Severity	High
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=5

Status	New

The size of the buffer used by BF_set_key in tmp, at line 555 of facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *php_crypt_blowfish_rn passes to "8b \xd0\xc1\xd2\xcf\xcc\xd8", at line 826 of facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination
File	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c
Line	829	607
Object	"8b \xd0\xc1\xd2\xcf\xcc\xd8"	tmp

Code Snippet

File Name facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c

Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
829.    const char *test_key = "8b \xd0\xc1\xd2\xcf\xcc\xd8";
```

File Name facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c

Method static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

```
....
607.        tmp[1] |= (BF_word_signed)(signed char)*ptr; /* bug */
```

Buffer Overflow LongString\Path 6:

Severity	High
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=6
Status	New

The size of the buffer used by BF_set_key in tmp, at line 555 of facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *php_crypt_blowfish_rn passes to "8b \xd0\xc1\xd2\xcf\xcc\xd8", at line 826 of facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination
File	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c
Line	829	605
Object	"8b \xd0\xc1\xd2\xcf\xcc\xd8"	tmp

Code Snippet

File Name facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c

Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
829.    const char *test_key = "8b \xd0\xc1\xd2\xcf\xcc\xd8";
```

File Name facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c

Method static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

```
....
605.        tmp[0] |= (unsigned char)*ptr; /* correct */
```

Buffer Overflow LongString\Path 7:

Severity	High
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=7
Status	New

The size of the buffer used by BF_set_key in tmp, at line 555 of facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *php_crypt_blowfish_rn passes to "\xff\xa3", at line 826 of facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination
File	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c
Line	868	607
Object	"\xff\xa3"	tmp

Code Snippet

File Name facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c

Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
868.      const char *k = "\xff\xa3" "34" "\xff\xff\xff\xa3" "345";
```

File Name facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c

Method static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

```
....
607.        tmp[1] |= (BF_word_signed)(signed char)*ptr; /* bug */
```

Buffer Overflow LongString\Path 8:

Severity	High
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=8

Status	New

The size of the buffer used by BF_set_key in tmp, at line 555 of facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *php_crypt_blowfish_rn passes to "\xff\xa3", at line 826 of facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination
File	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c
Line	868	605
Object	"\xff\xa3"	tmp

Code Snippet

File Name facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c

Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
868.      const char *k = "\xff\xa3" "34" "\xff\xff\xff\xa3" "345";
```

File Name facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c

Method static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

```
....
605.        tmp[0] |= (unsigned char)*ptr; /* correct */
```

Buffer Overflow LongString\Path 9:

Severity	High
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=9
Status	New

The size of the buffer used by BF_set_key in tmp, at line 555 of facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *php_crypt_blowfish_rn passes to "8b \xd0\xc1\xd2\xcf\xcc\xd8", at line 826 of facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination
File	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c
Line	829	607
Object	"8b \xd0\xc1\xd2\xcf\xcc\xd8"	tmp

Code Snippet

File Name facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c

Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
829.    const char *test_key = "8b \xd0\xc1\xd2\xcf\xcc\xd8";
```

File Name facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c

Method static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

```
....
607.        tmp[1] |= (BF_word_signed)(signed char)*ptr; /* bug */
```

Buffer Overflow LongString\Path 10:

Severity	High
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=10
Status	New

The size of the buffer used by BF_set_key in tmp, at line 555 of facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *php_crypt_blowfish_rn passes to "8b \xd0\xc1\xd2\xcf\xcc\xd8", at line 826 of facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination
File	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c
Line	829	605
Object	"8b \xd0\xc1\xd2\xcf\xcc\xd8"	tmp

Code Snippet

File Name facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c

Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
829.    const char *test_key = "8b \xd0\xc1\xd2\xcf\xcc\xd8";
```

File Name facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c

Method static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

```
....
605.        tmp[0] |= (unsigned char)*ptr; /* correct */
```

Buffer Overflow LongString\Path 11:

Severity	High
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=11

Status	New

The size of the buffer used by BF_set_key in tmp, at line 555 of facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *php_crypt_blowfish_rn passes to "\xff\xa3", at line 826 of facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination
File	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c
Line	868	607
Object	"\xff\xa3"	tmp

Code Snippet

File Name facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c

Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
868.      const char *k = "\xff\xa3" "34" "\xff\xff\xff\xa3" "345";
```

File Name facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c

Method static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

```
....
607.        tmp[1] |= (BF_word_signed)(signed char)*ptr; /* bug */
```

Buffer Overflow LongString\Path 12:

Severity	High
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=12
Status	New

The size of the buffer used by BF_set_key in tmp, at line 555 of facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *php_crypt_blowfish_rn passes to "\xff\xa3", at line 826 of facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination
File	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c
Line	868	605
Object	"\xff\xa3"	tmp

Code Snippet

File Name facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c

Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
868.      const char *k = "\xff\xa3" "34" "\xff\xff\xff\xa3" "345";
```

File Name facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c

Method static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

```
....
605.        tmp[0] |= (unsigned char)*ptr; /* correct */
```

Dangerous Functions

Query Path:
CPP\Cx\CPP Medium Threat\Dangerous Functions Version:1

Categories

OWASP Top 10 2013: A9-Using Components with Known Vulnerabilities
OWASP Top 10 2017: A9-Using Components with Known Vulnerabilities

Description

Dangerous Functions\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=62
Status	New

The dangerous function, memcpy, was found in use at line 43 in facebook@@hermes-v0.6.0-CVE-2022-32234-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hermes-v0.6.0-CVE-2022-32234-TP.c	facebook@@hermes-v0.6.0-CVE-2022-32234-TP.c
Line	58	58
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hermes-v0.6.0-CVE-2022-32234-TP.c
Method void SmallVectorBase::grow_pod(void *FirstEl, size_t MinCapacity,

```
....
58.       memcpy(NewElts, this->BeginX, size() * TSize);
```

Dangerous Functions\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12

	&pathid=63
Status	New

The dangerous function, memcpy, was found in use at line 43 in facebook@@hermes-v0.8.0-CVE-2022-32234-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hermes-v0.8.0-CVE-2022-32234-TP.c	facebook@@hermes-v0.8.0-CVE-2022-32234-TP.c
Line	58	58
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hermes-v0.8.0-CVE-2022-32234-TP.c
Method void SmallVectorBase::grow_pod(void *FirstEl, size_t MinCapacity,

```
....
58.      memcpy(NewElts, this->BeginX, size() * TSize);
```

Dangerous Functions\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=64
Status	New

The dangerous function, memcpy, was found in use at line 43 in facebook@@hermes-v0.9.0-CVE-2022-32234-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hermes-v0.9.0-CVE-2022-32234-TP.c	facebook@@hermes-v0.9.0-CVE-2022-32234-TP.c
Line	58	58
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hermes-v0.9.0-CVE-2022-32234-TP.c
Method void SmallVectorBase::grow_pod(void *FirstEl, size_t MinCapacity,

```
....
58.      memcpy(NewElts, this->BeginX, size() * TSize);
```

Dangerous Functions\Path 4:

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

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=65
Status	New

The dangerous function, memcpy, was found in use at line 826 in facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Line	854	854
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
854.    memcpy(buf.s, test_setting, sizeof(buf.s));
```

Dangerous Functions\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=66
Status	New

The dangerous function, memcpy, was found in use at line 656 in facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Line	708	708
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Method static char *BF_crypt(const char *key, const char *setting,

```
....
708.    memcpy(data.ctx.S, BF_init_state.S, sizeof(data.ctx.S));
```

Dangerous Functions\Path 6:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=67
Status	New

The dangerous function, memcpy, was found in use at line 656 in facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Line	778	778
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Method static char *BF_crypt(const char *key, const char *setting,

```
....
778.    memcpy(output, setting, 7 + 22 - 1);
```

Dangerous Functions\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=68
Status	New

The dangerous function, memcpy, was found in use at line 826 in facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c
Line	854	854
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c
Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
854.    memcpy(buf.s, test_setting, sizeof(buf.s));
```

Dangerous Functions\Path 8:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=69
Status	New

The dangerous function, memcpy, was found in use at line 656 in facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c
Line	708	708
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c
Method static char *BF_crypt(const char *key, const char *setting,

```
....
708.    memcpy(data.ctx.S, BF_init_state.S, sizeof(data.ctx.S));
```

Dangerous Functions\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=70
Status	New

The dangerous function, memcpy, was found in use at line 656 in facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c
Line	778	778
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c
Method static char *BF_crypt(const char *key, const char *setting,

```
....
778.    memcpy(output, setting, 7 + 22 - 1);
```

Dangerous Functions\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=71
Status	New

The dangerous function, memcpy, was found in use at line 826 in facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c
Line	854	854
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c
Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
854.    memcpy(buf.s, test_setting, sizeof(buf.s));
```

Dangerous Functions\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=72
Status	New

The dangerous function, memcpy, was found in use at line 656 in facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c
Line	708	708
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c
Method static char *BF_crypt(const char *key, const char *setting,

```
....
708.    memcpy(data.ctx.S, BF_init_state.S, sizeof(data.ctx.S));
```

Dangerous Functions\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=73
Status	New

The dangerous function, memcpy, was found in use at line 656 in facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c
Line	778	778
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c
Method static char *BF_crypt(const char *key, const char *setting,

```
....
778.    memcpy(output, setting, 7 + 22 - 1);
```

Dangerous Functions\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=74
Status	New

The dangerous function, memcpy, was found in use at line 190 in FFmpeg@@FFmpeg-n4.3.2-CVE-2020-22021-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	FFmpeg@@FFmpeg-n4.3.2-CVE-2020-22021-FP.c	FFmpeg@@FFmpeg-n4.3.2-CVE-2020-22021-FP.c
Line	222	222
Object	memcpy	memcpy

Code Snippet
File Name FFmpeg@@FFmpeg-n4.3.2-CVE-2020-22021-FP.c
Method static int filter_slice(AVFilterContext *ctx, void *arg, int jobnr, int nb_jobs)

```
....
222.              memcpy(&td->frame->data[td->plane][y * td->frame-
>linesize[td->plane]],
```

Dangerous Functions\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=75
Status	New

The dangerous function, memcpy, was found in use at line 190 in ffmpeg@@ffmpeg-n4.3.2-CVE-2020-22021-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ffmpeg@@ffmpeg-n4.3.2-CVE-2020-22021-TP.c	ffmpeg@@ffmpeg-n4.3.2-CVE-2020-22021-TP.c
Line	222	222
Object	memcpy	memcpy

Code Snippet
File Name ffmpeg@@ffmpeg-n4.3.2-CVE-2020-22021-TP.c
Method static int filter_slice(AVFilterContext *ctx, void *arg, int jobnr, int nb_jobs)

```
....
222.              memcpy(&td->frame->data[td->plane][y * td->frame-
>linesize[td->plane]],
```

Dangerous Functions\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=76
Status	New

The dangerous function, memcpy, was found in use at line 52 in ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Line	54	54
Object	memcpy	memcpy

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Method static void ref_from_h264pic(H264Ref *dst, H264Picture *src)

```
....
54.      memcpy(dst->data,     src->f->data,     sizeof(dst->data));
```

Dangerous Functions\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=77
Status	New

The dangerous function, memcpy, was found in use at line 52 in ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Line	55	55
Object	memcpy	memcpy

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Method static void ref_from_h264pic(H264Ref *dst, H264Picture *src)

```
....
55.      memcpy(dst->linesize, src->f->linesize, sizeof(dst->linesize));
```

Dangerous Functions\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=78
Status	New

The dangerous function, memcpy, was found in use at line 112 in FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c
Line	203	203
Object	memcpy	memcpy

Code Snippet	
File Name	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c
Method	static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
203.                     memcpy(block_values + y * 4, pixel_ptr + y *
stride, 4);
```

Dangerous Functions\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=79
Status	New

The dangerous function, memcpy, was found in use at line 112 in FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c
Line	208	208
Object	memcpy	memcpy

Code Snippet	
File Name	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c
Method	static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
208.                     memcpy(distinct_values, s->next_distinct_values,
sizeof(s->distinct_values));
```

Dangerous Functions\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=80
Status	New

The dangerous function, memcpy, was found in use at line 489 in FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c

Line	516	516
Object	memcpy	memcpy

Code Snippet
File Name FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c
Method static int smc_encode_frame(AVCodecContext *avctx, AVPacket *pkt,

```
....
516.      memcpy(pal, frame->data[1], AVPALETTE_SIZE);
```

Dangerous Functions\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=81
Status	New

The dangerous function, memcpy, was found in use at line 112 in FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Line	203	203
Object	memcpy	memcpy

Code Snippet
File Name FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Method static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
203.                  memcpy(block_values + y * 4, pixel_ptr + y *
stride, 4);
```

Dangerous Functions\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=82
Status	New

The dangerous function, memcpy, was found in use at line 112 in FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination

File	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Line	208	208
Object	memcpy	memcpy

Code Snippet
File Name FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Method static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
208.                 memcpy(distinct_values, s->next_distinct_values,
sizeof(s->distinct_values));
```

Dangerous Functions\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=83
Status	New

The dangerous function, memcpy, was found in use at line 489 in FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Line	516	516
Object	memcpy	memcpy

Code Snippet
File Name FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Method static int smc_encode_frame(AVCodecContext *avctx, AVPacket *pkt,

```
....
516.      memcpy(pal, frame->data[1], AVPALETTE_SIZE);
```

Dangerous Functions\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=84
Status	New

The dangerous function, memcpy, was found in use at line 112 in ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c
Line	203	203
Object	memcpy	memcpy

Code Snippet
File Name ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c
Method static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
203.                 memcpy(block_values + y * 4, pixel_ptr + y *
stride, 4);
```

Dangerous Functions\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=85
Status	New

The dangerous function, memcpy, was found in use at line 112 in ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c
Line	208	208
Object	memcpy	memcpy

Code Snippet
File Name ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c
Method static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
208.                 memcpy(distinct_values, s->next_distinct_values,
sizeof(s->distinct_values));
```

Dangerous Functions\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=86
Status	New

The dangerous function, memcpy, was found in use at line 489 in ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c
Line	516	516
Object	memcpy	memcpy

Code Snippet
File Name ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c
Method static int smc_encode_frame(AVCodecContext *avctx, AVPacket *pkt,

```
....
516.      memcpy(pal, frame->data[1], AVPALETTE_SIZE);
```

Dangerous Functions\Path 26:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=87
Status New

The dangerous function, memcpy, was found in use at line 100 in facebook@@hhvm-HHVM-4.101.0-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.101.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.101.0-CVE-2022-36937-TP.c
Line	105	105
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hhvm-HHVM-4.101.0-CVE-2022-36937-TP.c
Method int SSLSocket::passwdCallback(char* buf, int num, int /*verify*/, void* data) {

```
....
105.      memcpy(buf, passphrase.data(), passphrase.size() + 1);
```

Dangerous Functions\Path 27:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=88
Status New

The dangerous function, memcpy, was found in use at line 100 in facebook@@hhvm-HHVM-4.115.0-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.115.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.115.0-CVE-2022-36937-TP.c
Line	105	105
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hhvm-HHVM-4.115.0-CVE-2022-36937-TP.c
Method int SSLSocket::passwdCallback(char* buf, int num, int /*verify*/, void* data) {

```
....
105.      memcpy(buf, passphrase.data(), passphrase.size() + 1);
```

Dangerous Functions\Path 28:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=89
Status New

The dangerous function, memcpy, was found in use at line 100 in facebook@@hhvm-HHVM-4.147.0-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.147.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.147.0-CVE-2022-36937-TP.c
Line	105	105
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hhvm-HHVM-4.147.0-CVE-2022-36937-TP.c
Method int SSLSocket::passwdCallback(char* buf, int num, int /*verify*/, void* data) {

```
....
105.      memcpy(buf, passphrase.data(), passphrase.size() + 1);
```

Dangerous Functions\Path 29:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=90

Status	New

The dangerous function, memcpy, was found in use at line 100 in facebook@@hhvm-HHVM-4.167.0-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.167.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.167.0-CVE-2022-36937-TP.c
Line	105	105
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hhvm-HHVM-4.167.0-CVE-2022-36937-TP.c
Method int SSLSocket::passwdCallback(char* buf, int num, int /*verify*/, void* data) {

```
....
105.      memcpy(buf, passphrase.data(), passphrase.size() + 1);
```

Dangerous Functions\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=91
Status	New

The dangerous function, memcpy, was found in use at line 100 in facebook@@hhvm-HHVM-4.45.0-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.45.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.45.0-CVE-2022-36937-TP.c
Line	105	105
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hhvm-HHVM-4.45.0-CVE-2022-36937-TP.c
Method int SSLSocket::passwdCallback(char* buf, int num, int /*verify*/, void* data) {

```
....
105.      memcpy(buf, passphrase.data(), passphrase.size() + 1);
```

Dangerous Functions\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12

	&pathid=92
Status	New

The dangerous function, memcpy, was found in use at line 100 in facebook@@hhvm-HHVM-4.58.1-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.58.1-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.58.1-CVE-2022-36937-TP.c
Line	105	105
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hhvm-HHVM-4.58.1-CVE-2022-36937-TP.c
Method int SSLSocket::passwdCallback(char* buf, int num, int /*verify*/, void* data) {

```
....
105.      memcpy(buf, passphrase.data(), passphrase.size() + 1);
```

Dangerous Functions\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=93
Status	New

The dangerous function, memcpy, was found in use at line 100 in facebook@@hhvm-HHVM-4.73.0-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.73.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.73.0-CVE-2022-36937-TP.c
Line	105	105
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hhvm-HHVM-4.73.0-CVE-2022-36937-TP.c
Method int SSLSocket::passwdCallback(char* buf, int num, int /*verify*/, void* data) {

```
....
105.      memcpy(buf, passphrase.data(), passphrase.size() + 1);
```

Dangerous Functions\Path 33:

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

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=94
Status	New

The dangerous function, memcpy, was found in use at line 100 in facebook@@hhvm-nightly-2020.12.10-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-nightly-2020.12.10-CVE-2022-36937-TP.c	facebook@@hhvm-nightly-2020.12.10-CVE-2022-36937-TP.c
Line	105	105
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hhvm-nightly-2020.12.10-CVE-2022-36937-TP.c
Method int SSLSocket::passwdCallback(char* buf, int num, int /*verify*/, void* data) {

```
....
105.      memcpy(buf, passphrase.data(), passphrase.size() + 1);
```

Dangerous Functions\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=95
Status	New

The dangerous function, memcpy, was found in use at line 100 in facebook@@hhvm-nightly-2021.10.10-CVE-2022-36937-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-nightly-2021.10.10-CVE-2022-36937-FP.c	facebook@@hhvm-nightly-2021.10.10-CVE-2022-36937-FP.c
Line	105	105
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hhvm-nightly-2021.10.10-CVE-2022-36937-FP.c
Method int SSLSocket::passwdCallback(char* buf, int num, int /*verify*/, void* data) {

```
....
105.      memcpy(buf, passphrase.data(), passphrase.size() + 1);
```

Dangerous Functions\Path 35:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=96
Status	New

The dangerous function, memcpy, was found in use at line 100 in facebook@@hhvm-nightly-2022.11.25-CVE-2022-36937-FP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-nightly-2022.11.25-CVE-2022-36937-FP.c	facebook@@hhvm-nightly-2022.11.25-CVE-2022-36937-FP.c
Line	105	105
Object	memcpy	memcpy

Code Snippet
File Name facebook@@hhvm-nightly-2022.11.25-CVE-2022-36937-FP.c
Method int SSLSocket::passwdCallback(char* buf, int num, int /*verify*/, void* data) {

```
....
105.      memcpy(buf, passphrase.data(), passphrase.size() + 1);
```

Dangerous Functions\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=97
Status	New

The dangerous function, strlen, was found in use at line 578 in facebook@@hhvm-HHVM-4.101.0-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.101.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.101.0-CVE-2022-36937-TP.c
Line	619	619
Object	strlen	strlen

Code Snippet
File Name facebook@@hhvm-HHVM-4.101.0-CVE-2022-36937-TP.c
Method bool SSLSocket::applyVerificationPolicy(X509 *peer) {

```
....
619.      } else if (name_len != (int)strlen(buf)) {
```

Dangerous Functions\Path 37:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=98
Status	New

The dangerous function, strlen, was found in use at line 578 in facebook@@hhvm-HHVM-4.101.0-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.101.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.101.0-CVE-2022-36937-TP.c
Line	629	629
Object	strlen	strlen

Code Snippet
File Name facebook@@hhvm-HHVM-4.101.0-CVE-2022-36937-TP.c
Method bool SSLSocket::applyVerificationPolicy(X509 *peer) {

```
....
629.      if (!match && strlen(buf) > 3 && buf[0] == '*' && buf[1] ==
'.') {
```

Dangerous Functions\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=99
Status	New

The dangerous function, strlen, was found in use at line 578 in facebook@@hhvm-HHVM-4.115.0-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.115.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.115.0-CVE-2022-36937-TP.c
Line	619	619
Object	strlen	strlen

Code Snippet
File Name facebook@@hhvm-HHVM-4.115.0-CVE-2022-36937-TP.c
Method bool SSLSocket::applyVerificationPolicy(X509 *peer) {

```
....
619.      } else if (name_len != (int)strlen(buf)) {
```

Dangerous Functions\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=100
Status	New

The dangerous function, strlen, was found in use at line 578 in facebook@@hhvm-HHVM-4.115.0-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.115.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.115.0-CVE-2022-36937-TP.c
Line	629	629
Object	strlen	strlen

Code Snippet
File Name facebook@@hhvm-HHVM-4.115.0-CVE-2022-36937-TP.c
Method bool SSLSocket::applyVerificationPolicy(X509 *peer) {

```
....
629.      if (!match && strlen(buf) > 3 && buf[0] == '*' && buf[1] ==
'.') {
```

Dangerous Functions\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=101
Status	New

The dangerous function, strlen, was found in use at line 578 in facebook@@hhvm-HHVM-4.147.0-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.147.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.147.0-CVE-2022-36937-TP.c
Line	619	619
Object	strlen	strlen

Code Snippet
File Name facebook@@hhvm-HHVM-4.147.0-CVE-2022-36937-TP.c
Method bool SSLSocket::applyVerificationPolicy(X509 *peer) {

```
....
619.      } else if (name_len != (int)strlen(buf)) {
```

Dangerous Functions\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=102
Status	New

The dangerous function, strlen, was found in use at line 578 in facebook@@hhvm-HHVM-4.147.0-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.147.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.147.0-CVE-2022-36937-TP.c
Line	629	629
Object	strlen	strlen

Code Snippet
File Name facebook@@hhvm-HHVM-4.147.0-CVE-2022-36937-TP.c
Method bool SSLSocket::applyVerificationPolicy(X509 *peer) {

```
....
629.      if (!match && strlen(buf) > 3 && buf[0] == '*' && buf[1] ==
'.') {
```

Dangerous Functions\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=103
Status	New

The dangerous function, strlen, was found in use at line 578 in facebook@@hhvm-HHVM-4.167.0-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.167.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.167.0-CVE-2022-36937-TP.c
Line	619	619
Object	strlen	strlen

Code Snippet

File Name	facebook@@hhvm-HHVM-4.167.0-CVE-2022-36937-TP.c
Method	bool SSLSocket::applyVerificationPolicy(X509 *peer) {

```
....
619.        } else if (name_len != (int)strlen(buf)) {
```

Dangerous Functions\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=104
Status	New

The dangerous function, strlen, was found in use at line 578 in facebook@@hhvm-HHVM-4.167.0-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.167.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.167.0-CVE-2022-36937-TP.c
Line	629	629
Object	strlen	strlen

Code Snippet	
File Name	facebook@@hhvm-HHVM-4.167.0-CVE-2022-36937-TP.c
Method	bool SSLSocket::applyVerificationPolicy(X509 *peer) {

```
....
629.        if (!match && strlen(buf) > 3 && buf[0] == '*' && buf[1] ==
'.') {
```

Dangerous Functions\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=105
Status	New

The dangerous function, strlen, was found in use at line 578 in facebook@@hhvm-HHVM-4.45.0-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.45.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.45.0-CVE-2022-36937-TP.c
Line	619	619
Object	strlen	strlen

Code Snippet	
File Name	facebook@@hhvm-HHVM-4.45.0-CVE-2022-36937-TP.c
Method	bool SSLSocket::applyVerificationPolicy(X509 *peer) {

```
....
619.        } else if (name_len != (int)strlen(buf)) {
```

Dangerous Functions\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=106
Status	New

The dangerous function, strlen, was found in use at line 578 in facebook@@hhvm-HHVM-4.45.0-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.45.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.45.0-CVE-2022-36937-TP.c
Line	629	629
Object	strlen	strlen

Code Snippet	
File Name	facebook@@hhvm-HHVM-4.45.0-CVE-2022-36937-TP.c
Method	bool SSLSocket::applyVerificationPolicy(X509 *peer) {

```
....
629.        if (!match && strlen(buf) > 3 && buf[0] == '*' && buf[1] ==
'.') {
```

Dangerous Functions\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=107
Status	New

The dangerous function, strlen, was found in use at line 578 in facebook@@hhvm-HHVM-4.58.1-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.58.1-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.58.1-CVE-2022-36937-TP.c
Line	619	619

Object	strlen	strlen

Code Snippet
File Name facebook@@hhvm-HHVM-4.58.1-CVE-2022-36937-TP.c
Method bool SSLSocket::applyVerificationPolicy(X509 *peer) {

```
....
619.      } else if (name_len != (int)strlen(buf)) {
```

Dangerous Functions\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=108
Status	New

The dangerous function, strlen, was found in use at line 578 in facebook@@hhvm-HHVM-4.58.1-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.58.1-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.58.1-CVE-2022-36937-TP.c
Line	629	629
Object	strlen	strlen

Code Snippet
File Name facebook@@hhvm-HHVM-4.58.1-CVE-2022-36937-TP.c
Method bool SSLSocket::applyVerificationPolicy(X509 *peer) {

```
....
629.      if (!match && strlen(buf) > 3 && buf[0] == '*' && buf[1] ==
'.') {
```

Dangerous Functions\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=109
Status	New

The dangerous function, strlen, was found in use at line 578 in facebook@@hhvm-HHVM-4.73.0-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.73.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.73.0-CVE-2022-36937-TP.c

Line	619	619
Object	strlen	strlen

Code Snippet
File Name facebook@@hhvm-HHVM-4.73.0-CVE-2022-36937-TP.c
Method bool SSLSocket::applyVerificationPolicy(X509 *peer) {

```
....
619.      } else if (name_len != (int)strlen(buf)) {
```

Dangerous Functions\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=110
Status	New

The dangerous function, strlen, was found in use at line 578 in facebook@@hhvm-HHVM-4.73.0-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	facebook@@hhvm-HHVM-4.73.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.73.0-CVE-2022-36937-TP.c
Line	629	629
Object	strlen	strlen

Code Snippet
File Name facebook@@hhvm-HHVM-4.73.0-CVE-2022-36937-TP.c
Method bool SSLSocket::applyVerificationPolicy(X509 *peer) {

```
....
629.      if (!match && strlen(buf) > 3 && buf[0] == '*' && buf[1] ==
'.') {
```

Dangerous Functions\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=111
Status	New

The dangerous function, strlen, was found in use at line 578 in facebook@@hhvm-nightly-2020.12.10-CVE-2022-36937-TP.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination

File	facebook@@hhvm-nightly-2020.12.10-CVE-2022-36937-TP.c	facebook@@hhvm-nightly-2020.12.10-CVE-2022-36937-TP.c
Line	619	619
Object	strlen	strlen

Code Snippet
File Name facebook@@hhvm-nightly-2020.12.10-CVE-2022-36937-TP.c
Method bool SSLSocket::applyVerificationPolicy(X509 *peer) {

```
....
619.      } else if (name_len != (int)strlen(buf)) {
```

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-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=13
Status	New

The size of the buffer used by *php_crypt_blowfish_rn in Namespace1979057365, at line 826 of facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *php_crypt_blowfish_rn passes to Namespace1979057365, at line 826 of facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination
File	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Line	854	854
Object	Namespace1979057365	Namespace1979057365

Code Snippet
File Name facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
854.      memcpy(buf.s, test_setting, sizeof(buf.s));
```

Buffer Overflow boundcpy WrongSizeParam\Path 2:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=14
Status	New

The size of the buffer used by *BF_crypt in Namespace1979057365, at line 656 of facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *BF_crypt passes to Namespace1979057365, at line 656 of facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination
File	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Line	708	708
Object	Namespace1979057365	Namespace1979057365

Code Snippet
File Name facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Method static char *BF_crypt(const char *key, const char *setting,

```
....
708.    memcpy(data.ctx.S, BF_init_state.S, sizeof(data.ctx.S));
```

Buffer Overflow boundcpy WrongSizeParam\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=15
Status	New

The size of the buffer used by *php_crypt_blowfish_rn in Namespace1277136062, at line 826 of facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *php_crypt_blowfish_rn passes to Namespace1277136062, at line 826 of facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination
File	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c
Line	854	854
Object	Namespace1277136062	Namespace1277136062

Code Snippet
File Name facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c
Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
854.    memcpy(buf.s, test_setting, sizeof(buf.s));
```

Buffer Overflow boundcpy WrongSizeParam\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=16
Status	New

The size of the buffer used by *BF_crypt in Namespace1277136062, at line 656 of facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *BF_crypt passes to Namespace1277136062, at line 656 of facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination
File	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c
Line	708	708
Object	Namespace1277136062	Namespace1277136062

Code Snippet
File Name facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c
Method static char *BF_crypt(const char *key, const char *setting,

```
....
708.    memcpy(data.ctx.S, BF_init_state.S, sizeof(data.ctx.S));
```

Buffer Overflow boundcpy WrongSizeParam\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=17
Status	New

The size of the buffer used by *php_crypt_blowfish_rn in Namespace892261426, at line 826 of facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *php_crypt_blowfish_rn passes to Namespace892261426, at line 826 of facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination
File	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c
Line	854	854
Object	Namespace892261426	Namespace892261426

Code Snippet
File Name facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c
Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
854.    memcpy(buf.s, test_setting, sizeof(buf.s));
```

Buffer Overflow boundcpy WrongSizeParam\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=18
Status	New

The size of the buffer used by *BF_crypt in Namespace892261426, at line 656 of facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *BF_crypt passes to Namespace892261426, at line 656 of facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination
File	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c
Line	708	708
Object	Namespace892261426	Namespace892261426

Code Snippet
File Name facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c
Method static char *BF_crypt(const char *key, const char *setting,

```
....
708.    memcpy(data.ctx.S, BF_init_state.S, sizeof(data.ctx.S));
```

Buffer Overflow boundcpy WrongSizeParam\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=19
Status	New

The size of the buffer used by ref_from_h264pic in ->, at line 52 of ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ref_from_h264pic passes to ->, at line 52 of ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c, to overwrite the target buffer.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Line	54	54
Object	->	->

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c

Method	static void ref_from_h264pic(H264Ref *dst, H264Picture *src)

```
....
54.      memcpy(dst->data,      src->f->data,      sizeof(dst->data));
```

Buffer Overflow boundcpy WrongSizeParam\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=20
Status	New

The size of the buffer used by ref_from_h264pic in ->, at line 52 of ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ref_from_h264pic passes to ->, at line 52 of ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c, to overwrite the target buffer.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Line	55	55
Object	->	->

Code Snippet	
File Name	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Method	static void ref_from_h264pic(H264Ref *dst, H264Picture *src)

```
....
55.      memcpy(dst->linesize, src->f->linesize, sizeof(dst->linesize));
```

Buffer Overflow boundcpy WrongSizeParam\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=21
Status	New

The size of the buffer used by smc_encode_stream in ->, at line 112 of FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that smc_encode_stream passes to ->, at line 112 of FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c, to overwrite the target buffer.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c
Line	208	208
Object	->	->

Code Snippet

File Name	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c
Method	static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
208.                    memcpy(distinct_values, s->next_distinct_values,
sizeof(s->distinct_values));
```

Buffer Overflow boundcpy WrongSizeParam\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=22
Status	New

The size of the buffer used by smc_encode_stream in ->, at line 112 of FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that smc_encode_stream passes to ->, at line 112 of FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c, to overwrite the target buffer.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Line	208	208
Object	->	->

Code Snippet

File Name	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Method	static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
208.                    memcpy(distinct_values, s->next_distinct_values,
sizeof(s->distinct_values));
```

Buffer Overflow boundcpy WrongSizeParam\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=23
Status	New

The size of the buffer used by smc_encode_stream in ->, at line 112 of ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that smc_encode_stream passes to ->, at line 112 of ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c, to overwrite the target buffer.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c
Line	208	208

Object	->	->

Code Snippet
File Name ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c
Method static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
208.                    memcpy(distinct_values, s->next_distinct_values,
sizeof(s->distinct_values));
```

Buffer Overflow boundcpy WrongSizeParam\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=24
Status	New

The size of the buffer used by *php_crypt_blowfish_rn in Namespace1979057365, at line 826 of facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *php_crypt_blowfish_rn passes to Namespace1979057365, at line 826 of facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination
File	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Line	857	857
Object	Namespace1979057365	Namespace1979057365

Code Snippet
File Name facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
857.    memset(buf.o, 0x55, sizeof(buf.o));
```

Buffer Overflow boundcpy WrongSizeParam\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=25
Status	New

The size of the buffer used by *php_crypt_blowfish_rn in Namespace1277136062, at line 826 of facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *php_crypt_blowfish_rn passes to Namespace1277136062, at line 826 of facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination

File	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c
Line	857	857
Object	Namespace1277136062	Namespace1277136062

Code Snippet
File Name facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c
Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
857.    memset(buf.o, 0x55, sizeof(buf.o));
```

Buffer Overflow boundcpy WrongSizeParam\Path 14:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=26
Status New

The size of the buffer used by *php_crypt_blowfish_rn in Namespace892261426, at line 826 of facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *php_crypt_blowfish_rn passes to Namespace892261426, at line 826 of facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c, to overwrite the target buffer.

	Source	Destination
File	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c
Line	857	857
Object	Namespace892261426	Namespace892261426

Code Snippet
File Name facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c
Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
857.    memset(buf.o, 0x55, sizeof(buf.o));
```

Buffer Overflow boundcpy WrongSizeParam\Path 15:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=27
Status New

The size of the buffer used by ff_h264_remove_all_refs in ->, at line 565 of ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow

attack, using the source buffer that ff_h264_remove_all_refs passes to ->, at line 565 of ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c, to overwrite the target buffer.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Line	585	585
Object	->	->

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Method void ff_h264_remove_all_refs(H264Context *h)

```
....
585.      memset(h->default_ref, 0, sizeof(h->default_ref));
```

Buffer Overflow boundcpy WrongSizeParam\Path 16:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=28
Status New

The size of the buffer used by smc_encode_stream in ->, at line 112 of FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that smc_encode_stream passes to ->, at line 112 of FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c, to overwrite the target buffer.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c
Line	129	129
Object	->	->

Code Snippet
File Name FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c
Method static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
129.      memset(s->color_pairs, 0, sizeof(s->color_pairs));
```

Buffer Overflow boundcpy WrongSizeParam\Path 17:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=29
Status New

The size of the buffer used by smc_encode_stream in ->, at line 112 of FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that smc_encode_stream passes to ->, at line 112 of FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c, to overwrite the target buffer.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c
Line	130	130
Object	->	->

Code Snippet
File Name FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c
Method static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
130.      memset(s->color_quads, 0, sizeof(s->color_quads));
```

Buffer Overflow boundcpy WrongSizeParam\Path 18:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=30
Status New

The size of the buffer used by smc_encode_stream in ->, at line 112 of FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that smc_encode_stream passes to ->, at line 112 of FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c, to overwrite the target buffer.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c
Line	131	131
Object	->	->

Code Snippet
File Name FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c
Method static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
131.      memset(s->color_octets, 0, sizeof(s->color_octets));
```

Buffer Overflow boundcpy WrongSizeParam\Path 19:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=31
Status New

The size of the buffer used by smc_encode_stream in ->, at line 112 of FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that smc_encode_stream passes to ->, at line 112 of FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c, to overwrite the target buffer.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Line	129	129
Object	->	->

Code Snippet
File Name FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Method static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
129.      memset(s->color_pairs, 0, sizeof(s->color_pairs));
```

Buffer Overflow boundcpy WrongSizeParam\Path 20:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=32
Status New

The size of the buffer used by smc_encode_stream in ->, at line 112 of FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that smc_encode_stream passes to ->, at line 112 of FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c, to overwrite the target buffer.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Line	130	130
Object	->	->

Code Snippet
File Name FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Method static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
130.      memset(s->color_quads, 0, sizeof(s->color_quads));
```

Buffer Overflow boundcpy WrongSizeParam\Path 21:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=33

Status	New

The size of the buffer used by smc_encode_stream in ->, at line 112 of FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that smc_encode_stream passes to ->, at line 112 of FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c, to overwrite the target buffer.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Line	131	131
Object	->	->

Code Snippet
File Name FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Method static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
131.      memset(s->color_octets, 0, sizeof(s->color_octets));
```

Buffer Overflow boundcpy WrongSizeParam\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=34
Status	New

The size of the buffer used by smc_encode_stream in ->, at line 112 of ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that smc_encode_stream passes to ->, at line 112 of ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c, to overwrite the target buffer.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c
Line	129	129
Object	->	->

Code Snippet
File Name ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c
Method static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
129.      memset(s->color_pairs, 0, sizeof(s->color_pairs));
```

Buffer Overflow boundcpy WrongSizeParam\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12

	&pathid=35
Status	New

The size of the buffer used by smc_encode_stream in ->, at line 112 of ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that smc_encode_stream passes to ->, at line 112 of ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c, to overwrite the target buffer.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c
Line	130	130
Object	->	->

Code Snippet
File Name ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c
Method static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
130.      memset(s->color_quads, 0, sizeof(s->color_quads));
```

Buffer Overflow boundcpy WrongSizeParam\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=36
Status	New

The size of the buffer used by smc_encode_stream in ->, at line 112 of ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that smc_encode_stream passes to ->, at line 112 of ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c, to overwrite the target buffer.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c
Line	131	131
Object	->	->

Code Snippet
File Name ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c
Method static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
131.      memset(s->color_octets, 0, sizeof(s->color_octets));
```

Buffer Overflow boundcpy WrongSizeParam\Path 25:

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

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=37
Status	New

The size of the buffer used by remove_short_at_index in H264Picture, at line 514 of ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that remove_short_at_index passes to H264Picture, at line 514 of ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c, to overwrite the target buffer.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Line	520	520
Object	H264Picture	H264Picture

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Method static void remove_short_at_index(H264Context *h, int i)

```
....
520.                    (h->short_ref_count - i) * sizeof(H264Picture*));
```

Buffer Overflow boundcpy WrongSizeParam\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=38
Status	New

The size of the buffer used by ff_h264_execute_ref_pic_marking in h, at line 610 of ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ff_h264_execute_ref_pic_marking passes to h, at line 610 of ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c, to overwrite the target buffer.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Line	764	764
Object	h	h

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Method int ff_h264_execute_ref_pic_marking(H264Context *h)

```
....
764.                            h->short_ref_count *
sizeof(H264Picture*));
```

Buffer Overflow boundcpy WrongSizeParam\Path 27:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=39
Status	New

The size of the buffer used by ff_h264_execute_ref_pic_marking in H264Picture, at line 610 of ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ff_h264_execute_ref_pic_marking passes to H264Picture, at line 610 of ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c, to overwrite the target buffer.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Line	764	764
Object	H264Picture	H264Picture

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Method int ff_h264_execute_ref_pic_marking(H264Context *h)

```
....
764.                            h->short_ref_count *
sizeof(H264Picture*));
```

Buffer Overflow boundcpy WrongSizeParam\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=40
Status	New

The size of the buffer used by h264_initialise_ref_list in H264Ref, at line 135 of ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that h264_initialise_ref_list passes to H264Ref, at line 135 of ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c, to overwrite the target buffer.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Line	163	163
Object	H264Ref	H264Ref

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Method static void h264_initialise_ref_list(H264Context *h, H264SliceContext *sl)

```
....
163.                    memset(&sl->ref_list[list][len], 0,
sizeof(H264Ref) * (sl->ref_count[list] - len));
```

Buffer Overflow boundcpy WrongSizeParam\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=41
Status	New

The size of the buffer used by h264_initialise_ref_list in H264Ref, at line 135 of ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that h264_initialise_ref_list passes to H264Ref, at line 135 of ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c, to overwrite the target buffer.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Line	184	184
Object	H264Ref	H264Ref

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Method static void h264_initialise_ref_list(H264Context *h, H264SliceContext *sl)

```
....
184.              memset(&sl->ref_list[0][len], 0, sizeof(H264Ref) *
(sl->ref_count[0] - len));
```

Buffer Overflow boundcpy WrongSizeParam\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=42
Status	New

The size of the buffer used by smc_encode_frame in AVPALETTE_SIZE, at line 489 of FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that smc_encode_frame passes to AVPALETTE_SIZE, at line 489 of FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c, to overwrite the target buffer.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c
Line	516	516
Object	AVPALETTE_SIZE	AVPALETTE_SIZE

Code Snippet
File Name FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c
Method static int smc_encode_frame(AVCodecContext *avctx, AVPacket *pkt,

```
....
516.      memcpy(pal, frame->data[1], AVPALETTE_SIZE);
```

Buffer Overflow boundcpy WrongSizeParam\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=43
Status	New

The size of the buffer used by smc_encode_frame in AVPALETTE_SIZE, at line 489 of FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that smc_encode_frame passes to AVPALETTE_SIZE, at line 489 of FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c, to overwrite the target buffer.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Line	516	516
Object	AVPALETTE_SIZE	AVPALETTE_SIZE

Code Snippet
File Name FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Method static int smc_encode_frame(AVCodecContext *avctx, AVPacket *pkt,

```
....
516.      memcpy(pal, frame->data[1], AVPALETTE_SIZE);
```

Buffer Overflow boundcpy WrongSizeParam\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=44
Status	New

The size of the buffer used by smc_encode_frame in AVPALETTE_SIZE, at line 489 of ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that smc_encode_frame passes to AVPALETTE_SIZE, at line 489 of ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c, to overwrite the target buffer.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c
Line	516	516
Object	AVPALETTE_SIZE	AVPALETTE_SIZE

Code Snippet
File Name ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c

Method	static int smc_encode_frame(AVCodecContext *avctx, AVPacket *pkt,

```
....
516.      memcpy(pal, frame->data[1], AVPALETTE_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-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=127
Status	New

The variable declared in hw_type at ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c in line 138 is not initialized when it is used by hw_type at ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c in line 138.

	Source	Destination
File	ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c
Line	142	169
Object	hw_type	hw_type

Code Snippet	
File Name	ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c
Method	AVBufferRef *av_hwdevice_ctx_alloc(enum AVHWDeviceType type)

```
....
142.      const HWContextType *hw_type = NULL;
....
169.          ctx->hwctx = av_mallocz(hw_type->device_hwctx_size);
```

Use of Zero Initialized Pointer\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=128
Status	New

The variable declared in hw_type at ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c in line 138 is not initialized when it is used by hw_type at ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c in line 138.

	Source	Destination

File	ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c
Line	142	168
Object	hw_type	hw_type

Code Snippet
File Name ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c
Method AVBufferRef *av_hwdevice_ctx_alloc(enum AVHWDeviceType type)

```
....
142.      const HWContextType *hw_type = NULL;
....
168.      if (hw_type->device_hwctx_size) {
```

Use of Zero Initialized Pointer\Path 3:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=129
Status New

The variable declared in hw_type at ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c in line 138 is not initialized when it is used by hw_type at ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c in line 138.

	Source	Destination
File	ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c
Line	142	163
Object	hw_type	hw_type

Code Snippet
File Name ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c
Method AVBufferRef *av_hwdevice_ctx_alloc(enum AVHWDeviceType type)

```
....
142.      const HWContextType *hw_type = NULL;
....
163.          ctx->internal->priv = av_mallocz(hw_type-
>device_priv_size);
```

Use of Zero Initialized Pointer\Path 4:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=130
Status New

The variable declared in hw_type at ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c in line 138 is not initialized when it is used by hw_type at ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c in line 138.

	Source	Destination
File	ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c
Line	142	162
Object	hw_type	hw_type

Code Snippet
File Name ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c
Method AVBufferRef *av_hwdevice_ctx_alloc(enum AVHWDeviceType type)

```
....
142.      const HWContextType *hw_type = NULL;
....
162.      if (hw_type->device_priv_size) {
```

Use of Zero Initialized Pointer\Path 5:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=131
Status New

The variable declared in hw_type at ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c in line 142 is not initialized when it is used by hw_type at ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c in line 142.

	Source	Destination
File	ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c
Line	146	173
Object	hw_type	hw_type

Code Snippet
File Name ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c
Method AVBufferRef *av_hwdevice_ctx_alloc(enum AVHWDeviceType type)

```
....
146.      const HWContextType *hw_type = NULL;
....
173.          ctx->hwctx = av_mallocz(hw_type->device_hwctx_size);
```

Use of Zero Initialized Pointer\Path 6:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=132

Status	New

The variable declared in hw_type at ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c in line 142 is not initialized when it is used by hw_type at ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c in line 142.

	Source	Destination
File	ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c
Line	146	172
Object	hw_type	hw_type

Code Snippet
File Name ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c
Method AVBufferRef *av_hwdevice_ctx_alloc(enum AVHWDeviceType type)

```
....
146.      const HWContextType *hw_type = NULL;
....
172.      if (hw_type->device_hwctx_size) {
```

Use of Zero Initialized Pointer\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=133
Status	New

The variable declared in hw_type at ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c in line 142 is not initialized when it is used by hw_type at ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c in line 142.

	Source	Destination
File	ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c
Line	146	167
Object	hw_type	hw_type

Code Snippet
File Name ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c
Method AVBufferRef *av_hwdevice_ctx_alloc(enum AVHWDeviceType type)

```
....
146.      const HWContextType *hw_type = NULL;
....
167.          ctx->internal->priv = av_mallocz(hw_type-
>device_priv_size);
```

Use of Zero Initialized Pointer\Path 8:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=134
Status	New

The variable declared in hw_type at ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c in line 142 is not initialized when it is used by hw_type at ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c in line 142.

	Source	Destination
File	ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c
Line	146	166
Object	hw_type	hw_type

Code Snippet
File Name ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c
Method AVBufferRef *av_hwdevice_ctx_alloc(enum AVHWDeviceType type)

```
....
146.      const HWContextType *hw_type = NULL;
....
166.      if (hw_type->device_priv_size) {
```

Use of Zero Initialized Pointer\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=135
Status	New

The variable declared in hw_type at ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c in line 142 is not initialized when it is used by hw_type at ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c in line 142.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c
Line	146	173
Object	hw_type	hw_type

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c
Method AVBufferRef *av_hwdevice_ctx_alloc(enum AVHWDeviceType type)

```
....
146.      const HWContextType *hw_type = NULL;
....
173.          ctx->hwctx = av_mallocz(hw_type->device_hwctx_size);
```

Use of Zero Initialized Pointer\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=136
Status	New

The variable declared in hw_type at ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c in line 142 is not initialized when it is used by hw_type at ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c in line 142.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c
Line	146	172
Object	hw_type	hw_type

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c
Method AVBufferRef *av_hwdevice_ctx_alloc(enum AVHWDeviceType type)

```
....
146.      const HWContextType *hw_type = NULL;
....
172.      if (hw_type->device_hwctx_size) {
```

Use of Zero Initialized Pointer\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=137
Status	New

The variable declared in hw_type at ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c in line 142 is not initialized when it is used by hw_type at ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c in line 142.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c
Line	146	167
Object	hw_type	hw_type

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c
Method AVBufferRef *av_hwdevice_ctx_alloc(enum AVHWDeviceType type)

```
....
146.      const HWContextType *hw_type = NULL;
....
167.          ctx->internal->priv = av_mallocz(hw_type-
>device_priv_size);
```

Use of Zero Initialized Pointer\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=138
Status	New

The variable declared in hw_type at ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c in line 142 is not initialized when it is used by hw_type at ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c in line 142.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c
Line	146	166
Object	hw_type	hw_type

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c
Method AVBufferRef *av_hwdevice_ctx_alloc(enum AVHWDeviceType type)

```
....
146.      const HWContextType *hw_type = NULL;
....
166.      if (hw_type->device_priv_size) {
```

Use of Zero Initialized Pointer\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=139
Status	New

The variable declared in hw_type at FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c in line 143 is not initialized when it is used by hw_type at FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c in line 143.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c	FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c
Line	147	174
Object	hw_type	hw_type

Code Snippet	
File Name	FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c
Method	AVBufferRef *av_hwdevice_ctx_alloc(enum AVHWDeviceType type)

```
....
147.      const HWContextType *hw_type = NULL;
....
174.          ctx->hwctx = av_mallocz(hw_type->device_hwctx_size);
```

Use of Zero Initialized Pointer\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=140
Status	New

The variable declared in hw_type at FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c in line 143 is not initialized when it is used by hw_type at FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c in line 143.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c	FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c
Line	147	173
Object	hw_type	hw_type

Code Snippet	
File Name	FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c
Method	AVBufferRef *av_hwdevice_ctx_alloc(enum AVHWDeviceType type)

```
....
147.      const HWContextType *hw_type = NULL;
....
173.      if (hw_type->device_hwctx_size) {
```

Use of Zero Initialized Pointer\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=141
Status	New

The variable declared in hw_type at FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c in line 143 is not initialized when it is used by hw_type at FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c in line 143.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c	FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c
Line	147	168

Object	hw_type	hw_type

Code Snippet
File Name FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c
Method AVBufferRef *av_hwdevice_ctx_alloc(enum AVHWDeviceType type)

```
....
147.      const HWContextType *hw_type = NULL;
....
168.          ctx->internal->priv = av_mallocz(hw_type-
>device_priv_size);
```

Use of Zero Initialized Pointer\Path 16:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=142
Status New

The variable declared in hw_type at FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c in line 143 is not initialized when it is used by hw_type at FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c in line 143.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c	FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c
Line	147	167
Object	hw_type	hw_type

Code Snippet
File Name FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c
Method AVBufferRef *av_hwdevice_ctx_alloc(enum AVHWDeviceType type)

```
....
147.      const HWContextType *hw_type = NULL;
....
167.      if (hw_type->device_priv_size) {
```

Use of Zero Initialized Pointer\Path 17:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=143
Status New

The variable declared in hw_type at ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c in line 143 is not initialized when it is used by hw_type at ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c in line 143.

	Source	Destination

File	ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c
Line	147	174
Object	hw_type	hw_type

Code Snippet
File Name ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c
Method AVBufferRef *av_hwdevice_ctx_alloc(enum AVHWDeviceType type)

```
....
147.      const HWContextType *hw_type = NULL;
....
174.          ctx->hwctx = av_mallocz(hw_type->device_hwctx_size);
```

Use of Zero Initialized Pointer\Path 18:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=144
Status New

The variable declared in hw_type at ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c in line 143 is not initialized when it is used by hw_type at ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c in line 143.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c
Line	147	173
Object	hw_type	hw_type

Code Snippet
File Name ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c
Method AVBufferRef *av_hwdevice_ctx_alloc(enum AVHWDeviceType type)

```
....
147.      const HWContextType *hw_type = NULL;
....
173.      if (hw_type->device_hwctx_size) {
```

Use of Zero Initialized Pointer\Path 19:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=145
Status New

The variable declared in hw_type at ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c in line 143 is not initialized when it is used by hw_type at ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c in line 143.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c
Line	147	168
Object	hw_type	hw_type

Code Snippet
File Name ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c
Method AVBufferRef *av_hwdevice_ctx_alloc(enum AVHWDeviceType type)

```
....
147.      const HWContextType *hw_type = NULL;
....
168.          ctx->internal->priv = av_mallocz(hw_type-
>device_priv_size);
```

Use of Zero Initialized Pointer\Path 20:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=146
Status New

The variable declared in hw_type at ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c in line 143 is not initialized when it is used by hw_type at ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c in line 143.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c
Line	147	167
Object	hw_type	hw_type

Code Snippet
File Name ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c
Method AVBufferRef *av_hwdevice_ctx_alloc(enum AVHWDeviceType type)

```
....
147.      const HWContextType *hw_type = NULL;
....
167.      if (hw_type->device_priv_size) {
```

MemoryFree on StackVariable

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

Description

MemoryFree on StackVariable\Path 1:

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

	PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=45
Status	New

Calling free() (line 118) on a variable that was not dynamically allocated (line 118) in file ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c may result with a crash.

	Source	Destination
File	ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c
Line	128	128
Object	ctx	ctx

Code Snippet
File Name ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c
Method static void hwdevice_ctx_free(void *opaque, uint8_t *data)

```
....
128.            ctx->free(ctx);
```

MemoryFree on StackVariable\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=46
Status	New

Calling free() (line 220) on a variable that was not dynamically allocated (line 220) in file ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c may result with a crash.

	Source	Destination
File	ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c
Line	231	231
Object	ctx	ctx

Code Snippet
File Name ffmpeg@@ffmpeg-n4.1.7-CVE-2024-31578-TP.c
Method static void hwframe_ctx_free(void *opaque, uint8_t *data)

```
....
231.            ctx->free(ctx);
```

MemoryFree on StackVariable\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12

	&pathid=47
Status	New

Calling free() (line 122) on a variable that was not dynamically allocated (line 122) in file ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c may result with a crash.

	Source	Destination
File	ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c
Line	132	132
Object	ctx	ctx

Code Snippet
File Name ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c
Method static void hwdevice_ctx_free(void *opaque, uint8_t *data)

```
....
132.            ctx->free(ctx);
```

MemoryFree on StackVariable\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=48
Status	New

Calling free() (line 224) on a variable that was not dynamically allocated (line 224) in file ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c may result with a crash.

	Source	Destination
File	ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c
Line	235	235
Object	ctx	ctx

Code Snippet
File Name ffmpeg@@ffmpeg-n4.3.2-CVE-2024-31578-TP.c
Method static void hwframe_ctx_free(void *opaque, uint8_t *data)

```
....
235.            ctx->free(ctx);
```

MemoryFree on StackVariable\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=49

Status	New

Calling free() (line 122) on a variable that was not dynamically allocated (line 122) in file ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c may result with a crash.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c
Line	132	132
Object	ctx	ctx

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c
Method static void hwdevice_ctx_free(void *opaque, uint8_t *data)

```
....
132.            ctx->free(ctx);
```

MemoryFree on StackVariable\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=50
Status	New

Calling free() (line 224) on a variable that was not dynamically allocated (line 224) in file ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c may result with a crash.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c
Line	235	235
Object	ctx	ctx

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2024-31578-TP.c
Method static void hwframe_ctx_free(void *opaque, uint8_t *data)

```
....
235.            ctx->free(ctx);
```

MemoryFree on StackVariable\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=51
Status	New

Calling free() (line 123) on a variable that was not dynamically allocated (line 123) in file FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c may result with a crash.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c	FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c
Line	133	133
Object	ctx	ctx

Code Snippet
File Name FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c
Method static void hwdevice_ctx_free(void *opaque, uint8_t *data)

```
....
133.            ctx->free(ctx);
```

MemoryFree on StackVariable\Path 8:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=52
Status New

Calling free() (line 225) on a variable that was not dynamically allocated (line 225) in file FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c may result with a crash.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c	FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c
Line	236	236
Object	ctx	ctx

Code Snippet
File Name FFmpeg@@FFmpeg-n5.1.1-CVE-2024-31578-TP.c
Method static void hwframe_ctx_free(void *opaque, uint8_t *data)

```
....
236.            ctx->free(ctx);
```

MemoryFree on StackVariable\Path 9:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=53
Status New

Calling free() (line 123) on a variable that was not dynamically allocated (line 123) in file ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c may result with a crash.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c
Line	133	133
Object	ctx	ctx

Code Snippet
File Name ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c
Method static void hwdevice_ctx_free(void *opaque, uint8_t *data)

```
....
133.            ctx->free(ctx);
```

MemoryFree on StackVariable\Path 10:

Severity Medium
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=54
Status New

Calling free() (line 225) on a variable that was not dynamically allocated (line 225) in file ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c may result with a crash.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c	ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c
Line	236	236
Object	ctx	ctx

Code Snippet
File Name ffmpeg@@ffmpeg-n5.1.3-CVE-2024-31578-TP.c
Method static void hwframe_ctx_free(void *opaque, uint8_t *data)

```
....
236.            ctx->free(ctx);
```

Heap Inspection

Query Path:
CPP\Cx\CPP Medium Threat\Heap Inspection Version:1

Categories

OWASP Top 10 2013: A6-Sensitive Data Exposure
FISMA 2014: Media Protection
NIST SP 800-53: SC-4 Information in Shared Resources (P1)
OWASP Top 10 2017: A3-Sensitive Data Exposure

Description

Heap Inspection\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=117
Status	New

Method SSLSocket::passwdCallback at line 100 of facebook@@hhvm-HHVM-4.101.0-CVE-2022-36937-TP.c defines passphrase, which is designated to contain user passwords. However, while plaintext passwords are later assigned to passphrase, this variable is never cleared from memory.

	Source	Destination
File	facebook@@hhvm-HHVM-4.101.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.101.0-CVE-2022-36937-TP.c
Line	103	103
Object	passphrase	passphrase

Code Snippet

File Name facebook@@hhvm-HHVM-4.101.0-CVE-2022-36937-TP.c

Method int SSLSocket::passwdCallback(char* buf, int num, int /*verify*/, void* data) {

```
....
103.    String passphrase = stream->m_context[s_passphrase].toString();
```

Heap Inspection\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=118
Status	New

Method SSLSocket::passwdCallback at line 100 of facebook@@hhvm-HHVM-4.115.0-CVE-2022-36937-TP.c defines passphrase, which is designated to contain user passwords. However, while plaintext passwords are later assigned to passphrase, this variable is never cleared from memory.

	Source	Destination
File	facebook@@hhvm-HHVM-4.115.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.115.0-CVE-2022-36937-TP.c
Line	103	103
Object	passphrase	passphrase

Code Snippet

File Name facebook@@hhvm-HHVM-4.115.0-CVE-2022-36937-TP.c

Method int SSLSocket::passwdCallback(char* buf, int num, int /*verify*/, void* data) {

```
....
103.    String passphrase = stream->m_context[s_passphrase].toString();
```

Heap Inspection\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=119
Status	New

Method SSLSocket::passwdCallback at line 100 of facebook@@hhvm-HHVM-4.147.0-CVE-2022-36937-TP.c defines passphrase, which is designated to contain user passwords. However, while plaintext passwords are later assigned to passphrase, this variable is never cleared from memory.

	Source	Destination
File	facebook@@hhvm-HHVM-4.147.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.147.0-CVE-2022-36937-TP.c
Line	103	103
Object	passphrase	passphrase

Code Snippet
File Name facebook@@hhvm-HHVM-4.147.0-CVE-2022-36937-TP.c
Method int SSLSocket::passwdCallback(char* buf, int num, int /*verify*/, void* data) {

```
....
103.    String passphrase = stream->m_context[s_passphrase].toString();
```

Heap Inspection\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=120
Status	New

Method SSLSocket::passwdCallback at line 100 of facebook@@hhvm-HHVM-4.167.0-CVE-2022-36937-TP.c defines passphrase, which is designated to contain user passwords. However, while plaintext passwords are later assigned to passphrase, this variable is never cleared from memory.

	Source	Destination
File	facebook@@hhvm-HHVM-4.167.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.167.0-CVE-2022-36937-TP.c
Line	103	103
Object	passphrase	passphrase

Code Snippet
File Name facebook@@hhvm-HHVM-4.167.0-CVE-2022-36937-TP.c
Method int SSLSocket::passwdCallback(char* buf, int num, int /*verify*/, void* data) {

```
....
103.    String passphrase = stream->m_context[s_passphrase].toString();
```

Heap Inspection\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=121
Status	New

Method SSLSocket::passwdCallback at line 100 of facebook@@hhvm-HHVM-4.45.0-CVE-2022-36937-TP.c defines passphrase, which is designated to contain user passwords. However, while plaintext passwords are later assigned to passphrase, this variable is never cleared from memory.

	Source	Destination
File	facebook@@hhvm-HHVM-4.45.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.45.0-CVE-2022-36937-TP.c
Line	103	103
Object	passphrase	passphrase

Code Snippet
File Name facebook@@hhvm-HHVM-4.45.0-CVE-2022-36937-TP.c
Method int SSLSocket::passwdCallback(char* buf, int num, int /*verify*/, void* data) {

```
....
103.    String passphrase = stream->m_context[s_passphrase].toString();
```

Heap Inspection\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=122
Status	New

Method SSLSocket::passwdCallback at line 100 of facebook@@hhvm-HHVM-4.58.1-CVE-2022-36937-TP.c defines passphrase, which is designated to contain user passwords. However, while plaintext passwords are later assigned to passphrase, this variable is never cleared from memory.

	Source	Destination
File	facebook@@hhvm-HHVM-4.58.1-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.58.1-CVE-2022-36937-TP.c
Line	103	103
Object	passphrase	passphrase

Code Snippet
File Name facebook@@hhvm-HHVM-4.58.1-CVE-2022-36937-TP.c
Method int SSLSocket::passwdCallback(char* buf, int num, int /*verify*/, void* data) {

```
....
103.    String passphrase = stream->m_context[s_passphrase].toString();
```

Heap Inspection\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=123
Status	New

Method SSLSocket::passwdCallback at line 100 of facebook@@hhvm-HHVM-4.73.0-CVE-2022-36937-TP.c defines passphrase, which is designated to contain user passwords. However, while plaintext passwords are later assigned to passphrase, this variable is never cleared from memory.

	Source	Destination
File	facebook@@hhvm-HHVM-4.73.0-CVE-2022-36937-TP.c	facebook@@hhvm-HHVM-4.73.0-CVE-2022-36937-TP.c
Line	103	103
Object	passphrase	passphrase

Code Snippet
File Name facebook@@hhvm-HHVM-4.73.0-CVE-2022-36937-TP.c
Method int SSLSocket::passwdCallback(char* buf, int num, int /*verify*/, void* data) {

```
....
103.    String passphrase = stream->m_context[s_passphrase].toString();
```

Heap Inspection\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=124
Status	New

Method SSLSocket::passwdCallback at line 100 of facebook@@hhvm-nightly-2020.12.10-CVE-2022-36937-TP.c defines passphrase, which is designated to contain user passwords. However, while plaintext passwords are later assigned to passphrase, this variable is never cleared from memory.

	Source	Destination
File	facebook@@hhvm-nightly-2020.12.10-CVE-2022-36937-TP.c	facebook@@hhvm-nightly-2020.12.10-CVE-2022-36937-TP.c
Line	103	103
Object	passphrase	passphrase

Code Snippet
File Name facebook@@hhvm-nightly-2020.12.10-CVE-2022-36937-TP.c
Method int SSLSocket::passwdCallback(char* buf, int num, int /*verify*/, void* data) {

```
....
103.    String passphrase = stream->m_context[s_passphrase].toString();
```

Heap Inspection\Path 9:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=125
Status	New

Method SSLSocket::passwdCallback at line 100 of facebook@@hhvm-nightly-2021.10.10-CVE-2022-36937-FP.c defines passphrase, which is designated to contain user passwords. However, while plaintext passwords are later assigned to passphrase, this variable is never cleared from memory.

	Source	Destination
File	facebook@@hhvm-nightly-2021.10.10-CVE-2022-36937-FP.c	facebook@@hhvm-nightly-2021.10.10-CVE-2022-36937-FP.c
Line	103	103
Object	passphrase	passphrase

Code Snippet
File Name facebook@@hhvm-nightly-2021.10.10-CVE-2022-36937-FP.c
Method int SSLSocket::passwdCallback(char* buf, int num, int /*verify*/, void* data) {

```
....
103.    String passphrase = stream->m_context[s_passphrase].toString();
```

Heap Inspection\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=126
Status	New

Method SSLSocket::passwdCallback at line 100 of facebook@@hhvm-nightly-2022.11.25-CVE-2022-36937-FP.c defines passphrase, which is designated to contain user passwords. However, while plaintext passwords are later assigned to passphrase, this variable is never cleared from memory.

	Source	Destination
File	facebook@@hhvm-nightly-2022.11.25-CVE-2022-36937-FP.c	facebook@@hhvm-nightly-2022.11.25-CVE-2022-36937-FP.c
Line	103	103
Object	passphrase	passphrase

Code Snippet
File Name facebook@@hhvm-nightly-2022.11.25-CVE-2022-36937-FP.c
Method int SSLSocket::passwdCallback(char* buf, int num, int /*verify*/, void* data) {

```
....
103.    String passphrase = stream->m_context[s_passphrase].toString();
```

Integer Overflow

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

Categories

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

Description

Integer Overflow\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=55
Status	New

A variable of a larger data type, sy, is being assigned to a smaller data type, in 112 of FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c
Line	176	176
Object	sy	sy

Code Snippet
File Name FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c
Method static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
176.                const int sy = offset / stride;
```

Integer Overflow\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=56
Status	New

A variable of a larger data type, sx, is being assigned to a smaller data type, in 112 of FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c
Line	177	177
Object	sx	sx

Code Snippet
File Name FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c

Method	static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
177.                const int sx = offset % stride;
```

Integer Overflow\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=57
Status	New

A variable of a larger data type, sy, is being assigned to a smaller data type, in 112 of FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Line	176	176
Object	sy	sy

Code Snippet	
File Name	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Method	static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
176.                const int sy = offset / stride;
```

Integer Overflow\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=58
Status	New

A variable of a larger data type, sx, is being assigned to a smaller data type, in 112 of FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Line	177	177
Object	sx	sx

Code Snippet	
File Name	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Method	static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
177.              const int sx = offset % stride;
```

Integer Overflow\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=59
Status	New

A variable of a larger data type, sy, is being assigned to a smaller data type, in 112 of ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c
Line	176	176
Object	sy	sy

Code Snippet
File Name ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c
Method static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
176.              const int sy = offset / stride;
```

Integer Overflow\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=60
Status	New

A variable of a larger data type, sx, is being assigned to a smaller data type, in 112 of ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c
Line	177	177
Object	sx	sx

Code Snippet
File Name ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c
Method static void smc_encode_stream(SMCContext *s, const AVFrame *frame,

```
....
177.              const int sx = offset % stride;
```

Unchecked Return Value

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

Categories

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

Description

Unchecked Return Value\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=147
Status	New

The IdentifierHashTable::remove method calls the remove function, at line 138 of facebook@@hermes-v0.6.0-CVE-2022-35289-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	facebook@@hermes-v0.6.0-CVE-2022-35289-TP.c	facebook@@hermes-v0.6.0-CVE-2022-35289-TP.c
Line	140	140
Object	remove	remove

Code Snippet
File Name facebook@@hermes-v0.6.0-CVE-2022-35289-TP.c
Method void IdentifierHashTable::remove(const StringPrimitive *str) {

```
....
140.      remove(str->castToASCIIRef());
```

Unchecked Return Value\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=148
Status	New

The IdentifierHashTable::remove method calls the remove function, at line 138 of facebook@@hermes-v0.6.0-CVE-2022-35289-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination

File	facebook@@hermes-v0.6.0-CVE-2022-35289-TP.c	facebook@@hermes-v0.6.0-CVE-2022-35289-TP.c
Line	142	142
Object	remove	remove

Code Snippet
File Name facebook@@hermes-v0.6.0-CVE-2022-35289-TP.c
Method void IdentifierHashTable::remove(const StringPrimitive *str) {

```
....
142.        remove(str->castToUTF16Ref());
```

Unchecked Return Value\Path 3:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=149
Status New

The IdentifierHashTable::remove method calls the remove function, at line 138 of facebook@@hermes-v0.8.0-CVE-2022-35289-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	facebook@@hermes-v0.8.0-CVE-2022-35289-TP.c	facebook@@hermes-v0.8.0-CVE-2022-35289-TP.c
Line	140	140
Object	remove	remove

Code Snippet
File Name facebook@@hermes-v0.8.0-CVE-2022-35289-TP.c
Method void IdentifierHashTable::remove(const StringPrimitive *str) {

```
....
140.        remove(str->castToASCIIRef());
```

Unchecked Return Value\Path 4:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=150
Status New

The IdentifierHashTable::remove method calls the remove function, at line 138 of facebook@@hermes-v0.8.0-CVE-2022-35289-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	facebook@@hermes-v0.8.0-CVE-2022-35289-TP.c	facebook@@hermes-v0.8.0-CVE-2022-35289-TP.c
Line	142	142
Object	remove	remove

Code Snippet
File Name facebook@@hermes-v0.8.0-CVE-2022-35289-TP.c
Method void IdentifierHashTable::remove(const StringPrimitive *str) {

```
....
142.        remove(str->castToUTF16Ref());
```

Unchecked Return Value\Path 5:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=151
Status New

The IdentifierHashTable::remove method calls the remove function, at line 138 of facebook@@hermes-v0.9.0-CVE-2022-35289-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	facebook@@hermes-v0.9.0-CVE-2022-35289-TP.c	facebook@@hermes-v0.9.0-CVE-2022-35289-TP.c
Line	140	140
Object	remove	remove

Code Snippet
File Name facebook@@hermes-v0.9.0-CVE-2022-35289-TP.c
Method void IdentifierHashTable::remove(const StringPrimitive *str) {

```
....
140.        remove(str->castToASCIIRef());
```

Unchecked Return Value\Path 6:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=152
Status New

The IdentifierHashTable::remove method calls the remove function, at line 138 of facebook@@hermes-v0.9.0-CVE-2022-35289-TP.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	facebook@@hermes-v0.9.0-CVE-2022-35289-TP.c	facebook@@hermes-v0.9.0-CVE-2022-35289-TP.c
Line	142	142
Object	remove	remove

Code Snippet
File Name facebook@@hermes-v0.9.0-CVE-2022-35289-TP.c
Method void IdentifierHashTable::remove(const StringPrimitive *str) {

```
....
142.      remove(str->castToUTF16Ref());
```

Unchecked Array Index

Query Path:
CPP\Cx\CPP Low Visibility\Unchecked Array Index Version:1

Categories

NIST SP 800-53: SI-10 Information Input Validation (P1)

Description

Unchecked Array Index\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=160
Status	New

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Line	117	117
Object	out_i	out_i

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Method static int add_sorted(H264Picture **sorted, H264Picture * const *src,

```
....
117.                  sorted[out_i] = src[i];
```

Unchecked Array Index\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=161
Status	New

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Line	665	665
Object	long_arg	long_arg

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Method int ff_h264_execute_ref_pic_marking(H264Context *h)

```
....
665.                    h->long_ref[ mmco[i].long_arg ] = pic;
```

Unchecked Array Index\Path 3:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=162
Status New

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Line	707	707
Object	long_arg	long_arg

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Method int ff_h264_execute_ref_pic_marking(H264Context *h)

```
....
707.                    h->long_ref[mmco[i].long_arg]           = h-
>cur_pic_ptr;
```

Unchecked Array Index\Path 4:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=163
Status New

	Source	Destination
File	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c	FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c

Line	84	84
Object	n	n

Code Snippet
File Name FFmpeg@@FFmpeg-n5.0.1-CVE-2022-3965-TP.c
Method static int count_distinct_items(const uint8_t *block_values,

```
....
84.             distinct_values[n] = block_values[i];
```

Unchecked Array Index\Path 5:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=164
Status New

	Source	Destination
File	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c	FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Line	84	84
Object	n	n

Code Snippet
File Name FFmpeg@@FFmpeg-n5.1.1-CVE-2022-3965-FP.c
Method static int count_distinct_items(const uint8_t *block_values,

```
....
84.             distinct_values[n] = block_values[i];
```

Unchecked Array Index\Path 6:

Severity Low
Result State To Verify
Online Results http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=165
Status New

	Source	Destination
File	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c	ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c
Line	84	84
Object	n	n

Code Snippet
File Name ffmpeg@@ffmpeg-n5.1.1-CVE-2022-3965-TP.c

Method	static int count_distinct_items(const uint8_t *block_values,

```
....
84.             distinct_values[n] = block_values[i];
```

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-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=157
Status	New

	Source	Destination
File	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Line	875	875
Object	ai	sizeof

Code Snippet	
File Name	facebook@@hhvm-HHVM-4.45.0-CVE-2020-1916-TP.c
Method	char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
875.          !memcmp(ai, yi, sizeof(ai));
```

Sizeof Pointer Argument\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=158
Status	New

	Source	Destination
File	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c
Line	875	875
Object	ai	sizeof

Code Snippet	
File Name	facebook@@hhvm-HHVM-4.58.1-CVE-2020-1916-TP.c
Method	char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
875.          !memcmp(ai, yi, sizeof(ai));
```

Sizeof Pointer Argument\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=159
Status	New

	Source	Destination
File	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c	facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c
Line	875	875
Object	ai	sizeof

Code Snippet
File Name facebook@@hhvm-HHVM-4.73.0-CVE-2020-1916-TP.c
Method char *php_crypt_blowfish_rn(const char *key, const char *setting,

```
....
875.          !memcmp(ai, yi, sizeof(ai));
```

Use of Sizeof On a Pointer Type

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

Description

Use of Sizeof On a Pointer Type\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=153
Status	New

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Line	520	520
Object	sizeof	sizeof

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Method static void remove_short_at_index(H264Context *h, int i)

```
....
520.                    (h->short_ref_count - i) * sizeof(H264Picture*));
```

Use of Sizeof On a Pointer Type\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=154
Status	New

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Line	764	764
Object	sizeof	sizeof

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Method int ff_h264_execute_ref_pic_marking(H264Context *h)

```
....
764.                            h->short_ref_count *
sizeof(H264Picture*));
```

NULL Pointer Dereference

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

Categories

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

Description

NULL Pointer Dereference\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=155
Status	New

The variable declared in null at ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c in line 299 is not initialized when it is used by f at ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c in line 52.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Line	316	55

Object	null	f

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Method int ff_h264_build_ref_list(H264Context *h, H264SliceContext *sl)

```
....
316.              H264Picture *ref = NULL;
```

File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Method static void ref_from_h264pic(H264Ref *dst, H264Picture *src)

```
....
55.      memcpy(dst->linesize, src->f->linesize, sizeof(dst->linesize));
```

NULL Pointer Dereference\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=156
Status	New

The variable declared in null at ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c in line 299 is not initialized when it is used by f at ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c in line 52.

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Line	316	54
Object	null	f

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Method int ff_h264_build_ref_list(H264Context *h, H264SliceContext *sl)

```
....
316.              H264Picture *ref = NULL;
```

File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Method static void ref_from_h264pic(H264Ref *dst, H264Picture *src)

```
....
54.      memcpy(dst->data,     src->f->data,     sizeof(dst->data));
```

Arithmenic Operation On Boolean

Query Path:
CPP\Cx\CPP Low Visibility\Arithmenic Operation On Boolean Version:1

Categories

FISMA 2014: Audit And Accountability
NIST SP 800-53: SC-5 Denial of Service Protection (P1)

Description

Arithmenic Operation On Boolean\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-PTJMSNK3USL/CxWebClient/ViewerMain.aspx?scanid=1000017&projectid=12&pathid=61
Status	New

	Source	Destination
File	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c	ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Line	203	203
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name ffmpeg@@ffmpeg-n5.0.1-CVE-2021-3520-FP.c
Method static void h264_initialise_ref_list(H264Context *h, H264SliceContext *sl)

```
....
203.      for (j = 0; j<1+(sl->slice_type_nos == AV_PICTURE_TYPE_B);
j++) {
```

Buffer Overflow LongString

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));
    }
}
```

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

MemoryFree on StackVariable

Risk

What might happen

Undefined Behavior may result with a crash. Crashes may give an attacker valuable information about the system and the program internals. Furthermore, it may leave unprotected files (e.g memory) that may be exploited.

Cause

How does it happen

Calling free() on a variable that was not dynamically allocated (e.g. malloc) will result with an Undefined Behavior.

General Recommendations

How to avoid it

Use free() only on dynamically allocated variables in order to prevent unexpected behavior from the compiler.

Source Code Examples

CPP

Bad - Calling free() on a static variable

```
void clean_up(){
  char temp[256];
  do_something();
  free(tmp);
  return;
}
```

Good - Calling free() only on variables that were dynamically allocated

```
void clean_up(){
  char *buff;
  buff = (char*) malloc(1024);
  free(buff);
  return;
}
```

Integer Overflow

Risk

What might happen

Assigning large data types into smaller data types, without proper checks and explicit casting, will lead to undefined behavior and unintentional effects, such as data corruption (e.g. value wraparound, wherein maximum values become minimum values); system crashes; infinite loops; logic errors, such as bypassing of security mechanisms; or even buffer overflows leading to arbitrary code execution.

Cause

How does it happen

This flaw can occur when implicitly casting numerical data types of a larger size, into a variable with a data type of a smaller size. This forces the program to discard some bits of information from the number. Depending on how the numerical data types are stored in memory, this is often the bits with the highest value, causing substantial corruption of the stored number. Alternatively, the sign bit of a signed integer could be lost, completely reversing the intention of the number.

General Recommendations

How to avoid it

- Avoid casting larger data types to smaller types.
- Prefer promoting the target variable to a large enough data type.
- If downcasting is necessary, always check that values are valid and in range of the target type, before casting

---

Source Code Examples

CPP

Unsafe Downsize Casting

```
int unsafe_addition(short op1, int op2) {

    // op2 gets forced from int into a short
    short total = op1 + op2;

    return total;
}
```

Safer Use of Proper Data Types

```
int safe_addition(short op1, int op2) {

    // total variable is of type int, the largest type that is needed
    int total = 0;

    // check if total will overflow available integer size
    if (INT_MAX - abs(op2) > op1)
```

```
    {
        total = op1 + op2;
    }
    else
    {
        // instead of overflow, saturate (but this is not always a good thing)
        total = INT_MAX
    }

    return total;
}
```

Dangerous Functions

Risk

What might happen

Use of dangerous functions may expose varying risks associated with each particular function, with potential impact of improper usage of these functions varying significantly. The presence of such functions indicates a flaw in code maintenance policies and adherence to secure coding practices, in a way that has allowed introducing known dangerous code into the application.

Cause

How does it happen

A dangerous function has been identified within the code. Functions are often deemed dangerous to use for numerous reasons, as there are different sets of vulnerabilities associated with usage of such functions. For example, some string copy and concatenation functions are vulnerable to Buffer Overflow, Memory Disclosure, Denial of Service and more. Use of these functions is not recommended.

General Recommendations

How to avoid it

- Deploy a secure and recommended alternative to any functions that were identified as dangerous.
    - If no secure alternative is found, conduct further researching and testing to identify whether current usage successfully sanitizes and verifies values, and thus successfully avoids the use-cases for whom the function is indeed dangerous
- Conduct a periodical review of methods that are in use, to ensure that all external libraries and built-in functions are up-to-date and whose use has not been excluded from best secure coding practices.

Source Code Examples

CPP

Buffer Overflow in gets()

```
int main()
{
	char buf[10];

	printf("Please enter your name: ");
	gets(buf); // veryveryverylongname
	if (buf == ACCEPTED_NAME)
	{
		// Do something
	}
	return 0;
}
```

Safe reading from user

```
int main()
{
	char buf[10];

	printf("Please enter your name: ");
	fgets(buf, sizeof(buf), stdin); //setting the amount of bytes to read
	if (buf == ACCEPTED_NAME)
	{
		//Do something
	}
	return 0;
}
```

Unsafe function for string copy

```
int main(int argc, char* argv[])
{
	char buf[10];
	strcpy(buf, argv[1]); // overflow occurs when len(argv[1]) > 10 bytes

	return 0;
}
```

Safe string copy

```
int main(int argc, char* argv[])
{
	char buf[10];
	strncpy(buf, argv[1], sizeof(buf));
	buf[9]= '\0'; //strncpy doesn't NULL terminates

	return 0;
}
```

Unsafe format string

```
int main(int argc, char* argv[])
{
	printf(argv[1]); // If argv[1] contains a format token, such as %s,%x or %d, will cause
an access violation
	return 0;
}
```

Safe format string

```
int main(int argc, char* argv[])
{
      printf("%s", argv[1]); // Second parameter is not a formattable string
      return 0;
}
```

Heap Inspection

Risk

What might happen

All variables stored by the application in unencrypted memory can potentially be retrieved by an unauthorized user, with privlieged access to the machine. For example, a privileged attacker could attach a debugger to the running process, or retrieve the process's memory from the swapfile or crash dump file.
Once the attacker finds the user passwords in memory, these can be reused to easily impersonate the user to the system.

Cause

How does it happen

String variables are immutable - in other words, once a string variable is assigned, its value cannot be changed or removed. Thus, these strings may remain around in memory, possibly in multiple locations, for an indefinite period of time until the garbage collector happens to remove it. Sensitive data, such as passwords, will remain exposed in memory as plaintext with no control over their lifetime.

General Recommendations

How to avoid it

Generic Guidance:

- Do not store senstiive data, such as passwords or encryption keys, in memory in plaintext, even for a short period of time.
- Prefer to use specialized classes that store encrypted memory.
- Alternatively, store secrets temporarily in mutable data types, such as byte arrays, and then promptly zeroize the memory locations.

Specific Recommendations - Java:

- Instead of storing passwords in immutable strings, prefer to use an encrypted memory object, such as SealedObject.

Specific Recommendations - .NET:

- Instead of storing passwords in immutable strings, prefer to use an encrypted memory object, such as SecureString or ProtectedData.

Source Code Examples

Java

Plaintext Password in Immutable String

```
class Heap_Inspection
{
  private string password;
```

```
  void setPassword()
 {
      password = System.console().readLine("Enter your password: ");
  }
}
```

Password Protected in Memory

```
class Heap_Inspection_Fixed
{
  private SealedObject password;

  void setPassword()
 {
      byte[] sKey = getKeyFromConfig();
      Cipher c = Cipher.getInstance("AES");
      c.init(Cipher.ENCRYPT_MODE, sKey);

      char[] input = System.console().readPassword("Enter your password: ");
      password = new SealedObject(Arrays.asList(input), c);

    //Zero out the possible password, for security.
    Arrays.fill(password, '0');
  }
}
```

CPP

Vulnerable C code

```
/* Vulnerable to heap inspection */

#include <stdio.h>


void somefunc(){
      printf("Yea, I'm just being called for the heap of it..\n");
}

void authfunc(){
        char* password = (char *) malloc(256);
        char ch;
        ssize_t k;
            int i=0;
        while(k = read(0, &ch, 1) > 0)
        {
                if (ch == '\n'){
                        password[i]='\0';
                        break;
                } else{
                        password[i++]=ch;
                        fflush(0);
                }
        }
        printf("Password: %s\n",&password[0]);
}
```

```
int main()
{
    printf("Please enter a password:\n");

      authfunc();
      printf("You can now dump memory to find this password!");
      somefunc();
      gets();

}
```

Safe C code

```
/* Pesumably safe heap */

#include <stdio.h>
#include <string.h>

#define STDIN_FILENO 0

void somefunc(){
        printf("Yea, I'm just being called for the heap of it..\n");
}

void authfunc(){
      char* password = (char*) malloc(256);
      int i=0;
      char ch;
      ssize_t k;
      while(k = read(STDIN_FILENO, &ch, 1) > 0)
      {
            if (ch == '\n'){
                  password[i]='\0';
                  break;
            } else{
                  password[i++]=ch;
                  fflush(0);
            }
      }
      i=0;
      memset(password,'\0',256);
}

int main()
{
      printf("Please enter a password:\n");
      authfunc();
      somefunc();
      char ch;
      while(read(STDIN_FILENO, &ch, 1) > 0)
      {
            if (ch == '\n')
                  break;
      }
}
```

Use of Zero Initialized Pointer

Risk

What might happen

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

Cause

How does it happen

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

General Recommendations

How to avoid it

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

Source Code Examples

CPP

Explicit NULL Dereference

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

Implicit NULL Dereference

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

Java

Explicit Null Dereference

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

Indicator of Poor Code Quality

Weakness ID: 398 *(Weakness Class)* **Status:** Draft

Description

Description Summary

The code has features that do not directly introduce a weakness or vulnerability, but indicate that the product has not been carefully developed or maintained.

Extended Description

Programs are more likely to be secure when good development practices are followed. If a program is complex, difficult to maintain, not portable, or shows evidence of neglect, then there is a higher likelihood that weaknesses are buried in the code.

Time of Introduction

- Architecture and Design
- Implementation

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	18	Source Code	**Development Concepts (primary)699**
ChildOf	Weakness Class	710	Coding Standards Violation	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	107	Struts: Unused Validation Form	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	110	Struts: Validator Without Form Field	**Research Concepts (primary)1000**
ParentOf	Category	399	Resource Management Errors	**Development Concepts (primary)699**
ParentOf	Weakness Base	401	Failure to Release Memory Before Removing Last Reference ('Memory Leak')	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	404	Improper Resource Shutdown or Release	Development Concepts699 **Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Variant	415	Double Free	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	416	Use After Free	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Variant	457	Use of Uninitialized Variable	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	474	Use of Function with Inconsistent Implementations	**Development Concepts (primary)699 Seven Pernicious Kingdoms (primary)700 Research Concepts (primary)1000**
ParentOf	Weakness Base	475	Undefined Behavior for Input to API	**Development Concepts (primary)699 Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	476	NULL Pointer Dereference	**Development Concepts**

				(primary)699 Seven Pernicious Kingdoms (primary)700 Research Concepts (primary)1000
ParentOf	Weakness Base	477	Use of Obsolete Functions	**Development Concepts (primary)699 Seven Pernicious Kingdoms (primary)700 Research Concepts (primary)1000**
ParentOf	Weakness Variant	478	Missing Default Case in Switch Statement	**Development Concepts (primary)699**
ParentOf	Weakness Variant	479	Unsafe Function Call from a Signal Handler	**Development Concepts (primary)699**
ParentOf	Weakness Variant	483	Incorrect Block Delimitation	**Development Concepts (primary)699**
ParentOf	Weakness Base	484	Omitted Break Statement in Switch	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Variant	546	Suspicious Comment	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Weakness Variant	547	Use of Hard-coded, Security-relevant Constants	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Weakness Variant	561	Dead Code	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Weakness Base	562	Return of Stack Variable Address	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Variant	563	Unused Variable	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Category	569	Expression Issues	**Development Concepts (primary)699**
ParentOf	Weakness Variant	585	Empty Synchronized Block	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Weakness Variant	586	Explicit Call to Finalize()	**Development Concepts (primary)699**
ParentOf	Weakness Variant	617	Reachable Assertion	**Development Concepts (primary)699**
ParentOf	Weakness Base	676	Use of Potentially Dangerous Function	**Development Concepts (primary)699 Research Concepts (primary)1000**
MemberOf	View	700	Seven Pernicious Kingdoms	**Seven Pernicious Kingdoms (primary)700**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name

7 Pernicious Kingdoms			Code Quality

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	7 Pernicious Kingdoms		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Description, Relationships, Taxonomy Mappings		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Relationships		
Previous Entry Names			
Change Date	**Previous Entry Name**		
2008-04-11	Code Quality		

BACK TO TOP

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

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

Scanned Languages

Language	Hash Number	Change Date
CPP	4541647240435660	1/6/2025
Common	0105849645654507	1/6/2025