awtk Scan Report

Project Name	awtk
Scan Start	Thursday, June 20, 2024 11:50:15 PM
Preset	Checkmarx Default
Scan Time	00h:05m:47s
Lines Of Code Scanned	28758
Files Scanned	21
Report Creation Time	Friday, June 21, 2024 12:10:08 AM
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004
Team	CxServer
Checkmarx Version	8.7.0
Scan Type	Full
Source Origin	LocalPath
Density	6/1000 (Vulnerabilities/LOC)
Visibility	Public

Filter Settings

Severity

Included: High, Medium, Low, Information

Excluded: None

Result State

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

Excluded: None

Assigned to

Included: All

Categories

Included:

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

Excluded:

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

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

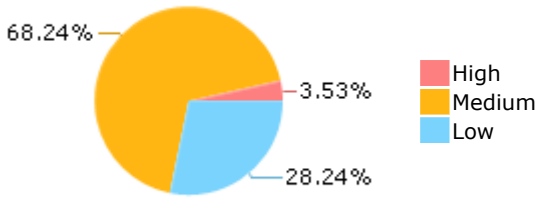
Results Limit

Results limit per query was set to 50

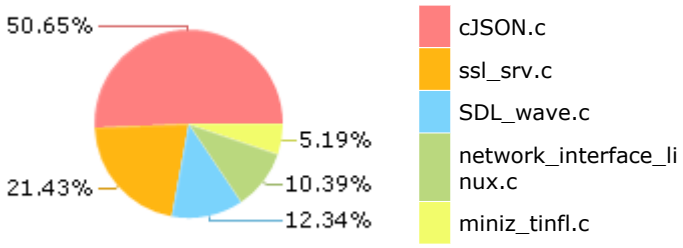
Selected Queries

Selected queries are listed in Result Summary

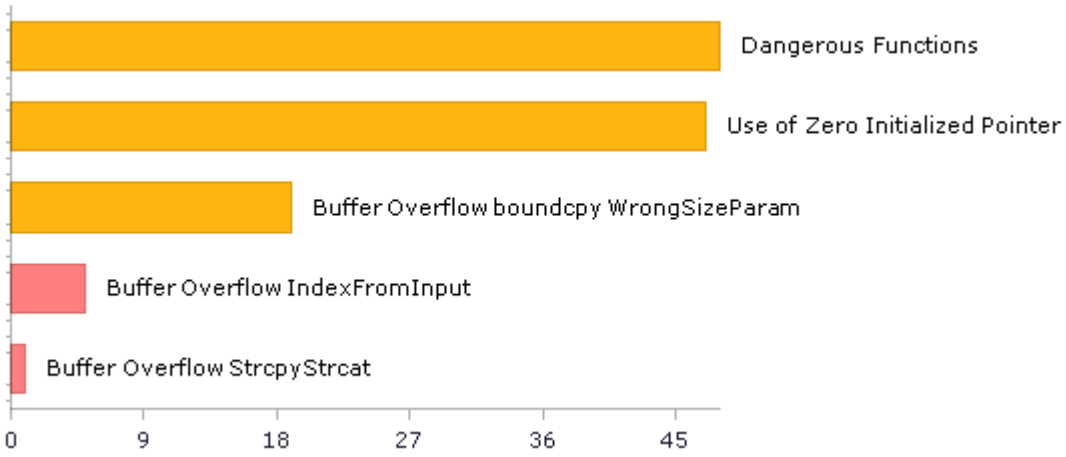
Result Summary



Most Vulnerable Files



Top 5 Vulnerabilities



Scan Summary - OWASP Top 10 2017

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

Category	Threat Agent	Exploitability	Weakness Prevalence	Weakness Detectability	Technical Impact	Business Impact	Issues Found	Best Fix Locations
A1-Injection	App. Specific	EASY	COMMON	EASY	SEVERE	App. Specific	54	30
A2-Broken Authentication	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	2	2
A3-Sensitive Data Exposure	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	App. Specific	1	1
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	48	48
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	1	1
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	48	48
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	1	1
PCI DSS (3.2) - 6.5.2 - Buffer overflows	21	21
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.	0	0
Configuration Management	Organizations must: (i) establish and maintain baseline configurations and inventories of organizational information systems (including hardware, software, firmware, and documentation) throughout the respective system development life cycles; and (ii) establish and enforce security configuration settings for information technology products employed in organizational information systems.	0	0
Identification And Authentication*	Organizations must identify information system users, processes acting on behalf of users, or devices and authenticate (or verify) the identities of those users, processes, or devices, as a prerequisite to allowing access to organizational information systems.	2	2
Media Protection	Organizations must: (i) protect information system media, both paper and digital; (ii) limit access to information on information system media to authorized users; and (iii) sanitize or destroy information system media before disposal or release for reuse.	1	1
System And Communications Protection	Organizations must: (i) monitor, control, and protect organizational communications (i.e., information transmitted or received by organizational information systems) at the external boundaries and key internal boundaries of the information systems; and (ii) employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational information systems.	0	0
System And Information Integrity	Organizations must: (i) identify, report, and correct information and information system flaws in a timely manner; (ii) provide protection from malicious code at appropriate locations within organizational information systems; and (iii) monitor information system security alerts and advisories and take appropriate actions in response.	1	1

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

Scan Summary - NIST SP 800-53

Category	Issues Found	Best Fix Locations
AC-12 Session Termination (P2)	0	0
AC-3 Access Enforcement (P1)	2	2
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)	1	1
SC-5 Denial of Service Protection (P1)*	75	21
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*	2	2
SI-11 Error Handling (P2)*	11	11
SI-15 Information Output Filtering (P0)	0	0
SI-16 Memory Protection (P1)	1	1

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

Scan Summary - 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

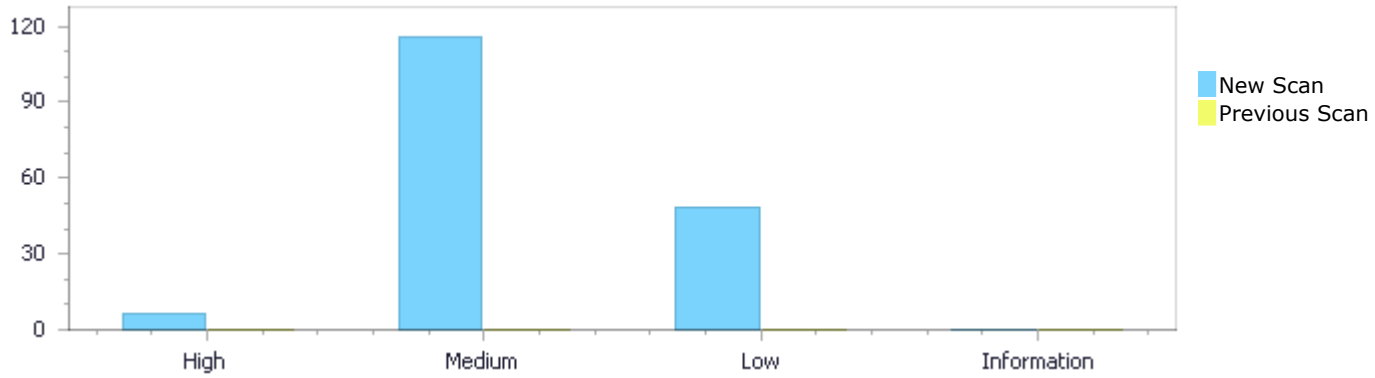
Scan Summary - Custom

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	6	116	48	0	170
Recurrent Issues	0	0	0	0	0
Total	6	116	48	0	170
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	6	116	48	0	170
Urgent	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0
Total	6	116	48	0	170

Result Summary

Vulnerability Type	Occurrences	Severity
Buffer Overflow IndexFromInput	5	High
Buffer Overflow StrcpyStrcat	1	High
Dangerous Functions	48	Medium
Use of Zero Initialized Pointer	47	Medium
Buffer Overflow boundcpy WrongSizeParam	19	Medium

Heap Inspection	1	Medium
Integer Overflow	1	Medium
NULL Pointer Dereference	28	Low
Unchecked Return Value	11	Low
Use of Sizeof On a Pointer Type	3	Low
Improper Resource Access Authorization	2	Low
TOCTOU	2	Low
Potential Off by One Error in Loops	1	Low
Sizeof Pointer Argument	1	Low

10 Most Vulnerable Files

High and Medium Vulnerabilities

File Name	Issues Found
awtk/cJSON.c	65
awtk/ssl_srv.c	33
awtk/miniz_tinfl.c	7
awtk/testgles.c	5
awtk/network_interface_linux.c	3
awtk/edit.c	2
awtk/SDL_pixels.c	2
awtk/SDL_wave.c	2
awtk/date_time.c	1
awtk/harness_argparser.c	1

Scan Results Details

Buffer Overflow IndexFromInput

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

Categories

OWASP Top 10 2017: A1-Injection

Description

Buffer Overflow IndexFromInput\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=21
Status	New

The size of the buffer used by main in i, at line 103 of awtk/testgles.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 103 of awtk/testgles.c, to overwrite the target buffer.

	Source	Destination
File	awtk/testgles.c	awtk/testgles.c
Line	103	141
Object	argv	i

Code Snippet
File Name awtk/testgles.c
Method main(int argc, char *argv[])

```
....
103.  main(int argc, char *argv[])
....
141.                      depth = SDL_atoi(argv[i]);
```

Buffer Overflow IndexFromInput\Path 2:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=22
Status	New

The size of the buffer used by main in i, at line 103 of awtk/testgles.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 103 of awtk/testgles.c, to overwrite the target buffer.

	Source	Destination
File	awtk/testgles.c	awtk/testgles.c

Line	103	138
Object	argv	i

Code Snippet
File Name awtk/testgles.c
Method main(int argc, char *argv[])

```
....
103.  main(int argc, char *argv[])
....
138.                  if (!argv[i]) {
```

Buffer Overflow IndexFromInput\Path 3:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=23
Status	New

The size of the buffer used by main in i, at line 103 of awtk/testgles.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 103 of awtk/testgles.c, to overwrite the target buffer.

	Source	Destination
File	awtk/testgles.c	awtk/testgles.c
Line	103	136
Object	argv	i

Code Snippet
File Name awtk/testgles.c
Method main(int argc, char *argv[])

```
....
103.  main(int argc, char *argv[])
....
136.              } else if (SDL_strcasecmp(argv[i], "--zdepth") == 0) {
```

Buffer Overflow IndexFromInput\Path 4:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=24
Status	New

The size of the buffer used by main in i, at line 103 of awtk/testgles.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 103 of awtk/testgles.c, to overwrite the target buffer.

	Source	Destination

File	awtk/testgles.c	awtk/testgles.c
Line	103	133
Object	argv	i

Code Snippet
File Name awtk/testgles.c
Method main(int argc, char *argv[])

```
....
103.  main(int argc, char *argv[])
....
133.              } else if (SDL_strcasecmp(argv[i], "--accel") == 0) {
```

Buffer Overflow IndexFromInput\Path 5:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=25
Status	New

The size of the buffer used by main in i, at line 103 of awtk/testgles.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 103 of awtk/testgles.c, to overwrite the target buffer.

	Source	Destination
File	awtk/testgles.c	awtk/testgles.c
Line	103	130
Object	argv	i

Code Snippet
File Name awtk/testgles.c
Method main(int argc, char *argv[])

```
....
103.  main(int argc, char *argv[])
....
130.              if (SDL_strcasecmp(argv[i], "--fsaa") == 0) {
```

Buffer Overflow StrcpyStrcat

Query Path:
CPP\Cx\CPP Buffer Overflow\Buffer Overflow StrcpyStrcat 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 StrcpyStrcat\Path 1:

Severity	High

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=20
Status	New

The size of the buffer used by print_string_ptr in output, at line 843 of awtk/cJSON.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that print_string_ptr passes to input, at line 843 of awtk/cJSON.c, to overwrite the target buffer.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	843	865
Object	input	output

Code Snippet
File Name awtk/cJSON.c
Method static cJSON_bool print_string_ptr(const unsigned char * const input, printbuffer * const output_buffer)

```
....
843.  static cJSON_bool print_string_ptr(const unsigned char * const
input, printbuffer * const output_buffer)
....
865.          strcpy((char*)output, "\"\"");
```

Dangerous Functions

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

Categories

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

Description
Dangerous Functions\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=71
Status	New

The dangerous function, memcpy, was found in use at line 159 in awtk/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	175	175
Object	memcpy	memcpy

Code Snippet	
File Name	awtk/cJSON.c
Method	static unsigned char* cJSON_strdup(const unsigned char* string, const internal_hooks * const hooks)

```
....
175.      memcpy(copy, string, length);
```

Dangerous Functions\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=72
Status	New

The dangerous function, memcpy, was found in use at line 383 in awtk/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	460	460
Object	memcpy	memcpy

Code Snippet	
File Name	awtk/cJSON.c
Method	static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
460.              memcpy(newbuffer, p->buffer, p->offset + 1);
```

Dangerous Functions\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=73
Status	New

The dangerous function, memcpy, was found in use at line 843 in awtk/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	906	906
Object	memcpy	memcpy

Code Snippet	
File Name	awtk/cJSON.c

Method	static cJSON_bool print_string_ptr(const unsigned char * const input, printbuffer * const output_buffer)

```
....
906.          memcpy(output + 1, input, output_length);
```

Dangerous Functions\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=74
Status	New

The dangerous function, memcpy, was found in use at line 1103 in awtk/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	1144	1144
Object	memcpy	memcpy

Code Snippet	
File Name	awtk/cJSON.c
Method	static unsigned char *print(const cJSON * const item, cJSON_bool format, const internal_hooks * const hooks)

```
....
1144.          memcpy(printed, buffer->buffer, cjson_min(buffer->length,
buffer->offset + 1));
```

Dangerous Functions\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=75
Status	New

The dangerous function, memcpy, was found in use at line 1283 in awtk/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	1338	1338
Object	memcpy	memcpy

Code Snippet	
File Name	awtk/cJSON.c

Method	static cJSON_bool print_value(const cJSON * const item, printbuffer * const output_buffer)

```
....
1338.              memcpy(output, item->valuestring, raw_length);
```

Dangerous Functions\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=76
Status	New

The dangerous function, memcpy, was found in use at line 1843 in awtk/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	1857	1857
Object	memcpy	memcpy

Code Snippet	
File Name	awtk/cJSON.c
Method	static cJSON *create_reference(const cJSON *item, const internal_hooks * const hooks)

```
....
1857.      memcpy(reference, item, sizeof(cJSON));
```

Dangerous Functions\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=77
Status	New

The dangerous function, memcpy, was found in use at line 174 in awtk/miniz_tinfl.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/miniz_tinfl.c	awtk/miniz_tinfl.c
Line	260	260
Object	memcpy	memcpy

Code Snippet	
File Name	awtk/miniz_tinfl.c

Method	tinfl_status tinfl_decompress(tinfl_decompressor *r, const mz_uint8 *pIn_buf_next, size_t *pIn_buf_size, mz_uint8 *pOut_buf_start, mz_uint8 *pOut_buf_next, size_t *pOut_buf_size, const mz_uint32 decomp_flags)

```
....
260.                   TINFL_MEMCPY(pOut_buf_cur, pIn_buf_cur, n);
```

Dangerous Functions\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=78
Status	New

The dangerous function, memcpy, was found in use at line 174 in awtk/miniz_tinfl.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/miniz_tinfl.c	awtk/miniz_tinfl.c
Line	391	391
Object	memcpy	memcpy

Code Snippet	
File Name	awtk/miniz_tinfl.c
Method	tinfl_status tinfl_decompress(tinfl_decompressor *r, const mz_uint8 *pIn_buf_next, size_t *pIn_buf_size, mz_uint8 *pOut_buf_start, mz_uint8 *pOut_buf_next, size_t *pOut_buf_size, const mz_uint32 decomp_flags)

```
....
391.                       TINFL_MEMCPY(r->m_tables[0].m_code_size, r-
>m_len_codes, r->m_table_sizes[0]);
```

Dangerous Functions\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=79
Status	New

The dangerous function, memcpy, was found in use at line 174 in awtk/miniz_tinfl.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/miniz_tinfl.c	awtk/miniz_tinfl.c
Line	392	392
Object	memcpy	memcpy

Code Snippet	
File Name	awtk/miniz_tinfl.c
Method	tinfl_status tinfl_decompress(tinfl_decompressor *r, const mz_uint8 *pIn_buf_next, size_t *pIn_buf_size, mz_uint8 *pOut_buf_start, mz_uint8 *pOut_buf_next, size_t *pOut_buf_size, const mz_uint32 decomp_flags)

```
....
392.                     TINFL_MEMCPY(r->m_tables[1].m_code_size, r-
>m_len_codes + r->m_table_sizes[0], r->m_table_sizes[1]);
```

Dangerous Functions\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=80
Status	New

The dangerous function, memcpy, was found in use at line 174 in awtk/miniz_tinfl.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/miniz_tinfl.c	awtk/miniz_tinfl.c
Line	527	527
Object	memcpy	memcpy

Code Snippet	
File Name	awtk/miniz_tinfl.c
Method	tinfl_status tinfl_decompress(tinfl_decompressor *r, const mz_uint8 *pIn_buf_next, size_t *pIn_buf_size, mz_uint8 *pOut_buf_start, mz_uint8 *pOut_buf_next, size_t *pOut_buf_size, const mz_uint32 decomp_flags)

```
....
527.                                     memcpy(pOut_buf_cur, pSrc,
sizeof(mz_uint32)*2);
```

Dangerous Functions\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=81
Status	New

The dangerous function, memcpy, was found in use at line 86 in awtk/network_interface_linux.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/network_interface_linux.c	awtk/network_interface_linux.c
Line	100	100

Object	memcpy	memcpy

Code Snippet
File Name awtk/network_interface_linux.c
Method static char* network_interface_linux_get_macaddr(network_interface_t* interface) {

```
....
100.    memcpy(m, ifr.ifr_hwaddr.sa_data, 6);
```

Dangerous Functions\Path 12:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=82
Status New

The dangerous function, memcpy, was found in use at line 49 in awtk/ssl_srv.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	61	61
Object	memcpy	memcpy

Code Snippet
File Name awtk/ssl_srv.c
Method int mbedtls_ssl_set_client_transport_id(mbedtls_ssl_context *ssl,

```
....
61.      memcpy( ssl->cli_id, info, ilen );
```

Dangerous Functions\Path 13:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=83
Status New

The dangerous function, memcpy, was found in use at line 474 in awtk/ssl_srv.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	536	536
Object	memcpy	memcpy

Code Snippet
File Name awtk/ssl_srv.c
Method static int ssl_parse_cid_ext(mbedtls_ssl_context *ssl,

```
....
536.      memcpy( ssl->handshake->peer_cid, buf, peer_cid_len );
```

Dangerous Functions\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=84
Status	New

The dangerous function, memcpy, was found in use at line 618 in awtk/ssl_srv.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	672	672
Object	memcpy	memcpy

Code Snippet
File Name awtk/ssl_srv.c
Method static int ssl_parse_session_ticket_ext(mbedtls_ssl_context *ssl,

```
....
672.      memcpy( &session.id, ssl->session_negotiate->id,
session.id_len );
```

Dangerous Functions\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=85
Status	New

The dangerous function, memcpy, was found in use at line 618 in awtk/ssl_srv.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	675	675
Object	memcpy	memcpy

Code Snippet

File Name	awtk/ssl_srv.c
Method	static int ssl_parse_session_ticket_ext(mbedtls_ssl_context *ssl,

```
....
675.      memcpy( ssl->session_negotiate, &session, sizeof(
mbedtls_ssl_session ) );
```

Dangerous Functions\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=86
Status	New

The dangerous function, memcpy, was found in use at line 780 in awtk/ssl_srv.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	889	889
Object	memcpy	memcpy

Code Snippet	
File Name	awtk/ssl_srv.c
Method	static int ssl_parse_use_srtp_ext(mbedtls_ssl_context *ssl,

```
....
889.          memcpy( ssl->dtls_srtp_info.mki_value, buf, mki_length );
```

Dangerous Functions\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=87
Status	New

The dangerous function, memcpy, was found in use at line 1148 in awtk/ssl_srv.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	1291	1291
Object	memcpy	memcpy

Code Snippet	
File Name	awtk/ssl_srv.c
Method	static int ssl_parse_client_hello_v2(mbedtls_ssl_context *ssl)

```
....
1291.      memcpy( ssl->session_negotiate->id, p, ssl-
>session_negotiate->id_len );
```

Dangerous Functions\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=88
Status	New

The dangerous function, memcpy, was found in use at line 1148 in awtk/ssl_srv.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	1295	1295
Object	memcpy	memcpy

Code Snippet
File Name awtk/ssl_srv.c
Method static int ssl_parse_client_hello_v2(mbedtls_ssl_context *ssl)

```
....
1295.      memcpy( ssl->handshake->randbytes + 32 - chal_len, p,
chal_len );
```

Dangerous Functions\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=89
Status	New

The dangerous function, memcpy, was found in use at line 1413 in awtk/ssl_srv.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	1526	1526
Object	memcpy	memcpy

Code Snippet
File Name awtk/ssl_srv.c
Method static int ssl_parse_client_hello(mbedtls_ssl_context *ssl)

```
....
1526.          memcpy( ssl->cur_out_ctr + 2, ssl->in_ctr + 2, 6 );
```

Dangerous Functions\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=90
Status	New

The dangerous function, memcpy, was found in use at line 1413 in awtk/ssl_srv.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	1727	1727
Object	memcpy	memcpy

Code Snippet
File Name awtk/ssl_srv.c
Method static int ssl_parse_client_hello(mbedtls_ssl_context *ssl)

```
....
1727.      memcpy( ssl->handshake->randbytes, buf + 2, 32 );
```

Dangerous Functions\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=91
Status	New

The dangerous function, memcpy, was found in use at line 1413 in awtk/ssl_srv.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	1748	1748
Object	memcpy	memcpy

Code Snippet
File Name awtk/ssl_srv.c
Method static int ssl_parse_client_hello(mbedtls_ssl_context *ssl)

```
....
1748.       memcpy( ssl->session_negotiate->id, buf + 35,
```

Dangerous Functions\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=92
Status	New

The dangerous function, memcpy, was found in use at line 2313 in awtk/ssl_srv.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	2354	2354
Object	memcpy	memcpy

Code Snippet
File Name awtk/ssl_srv.c
Method static void ssl_write_cid_ext(mbedtls_ssl_context *ssl,

```
....
2354.       memcpy( p, ssl->own_cid, ssl->own_cid_len );
```

Dangerous Functions\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=93
Status	New

The dangerous function, memcpy, was found in use at line 2455 in awtk/ssl_srv.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	2479	2479
Object	memcpy	memcpy

Code Snippet
File Name awtk/ssl_srv.c
Method static void ssl_write_renegotiation_ext(mbedtls_ssl_context *ssl,

```
....
2479.          memcpy( p, ssl->peer_verify_data, ssl->verify_data_len );
```

Dangerous Functions\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=94
Status	New

The dangerous function, memcpy, was found in use at line 2455 in awtk/ssl_srv.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	2481	2481
Object	memcpy	memcpy

Code Snippet
File Name awtk/ssl_srv.c
Method static void ssl_write_renegotiation_ext(mbedtls_ssl_context *ssl,

```
....
2481.          memcpy( p, ssl->own_verify_data, ssl->verify_data_len );
```

Dangerous Functions\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=95
Status	New

The dangerous function, memcpy, was found in use at line 2598 in awtk/ssl_srv.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	2629	2629
Object	memcpy	memcpy

Code Snippet
File Name awtk/ssl_srv.c
Method static void ssl_write_alpn_ext(mbedtls_ssl_context *ssl,

```
....
2629.      memcpy( buf + 7, ssl->alpn_chosen, *olen - 7 );
```

Dangerous Functions\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=96
Status	New

The dangerous function, memcpy, was found in use at line 2634 in awtk/ssl_srv.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	2699	2699
Object	memcpy	memcpy

Code Snippet
File Name awtk/ssl_srv.c
Method static void ssl_write_use_srtp_ext(mbedtls_ssl_context *ssl,

```
....
2699.      memcpy( &buf[9], ssl->dtls_srtp_info.mki_value, mki_len );
```

Dangerous Functions\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=97
Status	New

The dangerous function, memcpy, was found in use at line 2777 in awtk/ssl_srv.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	2842	2842
Object	memcpy	memcpy

Code Snippet
File Name awtk/ssl_srv.c
Method static int ssl_write_server_hello(mbedtls_ssl_context *ssl)

```
....
2842.      memcpy( ssl->handshake->randbytes + 32, buf + 6, 32 );
```

Dangerous Functions\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=98
Status	New

The dangerous function, memcpy, was found in use at line 2777 in awtk/ssl_srv.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	2914	2914
Object	memcpy	memcpy

Code Snippet
File Name awtk/ssl_srv.c
Method static int ssl_write_server_hello(mbedtls_ssl_context *ssl)

```
....
2914.      memcpy( p, ssl->session_negotiate->id, ssl-
>session_negotiate->id_len );
```

Dangerous Functions\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=99
Status	New

The dangerous function, sprintf, was found in use at line 95 in awtk/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	98	98
Object	sprintf	sprintf

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(const char*) cJSON_Version(void)

```
....
98.      sprintf(version, "%i.%i.%i", CJSON_VERSION_MAJOR,
CJSON_VERSION_MINOR, CJSON_VERSION_PATCH);
```

Dangerous Functions\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=100
Status	New

The dangerous function, sprintf, was found in use at line 490 in awtk/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	508	508
Object	sprintf	sprintf

Code Snippet

File Name awtk/cJSON.c

Method static cJSON_bool print_number(const cJSON * const item, printbuffer * const output_buffer)

```
....
508.          length = sprintf((char*)number_buffer, "null");
```

Dangerous Functions\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=101
Status	New

The dangerous function, sprintf, was found in use at line 490 in awtk/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	513	513
Object	sprintf	sprintf

Code Snippet

File Name awtk/cJSON.c

Method static cJSON_bool print_number(const cJSON * const item, printbuffer * const output_buffer)

```
....
513.          length = sprintf((char*)number_buffer, "%1.15g", d);
```

Dangerous Functions\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=102
Status	New

The dangerous function, sprintf, was found in use at line 490 in awtk/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	519	519
Object	sprintf	sprintf

Code Snippet

File Name awtk/cJSON.c

Method static cJSON_bool print_number(const cJSON * const item, printbuffer * const output_buffer)

```
....
519.              length = sprintf((char*)number_buffer, "%1.17g", d);
```

Dangerous Functions\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=103
Status	New

The dangerous function, sprintf, was found in use at line 843 in awtk/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	952	952
Object	sprintf	sprintf

Code Snippet

File Name awtk/cJSON.c

Method static cJSON_bool print_string_ptr(const unsigned char * const input, printbuffer * const output_buffer)

```
....
952.                        sprintf((char*)output_pointer, "u%04x",
*input_pointer);
```

Dangerous Functions\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=104
Status	New

The dangerous function, sscanf, was found in use at line 490 in awtk/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	516	516
Object	sscanf	sscanf

Code Snippet
File Name awtk/cJSON.c
Method static cJSON_bool print_number(const cJSON * const item, printbuffer * const output_buffer)

```
....
516.          if ((sscanf((char*)number_buffer, "%lg", &test) != 1) ||
!compare_double((double)test, d))
```

Dangerous Functions\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=105
Status	New

The dangerous function, strcpy, was found in use at line 843 in awtk/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	865	865
Object	strcpy	strcpy

Code Snippet
File Name awtk/cJSON.c

Method	static cJSON_bool print_string_ptr(const unsigned char * const input, printbuffer * const output_buffer)

```
....
865.          strcpy((char*)output, "\"\"");
```

Dangerous Functions\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=106
Status	New

The dangerous function, strcpy, was found in use at line 1283 in awtk/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	1300	1300
Object	strcpy	strcpy

Code Snippet	
File Name	awtk/cJSON.c
Method	static cJSON_bool print_value(const cJSON * const item, printbuffer * const output_buffer)

```
....
1300.              strcpy((char*)output, "null");
```

Dangerous Functions\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=107
Status	New

The dangerous function, strcpy, was found in use at line 1283 in awtk/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	1309	1309
Object	strcpy	strcpy

Code Snippet	
File Name	awtk/cJSON.c

Method	static cJSON_bool print_value(const cJSON * const item, printbuffer * const output_buffer)

```
....
1309.              strcpy((char*)output, "false");
```

Dangerous Functions\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=108
Status	New

The dangerous function, strcpy, was found in use at line 1283 in awtk/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	1318	1318
Object	strcpy	strcpy

Code Snippet	
File Name	awtk/cJSON.c
Method	static cJSON_bool print_value(const cJSON * const item, printbuffer * const output_buffer)

```
....
1318.              strcpy((char*)output, "true");
```

Dangerous Functions\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=109
Status	New

The dangerous function, strcpy, was found in use at line 63 in awtk/network_interface_linux.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/network_interface_linux.c	awtk/network_interface_linux.c
Line	72	72
Object	strcpy	strcpy

Code Snippet	
File Name	awtk/network_interface_linux.c

Method	static char* network_interface_linux_get_ipaddr(network_interface_t* interface) {

```
....
72.    strcpy(ifr.ifr_name, interface->interface_name);
```

Dangerous Functions\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=110
Status	New

The dangerous function, strcpy, was found in use at line 86 in awtk/network_interface_linux.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/network_interface_linux.c	awtk/network_interface_linux.c
Line	95	95
Object	strcpy	strcpy

Code Snippet	
File Name	awtk/network_interface_linux.c
Method	static char* network_interface_linux_get_macaddr(network_interface_t* interface) {

```
....
95.    strcpy(ifr.ifr_name, interface->interface_name);
```

Dangerous Functions\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=111
Status	New

The dangerous function, strlen, was found in use at line 159 in awtk/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	169	169
Object	strlen	strlen

Code Snippet	
File Name	awtk/cJSON.c

Method	static unsigned char* cJSON_strdup(const unsigned char* string, const internal_hooks * const hooks)

```
....
169.      length = strlen((const char*)string) + sizeof("");
```

Dangerous Functions\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=112
Status	New

The dangerous function, strlen, was found in use at line 471 in awtk/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	480	480
Object	strlen	strlen

Code Snippet	
File Name	awtk/cJSON.c
Method	static void update_offset(printbuffer * const buffer)

```
....
480.      buffer->offset += strlen((const char*)buffer_pointer);
```

Dangerous Functions\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=113
Status	New

The dangerous function, strlen, was found in use at line 1016 in awtk/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	1031	1031
Object	strlen	strlen

Code Snippet	
File Name	awtk/cJSON.c
Method	CJSON_PUBLIC(cJSON *) cJSON_ParseWithOpts(const char *value, const char **return_parse_end, cJSON_bool require_null_terminated)

```
....
1031.      buffer.length = strlen((const char*)value) + sizeof("");
```

Dangerous Functions\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=114
Status	New

The dangerous function, strlen, was found in use at line 1283 in awtk/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	1332	1332
Object	strlen	strlen

Code Snippet

File Name awtk/cJSON.c

Method static cJSON_bool print_value(const cJSON * const item, printbuffer * const output_buffer)

```
....
1332.              raw_length = strlen(item->valuestring) + sizeof("");
```

Dangerous Functions\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=115
Status	New

The dangerous function, strlen, was found in use at line 692 in awtk/ssl_srv.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	758	758
Object	strlen	strlen

Code Snippet

File Name awtk/ssl_srv.c

Method static int ssl_parse_alpn_ext(mbedtls_ssl_context *ssl,

```
....
758.          ours_len = strlen( *ours );
```

Dangerous Functions\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=116
Status	New

The dangerous function, strlen, was found in use at line 2598 in awtk/ssl_srv.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	2619	2619
Object	strlen	strlen

Code Snippet
File Name awtk/ssl_srv.c
Method static void ssl_write_alpn_ext(mbedtls_ssl_context *ssl,

```
....
2619.      *olen = 7 + strlen( ssl->alpn_chosen );
```

Dangerous Functions\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=117
Status	New

The dangerous function, strtok, was found in use at line 202 in awtk/harness_argparser.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/harness_argparser.c	awtk/harness_argparser.c
Line	245	245
Object	strtok	strtok

Code Snippet
File Name awtk/harness_argparser.c
Method ParseConfig(char* file, SDLVisualTest_HarnessState* state)

```
....
245.              argv[i] = strtok(i == 0 ? line : NULL, "=");
```

Dangerous Functions\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=118
Status	New

The dangerous function, wcslen, was found in use at line 349 in awtk/edit.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	awtk/edit.c	awtk/edit.c
Line	353	353
Object	wcslen	wcslen

Code Snippet
File Name awtk/edit.c
Method static ret_t edit_commit_str(widget_t* widget, const char* str) {

```
....
353.    return edit_paste(widget, wstr, wcslen(wstr));
```

Use of Zero Initialized Pointer

Query Path:
CPP\Cx\CPP Medium Threat\Use of Zero Initialized Pointer Version:1

Categories

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

Description

Use of Zero Initialized Pointer\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=120
Status	New

The variable declared in after_end at awtk/cJSON.c in line 276 is not initialized when it is used by after_end at awtk/cJSON.c in line 276.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	279	348

Object	after_end	after_end

Code Snippet
File Name awtk/cJSON.c
Method static cJSON_bool parse_number(cJSON * const item, parse_buffer * const input_buffer)

```
....
279.      unsigned char *after_end = NULL;
....
348.      input_buffer->offset += (size_t)(after_end - number_c_string);
```

Use of Zero Initialized Pointer\Path 2:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=121
Status New

The variable declared in current_item at awtk/cJSON.c in line 1357 is not initialized when it is used by prev at awtk/cJSON.c in line 1357.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	1360	1411
Object	current_item	prev

Code Snippet
File Name awtk/cJSON.c
Method static cJSON_bool parse_array(cJSON * const item, parse_buffer * const input_buffer)

```
....
1360.      cJSON *current_item = NULL;
....
1411.              new_item->prev = current_item;
```

Use of Zero Initialized Pointer\Path 3:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=122
Status New

The variable declared in current_item at awtk/cJSON.c in line 1513 is not initialized when it is used by prev at awtk/cJSON.c in line 1513.

	Source	Destination

File	awtk/cJSON.c	awtk/cJSON.c
Line	1516	1565
Object	current_item	prev

Code Snippet
File Name awtk/cJSON.c
Method static cJSON_bool parse_object(cJSON * const item, parse_buffer * const input_buffer)

```
....
1516.       cJSON *current_item = NULL;
....
1565.               new_item->prev = current_item;
```

Use of Zero Initialized Pointer\Path 4:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=123
Status New

The variable declared in p at awtk/cJSON.c in line 2461 is not initialized when it is used by prev at awtk/cJSON.c in line 1836.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2465	1839
Object	p	prev

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(cJSON *) cJSON_CreateFloatArray(const float *numbers, int count)

```
....
2465.       cJSON *p = NULL;
```

File Name awtk/cJSON.c
Method static void suffix_object(cJSON *prev, cJSON *item)

```
....
1839.       item->prev = prev;
```

Use of Zero Initialized Pointer\Path 5:

Severity Medium
Result State To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=124
Status	New

The variable declared in p at awtk/cJSON.c in line 2426 is not initialized when it is used by prev at awtk/cJSON.c in line 1836.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2430	1839
Object	p	prev

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(cJSON *) cJSON_CreateIntArray(const int *numbers, int count)

```
....
2430.       cJSON *p = NULL;
```

File Name awtk/cJSON.c
Method static void suffix_object(cJSON *prev, cJSON *item)

```
....
1839.       item->prev = prev;
```

Use of Zero Initialized Pointer\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=125
Status	New

The variable declared in p at awtk/cJSON.c in line 2497 is not initialized when it is used by prev at awtk/cJSON.c in line 1836.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2501	1839
Object	p	prev

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(cJSON *) cJSON_CreateDoubleArray(const double *numbers, int count)

```
....
2501.       cJSON *p = NULL;
```

File Name awtk/cJSON.c

Method static void suffix_object(cJSON *prev, cJSON *item)

```
....
1839.       item->prev = prev;
```

Use of Zero Initialized Pointer\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=126
Status	New

The variable declared in p at awtk/cJSON.c in line 2533 is not initialized when it is used by prev at awtk/cJSON.c in line 1836.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2537	1839
Object	p	prev

Code Snippet

File Name awtk/cJSON.c

Method CJSON_PUBLIC(cJSON *) cJSON_CreateStringArray(const char *const *strings, int count)

```
....
2537.       cJSON *p = NULL;
```

File Name awtk/cJSON.c

Method static void suffix_object(cJSON *prev, cJSON *item)

```
....
1839.       item->prev = prev;
```

Use of Zero Initialized Pointer\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=127
Status	New

The variable declared in p at awtk/cJSON.c in line 2426 is not initialized when it is used by p at awtk/cJSON.c in line 2426.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2430	2455
Object	p	p

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(cJSON *) cJSON_CreateIntArray(const int *numbers, int count)

```
....
2430.       cJSON *p = NULL;
....
2455.           p = n;
```

Use of Zero Initialized Pointer\Path 9:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=128
Status New

The variable declared in p at awtk/cJSON.c in line 2533 is not initialized when it is used by next at awtk/cJSON.c in line 1836.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2537	1838
Object	p	next

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(cJSON *) cJSON_CreateStringArray(const char *const *strings, int count)

```
....
2537.       cJSON *p = NULL;
```

File Name awtk/cJSON.c
Method static void suffix_object(cJSON *prev, cJSON *item)

```
....
1838.       prev->next = item;
```

Use of Zero Initialized Pointer\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=129
Status	New

The variable declared in p at awtk/cJSON.c in line 2426 is not initialized when it is used by next at awtk/cJSON.c in line 1836.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2430	1838
Object	p	next

Code Snippet

File Name awtk/cJSON.c

Method CJSON_PUBLIC(cJSON *) cJSON_CreateIntArray(const int *numbers, int count)

```
....
2430.       cJSON *p = NULL;
```

File Name awtk/cJSON.c

Method static void suffix_object(cJSON *prev, cJSON *item)

```
....
1838.       prev->next = item;
```

Use of Zero Initialized Pointer\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=130
Status	New

The variable declared in p at awtk/cJSON.c in line 2461 is not initialized when it is used by next at awtk/cJSON.c in line 1836.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2465	1838
Object	p	next

Code Snippet

File Name awtk/cJSON.c

Method	CJSON_PUBLIC(cJSON *) cJSON_CreateFloatArray(const float *numbers, int count)

```
....
2465.       cJSON *p = NULL;
```

File Name	awtk/cJSON.c
Method	static void suffix_object(cJSON *prev, cJSON *item)

```
....
1838.       prev->next = item;
```

Use of Zero Initialized Pointer\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=131
Status	New

The variable declared in p at awtk/cJSON.c in line 2497 is not initialized when it is used by next at awtk/cJSON.c in line 1836.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2501	1838
Object	p	next

Code Snippet	
File Name	awtk/cJSON.c
Method	CJSON_PUBLIC(cJSON *) cJSON_CreateDoubleArray(const double *numbers, int count)

```
....
2501.       cJSON *p = NULL;
```

File Name	awtk/cJSON.c
Method	static void suffix_object(cJSON *prev, cJSON *item)

```
....
1838.       prev->next = item;
```

Use of Zero Initialized Pointer\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=100

	04&pathid=132
Status	New

The variable declared in p at awtk/cJSON.c in line 2461 is not initialized when it is used by p at awtk/cJSON.c in line 2461.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2465	2491
Object	p	p

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(cJSON *) cJSON_CreateFloatArray(const float *numbers, int count)

```
....
2465.       cJSON *p = NULL;
....
2491.           p = n;
```

Use of Zero Initialized Pointer\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=133
Status	New

The variable declared in p at awtk/cJSON.c in line 2497 is not initialized when it is used by p at awtk/cJSON.c in line 2497.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2501	2527
Object	p	p

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(cJSON *) cJSON_CreateDoubleArray(const double *numbers, int count)

```
....
2501.       cJSON *p = NULL;
....
2527.           p = n;
```

Use of Zero Initialized Pointer\Path 15:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=134
Status	New

The variable declared in p at awtk/cJSON.c in line 2533 is not initialized when it is used by p at awtk/cJSON.c in line 2533.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2537	2563
Object	p	p

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(cJSON *) cJSON_CreateStringArray(const char *const *strings, int count)

```
....
2537.       cJSON *p = NULL;
....
2563.           p = n;
```

Use of Zero Initialized Pointer\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=135
Status	New

The variable declared in a_element at awtk/cJSON.c in line 2846 is not initialized when it is used by a_element at awtk/cJSON.c in line 2846.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2930	2935
Object	a_element	a_element

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(cJSON_bool) cJSON_Compare(const cJSON * const a, const cJSON * const b, const cJSON_bool case_sensitive)

```
....
2930.               cJSON *a_element = NULL;
....
2935.                   b_element = get_object_item(b, a_element->string,
case_sensitive);
```

Use of Zero Initialized Pointer\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=136
Status	New

The variable declared in addr at awtk/lcd_sdl2_mono.c in line 59 is not initialized when it is used by addr at awtk/lcd_sdl2_mono.c in line 59.

	Source	Destination
File	awtk/lcd_sdl2_mono.c	awtk/lcd_sdl2_mono.c
Line	64	85
Object	addr	addr

Code Snippet
File Name awtk/lcd_sdl2_mono.c
Method static ret_t lcd_sdl2_mono_flush(lcd_t* lcd) {

```
....
64.    void* addr = NULL;
....
85.          p = ((uint8_t*)addr) + j * pitch + i * 4;
```

Use of Zero Initialized Pointer\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=137
Status	New

The variable declared in pBuf at awtk/miniz_tinfl.c in line 647 is not initialized when it is used by pBuf at awtk/miniz_tinfl.c in line 647.

	Source	Destination
File	awtk/miniz_tinfl.c	awtk/miniz_tinfl.c
Line	650	679
Object	pBuf	pBuf

Code Snippet
File Name awtk/miniz_tinfl.c
Method void *tinfl_decompress_mem_to_heap(const void *pSrc_buf, size_t src_buf_len, size_t *pOut_len, int flags)

```
....
650.      void *pBuf = NULL, *pNew_buf;
....
679.          pBuf = pNew_buf;
```

Use of Zero Initialized Pointer\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=138
Status	New

The variable declared in pBuf at awtk/miniz_tinfl.c in line 647 is not initialized when it is used by pBuf at awtk/miniz_tinfl.c in line 647.

	Source	Destination
File	awtk/miniz_tinfl.c	awtk/miniz_tinfl.c
Line	650	657
Object	pBuf	pBuf

Code Snippet
File Name awtk/miniz_tinfl.c
Method void *tinfl_decompress_mem_to_heap(const void *pSrc_buf, size_t src_buf_len, size_t *pOut_len, int flags)

```
....
650.      void *pBuf = NULL, *pNew_buf;
....
657.          tinfl_status status = tinfl_decompress(&decomp, (const
mz_uint8 *)pSrc_buf + src_buf_ofs, &src_buf_size, (mz_uint8 *)pBuf, pBuf
? (mz_uint8 *)pBuf + *pOut_len : NULL, &dst_buf_size,
```

Use of Zero Initialized Pointer\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=139
Status	New

The variable declared in dig_signed at awtk/ssl_srv.c in line 3264 is not initialized when it is used by dig_signed at awtk/ssl_srv.c in line 3264.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	3272	3455
Object	dig_signed	dig_signed

Code Snippet
File Name awtk/ssl_srv.c
Method static int ssl_prepare_server_key_exchange(mbedtls_ssl_context *ssl,

```
....
3272.      unsigned char *dig_signed = NULL;
....
3455.           size_t dig_signed_len = ssl->out_msg + ssl->out_msglen -
dig_signed;
```

Use of Zero Initialized Pointer\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=140
Status	New

The variable declared in buffer at awtk/cJSON.c in line 383 is not initialized when it is used by newbuffer at awtk/cJSON.c in line 383.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	454	436
Object	buffer	newbuffer

Code Snippet
File Name awtk/cJSON.c
Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
454.              p->buffer = NULL;
....
436.          newbuffer = (unsigned char*)p->hooks.reallocate(p->buffer,
newsize);
```

Use of Zero Initialized Pointer\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=141
Status	New

The variable declared in buffer at awtk/cJSON.c in line 383 is not initialized when it is used by newbuffer at awtk/cJSON.c in line 383.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	441	436
Object	buffer	newbuffer

Code Snippet

File Name	awtk/cJSON.c
Method	static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
441.                p->buffer = NULL;
....
436.            newbuffer = (unsigned char*)p->hooks.reallocate(p->buffer,
newsize);
```

Use of Zero Initialized Pointer\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=142
Status	New

The variable declared in buffer at awtk/cJSON.c in line 383 is not initialized when it is used by output_pointer at awtk/cJSON.c in line 1622.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	441	1654
Object	buffer	output_pointer

Code Snippet	
File Name	awtk/cJSON.c
Method	static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
441.                p->buffer = NULL;
```

File Name	awtk/cJSON.c
Method	static cJSON_bool print_object(const cJSON * const item, printbuffer * const output_buffer)

```
....
1654.                output_pointer = ensure(output_buffer, output_buffer-
>depth);
```

Use of Zero Initialized Pointer\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=143
Status	New

The variable declared in buffer at awtk/cJSON.c in line 383 is not initialized when it is used by output_pointer at awtk/cJSON.c in line 1622.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	454	1654
Object	buffer	output_pointer

Code Snippet
File Name awtk/cJSON.c
Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
454.                p->buffer = NULL;
```

File Name awtk/cJSON.c

Method static cJSON_bool print_object(const cJSON * const item, printbuffer * const output_buffer)

```
....
1654.                output_pointer = ensure(output_buffer, output_buffer->depth);
```

Use of Zero Initialized Pointer\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=144
Status	New

The variable declared in buffer at awtk/cJSON.c in line 383 is not initialized when it is used by output at awtk/cJSON.c in line 843.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	441	896
Object	buffer	output

Code Snippet
File Name awtk/cJSON.c
Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
441.                p->buffer = NULL;
```

File Name awtk/cJSON.c

Method	static cJSON_bool print_string_ptr(const unsigned char * const input, printbuffer * const output_buffer)

```
....
896.      output = ensure(output_buffer, output_length +
sizeof("\"\""));
```

Use of Zero Initialized Pointer\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=145
Status	New

The variable declared in buffer at awtk/cJSON.c in line 383 is not initialized when it is used by newbuffer at awtk/cJSON.c in line 383.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	441	449
Object	buffer	newbuffer

Code Snippet	
File Name	awtk/cJSON.c
Method	static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
441.              p->buffer = NULL;
....
449.          newbuffer = (unsigned char*)p->hooks.allocate(newsize);
```

Use of Zero Initialized Pointer\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=146
Status	New

The variable declared in buffer at awtk/cJSON.c in line 383 is not initialized when it is used by newbuffer at awtk/cJSON.c in line 383.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	454	449
Object	buffer	newbuffer

Code Snippet

File Name awtk/cJSON.c
Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
454.                p->buffer = NULL;
....
449.            newbuffer = (unsigned char*)p->hooks.allocate(newsize);
```

Use of Zero Initialized Pointer\Path 28:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=147
Status New

The variable declared in buffer at awtk/cJSON.c in line 383 is not initialized when it is used by output_pointer at awtk/cJSON.c in line 1622.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	441	1715
Object	buffer	output_pointer

Code Snippet
File Name awtk/cJSON.c
Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
441.                p->buffer = NULL;
```

File Name awtk/cJSON.c
Method static cJSON_bool print_object(const cJSON * const item, printbuffer * const output_buffer)

```
....
1715.       output_pointer = ensure(output_buffer, output_buffer->format
? (output_buffer->depth + 1) : 2);
```

Use of Zero Initialized Pointer\Path 29:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=148
Status New

The variable declared in buffer at awtk/cJSON.c in line 383 is not initialized when it is used by output_pointer at awtk/cJSON.c in line 1622.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	454	1715
Object	buffer	output_pointer

Code Snippet
File Name awtk/cJSON.c
Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
454.                p->buffer = NULL;
```

File Name awtk/cJSON.c

Method static cJSON_bool print_object(const cJSON * const item, printbuffer * const output_buffer)

```
....
1715.       output_pointer = ensure(output_buffer, output_buffer->format
? (output_buffer->depth + 1) : 2);
```

Use of Zero Initialized Pointer\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=149
Status	New

The variable declared in buffer at awtk/cJSON.c in line 383 is not initialized when it is used by output_pointer at awtk/cJSON.c in line 1622.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	454	1695
Object	buffer	output_pointer

Code Snippet
File Name awtk/cJSON.c
Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
454.                p->buffer = NULL;
```

File Name awtk/cJSON.c

Method	static cJSON_bool print_object(const cJSON * const item, printbuffer * const output_buffer)

```
....
1695.          output_pointer = ensure(output_buffer, length + 1);
```

Use of Zero Initialized Pointer\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=150
Status	New

The variable declared in buffer at awtk/cJSON.c in line 383 is not initialized when it is used by output_pointer at awtk/cJSON.c in line 1622.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	441	1695
Object	buffer	output_pointer

Code Snippet	
File Name	awtk/cJSON.c
Method	static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
441.              p->buffer = NULL;
```

File Name	awtk/cJSON.c
Method	static cJSON_bool print_object(const cJSON * const item, printbuffer * const output_buffer)

```
....
1695.          output_pointer = ensure(output_buffer, length + 1);
```

Use of Zero Initialized Pointer\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=151
Status	New

The variable declared in buffer at awtk/cJSON.c in line 383 is not initialized when it is used by buffer at awtk/cJSON.c in line 471.

	Source	Destination

File	awtk/cJSON.c	awtk/cJSON.c
Line	454	478
Object	buffer	buffer

Code Snippet
File Name awtk/cJSON.c
Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
454.                p->buffer = NULL;
```

File Name awtk/cJSON.c
Method static void update_offset(printbuffer * const buffer)

```
....
478.        buffer_pointer = buffer->buffer + buffer->offset;
```

Use of Zero Initialized Pointer\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=152
Status	New

The variable declared in buffer at awtk/cJSON.c in line 383 is not initialized when it is used by buffer at awtk/cJSON.c in line 471.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	441	478
Object	buffer	buffer

Code Snippet
File Name awtk/cJSON.c
Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
441.                p->buffer = NULL;
```

File Name awtk/cJSON.c
Method static void update_offset(printbuffer * const buffer)

```
....
478.        buffer_pointer = buffer->buffer + buffer->offset;
```

Use of Zero Initialized Pointer\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=153
Status	New

The variable declared in buffer at awtk/cJSON.c in line 383 is not initialized when it is used by output_pointer at awtk/cJSON.c in line 1622.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	454	1674
Object	buffer	output_pointer

Code Snippet

File Name awtk/cJSON.c

Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
454.                p->buffer = NULL;
```

File Name awtk/cJSON.c

Method static cJSON_bool print_object(const cJSON * const item, printbuffer * const output_buffer)

```
....
1674.           output_pointer = ensure(output_buffer, length);
```

Use of Zero Initialized Pointer\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=154
Status	New

The variable declared in buffer at awtk/cJSON.c in line 383 is not initialized when it is used by output_pointer at awtk/cJSON.c in line 1622.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	441	1674
Object	buffer	output_pointer

Code Snippet

File Name	awtk/cJSON.c
Method	static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
441.                p->buffer = NULL;
```

File Name	awtk/cJSON.c
Method	static cJSON_bool print_object(const cJSON * const item, printbuffer * const output_buffer)

```
....
1674.           output_pointer = ensure(output_buffer, length);
```

Use of Zero Initialized Pointer\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=155
Status	New

The variable declared in buffer at awtk/cJSON.c in line 383 is not initialized when it is used by output_pointer at awtk/cJSON.c in line 1451.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	441	1500
Object	buffer	output_pointer

Code Snippet

File Name	awtk/cJSON.c
Method	static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
441.                p->buffer = NULL;
```

File Name	awtk/cJSON.c
Method	static cJSON_bool print_array(const cJSON * const item, printbuffer * const output_buffer)

```
....
1500.       output_pointer = ensure(output_buffer, 2);
```

Use of Zero Initialized Pointer\Path 37:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=156
Status	New

The variable declared in buffer at awtk/cJSON.c in line 383 is not initialized when it is used by output_pointer at awtk/cJSON.c in line 1451.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	454	1500
Object	buffer	output_pointer

Code Snippet
File Name awtk/cJSON.c
Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
454.                p->buffer = NULL;
```

File Name awtk/cJSON.c

Method static cJSON_bool print_array(const cJSON * const item, printbuffer * const output_buffer)

```
....
1500.       output_pointer = ensure(output_buffer, 2);
```

Use of Zero Initialized Pointer\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=157
Status	New

The variable declared in buffer at awtk/cJSON.c in line 383 is not initialized when it is used by output_pointer at awtk/cJSON.c in line 1451.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	454	1484
Object	buffer	output_pointer

Code Snippet
File Name awtk/cJSON.c
Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
454.              p->buffer = NULL;
```

File Name awtk/cJSON.c

Method static cJSON_bool print_array(const cJSON * const item, printbuffer * const output_buffer)

```
....
1484.              output_pointer = ensure(output_buffer, length + 1);
```

Use of Zero Initialized Pointer\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=158
Status	New

The variable declared in buffer at awtk/cJSON.c in line 383 is not initialized when it is used by output_pointer at awtk/cJSON.c in line 1451.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	441	1484
Object	buffer	output_pointer

Code Snippet

File Name awtk/cJSON.c

Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
441.              p->buffer = NULL;
```

File Name awtk/cJSON.c

Method static cJSON_bool print_array(const cJSON * const item, printbuffer * const output_buffer)

```
....
1484.              output_pointer = ensure(output_buffer, length + 1);
```

Use of Zero Initialized Pointer\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=159

Status	New

The variable declared in valuestring at awtk/cJSON.c in line 1513 is not initialized when it is used by current_item at awtk/cJSON.c in line 1513.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	1580	1579
Object	valuestring	current_item

Code Snippet
File Name awtk/cJSON.c
Method static cJSON_bool parse_object(cJSON * const item, parse_buffer * const input_buffer)

```
....
1580.           current_item->valuestring = NULL;
....
1579.           current_item->string = current_item->valuestring;
```

Use of Zero Initialized Pointer\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=160
Status	New

The variable declared in prev at awtk/cJSON.c in line 2190 is not initialized when it is used by next at awtk/cJSON.c in line 224.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2219	229
Object	prev	next

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(cJSON_bool) cJSON_ReplaceItemViaPointer(cJSON * const parent, cJSON * const item, cJSON * replacement)

```
....
2219.       item->prev = NULL;
```

File Name awtk/cJSON.c
Method CJSON_PUBLIC(void) cJSON_Delete(cJSON *item)

```
....
229.          next = item->next;
```

Use of Zero Initialized Pointer\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=161
Status	New

The variable declared in prev at awtk/cJSON.c in line 2091 is not initialized when it is used by next at awtk/cJSON.c in line 224.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2115	229
Object	prev	next

Code Snippet

File Name awtk/cJSON.c

Method CJSON_PUBLIC(cJSON *) cJSON_DetachItemViaPointer(cJSON *parent, cJSON * const item)

```
....
2115.      item->prev = NULL;
```

File Name awtk/cJSON.c

Method CJSON_PUBLIC(void) cJSON_Delete(cJSON *item)

```
....
229.          next = item->next;
```

Use of Zero Initialized Pointer\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=162
Status	New

The variable declared in next at awtk/cJSON.c in line 2091 is not initialized when it is used by next at awtk/cJSON.c in line 224.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c

Line	2116	229
Object	next	next

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(cJSON *) cJSON_DetachItemViaPointer(cJSON *parent, cJSON * const item)

```
....
2116.       item->next = NULL;
```

File Name awtk/cJSON.c
Method CJSON_PUBLIC(void) cJSON_Delete(cJSON *item)

```
....
229.            next = item->next;
```

Use of Zero Initialized Pointer\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=163
Status	New

The variable declared in next at awtk/SDL_pixels.c in line 537 is not initialized when it is used by next at awtk/SDL_pixels.c in line 496.

	Source	Destination
File	awtk/SDL_pixels.c	awtk/SDL_pixels.c
Line	596	527
Object	next	next

Code Snippet
File Name awtk/SDL_pixels.c
Method SDL_InitFormat(SDL_PixelFormat * format, Uint32 pixel_format)

```
....
596.       format->next = NULL;
```

File Name awtk/SDL_pixels.c
Method SDL_AllocFormat(Uint32 pixel_format)

```
....
527.            format->next = formats;
```

Use of Zero Initialized Pointer\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=164
Status	New

The variable declared in palette at awtk/SDL_pixels.c in line 537 is not initialized when it is used by next at awtk/SDL_pixels.c in line 496.

	Source	Destination
File	awtk/SDL_pixels.c	awtk/SDL_pixels.c
Line	594	527
Object	palette	next

Code Snippet

File Name awtk/SDL_pixels.c

Method SDL_InitFormat(SDL_PixelFormat * format, Uint32 pixel_format)

```
....
594.      format->palette = NULL;
```

File Name awtk/SDL_pixels.c

Method SDL_AllocFormat(Uint32 pixel_format)

```
....
527.          format->next = formats;
```

Use of Zero Initialized Pointer\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=165
Status	New

The variable declared in data at awtk/SDL_wave.c in line 448 is not initialized when it is used by data at awtk/SDL_wave.c in line 448.

	Source	Destination
File	awtk/SDL_wave.c	awtk/SDL_wave.c
Line	497	508
Object	data	data

Code Snippet

File Name awtk/SDL_wave.c

Method	SDL_LoadWAV_RW(SDL_RWops * src, int freesrc,

```
....
497.          chunk.data = NULL;
....
508.      format = (WaveFMT *) chunk.data;
```

Use of Zero Initialized Pointer\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=166
Status	New

The variable declared in data at awtk/SDL_wave.c in line 678 is not initialized when it is used by data at awtk/SDL_wave.c in line 448.

	Source	Destination
File	awtk/SDL_wave.c	awtk/SDL_wave.c
Line	688	508
Object	data	data

Code Snippet
File Name awtk/SDL_wave.c
Method ReadChunk(SDL_RWops * src, Chunk * chunk)

```
....
688.          chunk->data = NULL;
```

File Name awtk/SDL_wave.c
Method SDL_LoadWAV_RW(SDL_RWops * src, int freesrc,

```
....
508.      format = (WaveFMT *) chunk.data;
```

Buffer Overflow boundcpy WrongSizeParam

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

Categories

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

Description

Buffer Overflow boundcpy WrongSizeParam\Path 1:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=1
Status	New

The size of the buffer used by *create_reference in cJSON, at line 1843 of awtk/cJSON.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *create_reference passes to cJSON, at line 1843 of awtk/cJSON.c, to overwrite the target buffer.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	1857	1857
Object	cJSON	cJSON

Code Snippet
File Name awtk/cJSON.c
Method static cJSON *create_reference(const cJSON *item, const internal_hooks * const hooks)

```
....
1857.      memcpy(reference, item, sizeof(cJSON));
```

Buffer Overflow boundcpy WrongSizeParam\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=2
Status	New

The size of the buffer used by ssl_parse_session_ticket_ext in mbedtls_ssl_session, at line 618 of awtk/ssl_srv.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ssl_parse_session_ticket_ext passes to mbedtls_ssl_session, at line 618 of awtk/ssl_srv.c, to overwrite the target buffer.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	675	675
Object	mbedtls_ssl_session	mbedtls_ssl_session

Code Snippet
File Name awtk/ssl_srv.c
Method static int ssl_parse_session_ticket_ext(mbedtls_ssl_context *ssl,

```
....
675.      memcpy( ssl->session_negotiate, &session, sizeof(
mbedtls_ssl_session ) );
```

Buffer Overflow boundcpy WrongSizeParam\Path 3:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=3
Status	New

The size of the buffer used by *cJSON_New_Item in cJSON, at line 212 of awtk/cJSON.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *cJSON_New_Item passes to cJSON, at line 212 of awtk/cJSON.c, to overwrite the target buffer.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	217	217
Object	cJSON	cJSON

Code Snippet
File Name awtk/cJSON.c
Method static cJSON *cJSON_New_Item(const internal_hooks * const hooks)

```
....
217.          memset(node, '\0', sizeof(cJSON));
```

Buffer Overflow boundcpy WrongSizeParam\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=4
Status	New

The size of the buffer used by date_time_init in date_time_t, at line 67 of awtk/date_time.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that date_time_init passes to date_time_t, at line 67 of awtk/date_time.c, to overwrite the target buffer.

	Source	Destination
File	awtk/date_time.c	awtk/date_time.c
Line	70	70
Object	date_time_t	date_time_t

Code Snippet
File Name awtk/date_time.c
Method date_time_t* date_time_init(date_time_t* dt) {

```
....
70.    memset(dt, 0x00, sizeof(date_time_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=5
Status	New

The size of the buffer used by ssl_parse_client_hello_v2 in ->, at line 1148 of awtk/ssl_srv.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ssl_parse_client_hello_v2 passes to ->, at line 1148 of awtk/ssl_srv.c, to overwrite the target buffer.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	1290	1290
Object	->	->

Code Snippet
File Name awtk/ssl_srv.c
Method static int ssl_parse_client_hello_v2(mbedtls_ssl_context *ssl)

```
....
1290.               sizeof( ssl->session_negotiate->id ) );
```

Buffer Overflow boundcpy WrongSizeParam\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=6
Status	New

The size of the buffer used by ssl_parse_client_hello in ->, at line 1413 of awtk/ssl_srv.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ssl_parse_client_hello passes to ->, at line 1413 of awtk/ssl_srv.c, to overwrite the target buffer.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	1747	1747
Object	->	->

Code Snippet
File Name awtk/ssl_srv.c
Method static int ssl_parse_client_hello(mbedtls_ssl_context *ssl)

```
....
1747.               sizeof( ssl->session_negotiate->id ) );
```

Buffer Overflow boundcpy WrongSizeParam\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=7
Status	New

The size of the buffer used by tinfl_decompress in mz_uint32, at line 174 of awtk/miniz_tinfl.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that tinfl_decompress passes to mz_uint32, at line 174 of awtk/miniz_tinfl.c, to overwrite the target buffer.

	Source	Destination
File	awtk/miniz_tinfl.c	awtk/miniz_tinfl.c
Line	527	527
Object	mz_uint32	mz_uint32

Code Snippet
File Name awtk/miniz_tinfl.c
Method tinfl_status tinfl_decompress(tinfl_decompressor *r, const mz_uint8 *pIn_buf_next, size_t *pIn_buf_size, mz_uint8 *pOut_buf_start, mz_uint8 *pOut_buf_next, size_t *pOut_buf_size, const mz_uint32 decomp_flags)

```
....
527.                                    memcpy(pOut_buf_cur, pSrc,
sizeof(mz_uint32)*2);
```

Buffer Overflow boundcpy WrongSizeParam\Path 8:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=8
Status New

The size of the buffer used by cJSON_strdup in length, at line 159 of awtk/cJSON.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that cJSON_strdup passes to length, at line 159 of awtk/cJSON.c, to overwrite the target buffer.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	175	175
Object	length	length

Code Snippet
File Name awtk/cJSON.c
Method static unsigned char* cJSON_strdup(const unsigned char* string, const internal_hooks * const hooks)

```
....
175.      memcpy(copy, string, length);
```

Buffer Overflow boundcpy WrongSizeParam\Path 9:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=9
Status New

The size of the buffer used by print_string_ptr in output_length, at line 843 of awtk/cJSON.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that print_string_ptr passes to output_length, at line 843 of awtk/cJSON.c, to overwrite the target buffer.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	906	906
Object	output_length	output_length

Code Snippet
File Name awtk/cJSON.c
Method static cJSON_bool print_string_ptr(const unsigned char * const input, printbuffer * const output_buffer)

```
....
906.          memcpy(output + 1, input, output_length);
```

Buffer Overflow boundcpy WrongSizeParam\Path 10:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=10
Status New

The size of the buffer used by print_value in raw_length, at line 1283 of awtk/cJSON.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that print_value passes to raw_length, at line 1283 of awtk/cJSON.c, to overwrite the target buffer.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	1338	1338
Object	raw_length	raw_length

Code Snippet
File Name awtk/cJSON.c
Method static cJSON_bool print_value(const cJSON * const item, printbuffer * const output_buffer)

```
....
1338.              memcpy(output, item->valuestring, raw_length);
```

Buffer Overflow boundcpy WrongSizeParam\Path 11:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=11
Status New

The size of the buffer used by mbedtls_ssl_set_client_transport_id in ilen, at line 49 of awtk/ssl_srv.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that mbedtls_ssl_set_client_transport_id passes to ilen, at line 49 of awtk/ssl_srv.c, to overwrite the target buffer.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	61	61
Object	ilen	ilen

Code Snippet
File Name awtk/ssl_srv.c
Method int mbedtls_ssl_set_client_transport_id(mbedtls_ssl_context *ssl,

```
....
61.      memcpy( ssl->cli_id, info, ilen );
```

Buffer Overflow boundcpy WrongSizeParam\Path 12:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=12
Status New

The size of the buffer used by ssl_parse_client_hello_v2 in ssl, at line 1148 of awtk/ssl_srv.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ssl_parse_client_hello_v2 passes to ssl, at line 1148 of awtk/ssl_srv.c, to overwrite the target buffer.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	1291	1291
Object	ssl	ssl

Code Snippet
File Name awtk/ssl_srv.c
Method static int ssl_parse_client_hello_v2(mbedtls_ssl_context *ssl)

```
....
1291.      memcpy( ssl->session_negotiate->id, p, ssl-
>session_negotiate->id_len );
```

Buffer Overflow boundcpy WrongSizeParam\Path 13:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=13
Status New

The size of the buffer used by ssl_parse_client_hello_v2 in chal_len, at line 1148 of awtk/ssl_srv.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ssl_parse_client_hello_v2 passes to chal_len, at line 1148 of awtk/ssl_srv.c, to overwrite the target buffer.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	1295	1295
Object	chal_len	chal_len

Code Snippet
File Name awtk/ssl_srv.c
Method static int ssl_parse_client_hello_v2(mbedtls_ssl_context *ssl)

```
....
1295.      memcpy( ssl->handshake->randbytes + 32 - chal_len, p,
chal_len );
```

Buffer Overflow boundcpy WrongSizeParam\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=14
Status	New

The size of the buffer used by ssl_parse_client_hello in ssl, at line 1413 of awtk/ssl_srv.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ssl_parse_client_hello passes to ssl, at line 1413 of awtk/ssl_srv.c, to overwrite the target buffer.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	1749	1749
Object	ssl	ssl

Code Snippet
File Name awtk/ssl_srv.c
Method static int ssl_parse_client_hello(mbedtls_ssl_context *ssl)

```
....
1749.              ssl->session_negotiate->id_len );
```

Buffer Overflow boundcpy WrongSizeParam\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=15
Status	New

The size of the buffer used by ssl_write_cid_ext in ssl, at line 2313 of awtk/ssl_srv.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ssl_write_cid_ext passes to ssl, at line 2313 of awtk/ssl_srv.c, to overwrite the target buffer.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	2354	2354
Object	ssl	ssl

Code Snippet
File Name awtk/ssl_srv.c
Method static void ssl_write_cid_ext(mbedtls_ssl_context *ssl,

```
....
2354.       memcpy( p, ssl->own_cid, ssl->own_cid_len );
```

Buffer Overflow boundcpy WrongSizeParam\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=16
Status	New

The size of the buffer used by ssl_write_renegotiation_ext in ssl, at line 2455 of awtk/ssl_srv.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ssl_write_renegotiation_ext passes to ssl, at line 2455 of awtk/ssl_srv.c, to overwrite the target buffer.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	2479	2479
Object	ssl	ssl

Code Snippet
File Name awtk/ssl_srv.c
Method static void ssl_write_renegotiation_ext(mbedtls_ssl_context *ssl,

```
....
2479.           memcpy( p, ssl->peer_verify_data, ssl->verify_data_len );
```

Buffer Overflow boundcpy WrongSizeParam\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=17
Status	New

The size of the buffer used by ssl_write_renegotiation_ext in ssl, at line 2455 of awtk/ssl_srv.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ssl_write_renegotiation_ext passes to ssl, at line 2455 of awtk/ssl_srv.c, to overwrite the target buffer.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	2481	2481
Object	ssl	ssl

Code Snippet
File Name awtk/ssl_srv.c
Method static void ssl_write_renegotiation_ext(mbedtls_ssl_context *ssl,

```
....
2481.          memcpy( p, ssl->own_verify_data, ssl->verify_data_len );
```

Buffer Overflow boundcpy WrongSizeParam\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=18
Status	New

The size of the buffer used by ssl_write_use_srtp_ext in mki_len, at line 2634 of awtk/ssl_srv.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ssl_write_use_srtp_ext passes to mki_len, at line 2634 of awtk/ssl_srv.c, to overwrite the target buffer.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	2699	2699
Object	mki_len	mki_len

Code Snippet
File Name awtk/ssl_srv.c
Method static void ssl_write_use_srtp_ext(mbedtls_ssl_context *ssl,

```
....
2699.      memcpy( &buf[9], ssl->dtls_srtp_info.mki_value, mki_len );
```

Buffer Overflow boundcpy WrongSizeParam\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=19
Status	New

The size of the buffer used by ssl_write_server_hello in ssl, at line 2777 of awtk/ssl_srv.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ssl_write_server_hello passes to ssl, at line 2777 of awtk/ssl_srv.c, to overwrite the target buffer.

	Source	Destination

File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	2914	2914
Object	ssl	ssl

Code Snippet
File Name awtk/ssl_srv.c
Method static int ssl_write_server_hello(mbedtls_ssl_context *ssl)

```
....
2914.      memcpy( p, ssl->session_negotiate->id, ssl-
>session_negotiate->id_len );
```

Integer Overflow

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

Categories

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

Description
Integer Overflow\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=69
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 3913 of awtk/ssl_srv.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	awtk/ssl_srv.c	awtk/ssl_srv.c
Line	3954	3954
Object	AssignExpr	AssignExpr

Code Snippet
File Name awtk/ssl_srv.c
Method return(ret);

```
....
3954.       * padding, to protect against timing-based Bleichenbacher-
type
```

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-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=119
Status	New

Method edit_on_password_visible at line 1781 of awtk/edit.c defines password_visible, which is designated to contain user passwords. However, while plaintext passwords are later assigned to password_visible, this variable is never cleared from memory.

	Source	Destination
File	awtk/edit.c	awtk/edit.c
Line	1783	1783
Object	password_visible	password_visible

Code Snippet
File Name awtk/edit.c
Method static ret_t edit_on_password_visible(void* ctx, event_t* e) {

```
....
1783.    bool_t password_visible = FALSE;
```

NULL Pointer Dereference

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

Categories

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

Description

NULL Pointer Dereference\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=41
Status	New

The variable declared in null at awtk/cJSON.c in line 979 is not initialized when it is used by content at awtk/cJSON.c in line 979.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c

Line	983	981
Object	null	content

Code Snippet
File Name awtk/cJSON.c
Method static parse_buffer *buffer_skip_whitespace(parse_buffer * const buffer)

```
....
983.          return NULL;
....
981.      if ((buffer == NULL) || (buffer->content == NULL))
```

NULL Pointer Dereference\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=42
Status	New

The variable declared in null at awtk/cJSON.c in line 1000 is not initialized when it is used by content at awtk/cJSON.c in line 979.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	1004	981
Object	null	content

Code Snippet
File Name awtk/cJSON.c
Method static parse_buffer *skip_utf8_bom(parse_buffer * const buffer)

```
....
1004.          return NULL;
```

File Name awtk/cJSON.c
Method static parse_buffer *buffer_skip_whitespace(parse_buffer * const buffer)

```
....
981.      if ((buffer == NULL) || (buffer->content == NULL))
```

NULL Pointer Dereference\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=43
Status	New

The variable declared in null at awtk/cJSON.c in line 2846 is not initialized when it is used by valuestring at awtk/cJSON.c in line 2846.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2930	2897
Object	null	valuestring

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(cJSON_bool) cJSON_Compare(const cJSON * const a, const cJSON * const b, const cJSON_bool case_sensitive)

```
....
2930.               cJSON *a_element = NULL;
....
2897.               if (strcmp(a->valuestring, b->valuestring) == 0)
```

NULL Pointer Dereference\Path 4:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=44
Status New

The variable declared in null at awtk/cJSON.c in line 2846 is not initialized when it is used by valuestring at awtk/cJSON.c in line 2846.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2930	2893
Object	null	valuestring

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(cJSON_bool) cJSON_Compare(const cJSON * const a, const cJSON * const b, const cJSON_bool case_sensitive)

```
....
2930.               cJSON *a_element = NULL;
....
2893.               if ((a->valuestring == NULL) || (b->valuestring ==
NULL))
```

NULL Pointer Dereference\Path 5:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=45
Status	New

The variable declared in null at awtk/SDL_wave.c in line 448 is not initialized when it is used by index at awtk/SDL_wave.c in line 338.

	Source	Destination
File	awtk/SDL_wave.c	awtk/SDL_wave.c
Line	612	377
Object	null	index

Code Snippet
File Name awtk/SDL_wave.c
Method SDL_LoadWAV_RW(SDL_RWops * src, int freesrc,

```
....
612.      *audio_buf = NULL;
```

File Name awtk/SDL_wave.c
Method IMA_ADPCM_decode(Uint8 ** audio_buf, Uint32 * audio_len)

```
....
377.              state[c].index = *encoded++;
```

NULL Pointer Dereference\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=46
Status	New

The variable declared in null at awtk/SDL_wave.c in line 448 is not initialized when it is used by index at awtk/SDL_wave.c in line 338.

	Source	Destination
File	awtk/SDL_wave.c	awtk/SDL_wave.c
Line	615	377
Object	null	index

Code Snippet
File Name awtk/SDL_wave.c
Method SDL_LoadWAV_RW(SDL_RWops * src, int freesrc,

```
....
615.          *audio_buf = NULL;
```

File Name awtk/SDL_wave.c

Method IMA_ADPCM_decode(Uint8 ** audio_buf, Uint32 * audio_len)

```
....
377.              state[c].index = *encoded++;
```

NULL Pointer Dereference\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=47
Status	New

The variable declared in null at awtk/SDL_wave.c in line 448 is not initialized when it is used by hPredictor at awtk/SDL_wave.c in line 119.

	Source	Destination
File	awtk/SDL_wave.c	awtk/SDL_wave.c
Line	612	150
Object	null	hPredictor

Code Snippet

File Name awtk/SDL_wave.c

Method SDL_LoadWAV_RW(SDL_RWops * src, int freesrc,

```
....
612.      *audio_buf = NULL;
```

File Name awtk/SDL_wave.c

Method MS_ADPCM_decode(Uint8 ** audio_buf, Uint32 * audio_len)

```
....
150.              state[1]->hPredictor = *encoded++;
```

NULL Pointer Dereference\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=48
Status	New

The variable declared in null at awtk/SDL_wave.c in line 448 is not initialized when it is used by hPredictor at awtk/SDL_wave.c in line 119.

	Source	Destination
File	awtk/SDL_wave.c	awtk/SDL_wave.c
Line	615	150
Object	null	hPredictor

Code Snippet
File Name awtk/SDL_wave.c
Method SDL_LoadWAV_RW(SDL_RWops * src, int freesrc,

```
....
615.           *audio_buf = NULL;
```

File Name awtk/SDL_wave.c
Method MS_ADPCM_decode(Uint8 ** audio_buf, Uint32 * audio_len)

```
....
150.               state[1]->hPredictor = *encoded++;
```

NULL Pointer Dereference\Path 9:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=49
Status New

The variable declared in null at awtk/SDL_wave.c in line 448 is not initialized when it is used by hPredictor at awtk/SDL_wave.c in line 119.

	Source	Destination
File	awtk/SDL_wave.c	awtk/SDL_wave.c
Line	612	148
Object	null	hPredictor

Code Snippet
File Name awtk/SDL_wave.c
Method SDL_LoadWAV_RW(SDL_RWops * src, int freesrc,

```
....
612.       *audio_buf = NULL;
```

File Name awtk/SDL_wave.c
Method MS_ADPCM_decode(Uint8 ** audio_buf, Uint32 * audio_len)

```
....
148.          state[0]->hPredictor = *encoded++;
```

NULL Pointer Dereference\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=50
Status	New

The variable declared in null at awtk/SDL_wave.c in line 448 is not initialized when it is used by hPredictor at awtk/SDL_wave.c in line 119.

	Source	Destination
File	awtk/SDL_wave.c	awtk/SDL_wave.c
Line	615	148
Object	null	hPredictor

Code Snippet

File Name awtk/SDL_wave.c

Method SDL_LoadWAV_RW(SDL_RWops * src, int freesrc,

```
....
615.          *audio_buf = NULL;
```

File Name awtk/SDL_wave.c

Method MS_ADPCM_decode(Uint8 ** audio_buf, Uint32 * audio_len)

```
....
148.          state[0]->hPredictor = *encoded++;
```

NULL Pointer Dereference\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=51
Status	New

The variable declared in newitem at awtk/cJSON.c in line 2570 is not initialized when it is used by valueint at awtk/cJSON.c in line 2570.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2572	2590

Object	newitem	valueint

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(cJSON *) cJSON_Duplicate(const cJSON *item, cJSON_bool recurse)

```
....
2572.       cJSON *newitem = NULL;
....
2590.       newitem->valueint = item->valueint;
```

NULL Pointer Dereference\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=52
Status	New

The variable declared in newitem at awtk/cJSON.c in line 2570 is not initialized when it is used by type at awtk/cJSON.c in line 2570.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2572	2589
Object	newitem	type

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(cJSON *) cJSON_Duplicate(const cJSON *item, cJSON_bool recurse)

```
....
2572.       cJSON *newitem = NULL;
....
2589.       newitem->type = item->type & (~cJSON_IsReference);
```

NULL Pointer Dereference\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=53
Status	New

The variable declared in newitem at awtk/cJSON.c in line 2570 is not initialized when it is used by valuedouble at awtk/cJSON.c in line 2570.

	Source	Destination

File	awtk/cJSON.c	awtk/cJSON.c
Line	2572	2591
Object	newitem	valuedouble

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(cJSON *) cJSON_Duplicate(const cJSON *item, cJSON_bool recurse)

```
....
2572.       cJSON *newitem = NULL;
....
2591.       newitem->valuedouble = item->valuedouble;
```

NULL Pointer Dereference\Path 14:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=54
Status New

The variable declared in newitem at awtk/cJSON.c in line 2570 is not initialized when it is used by valuestring at awtk/cJSON.c in line 2570.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2572	2594
Object	newitem	valuestring

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(cJSON *) cJSON_Duplicate(const cJSON *item, cJSON_bool recurse)

```
....
2572.       cJSON *newitem = NULL;
....
2594.           newitem->valuestring = (char*)cJSON_strdup((unsigned
char*)item->valuestring, &global_hooks);
```

NULL Pointer Dereference\Path 15:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=55
Status New

The variable declared in newitem at awtk/cJSON.c in line 2570 is not initialized when it is used by valuestring at awtk/cJSON.c in line 2570.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2572	2595
Object	newitem	valuestring

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(cJSON *) cJSON_Duplicate(const cJSON *item, cJSON_bool recurse)

```
....
2572.       cJSON *newitem = NULL;
....
2595.           if (!newitem->valuestring)
```

NULL Pointer Dereference\Path 16:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=56
Status New

The variable declared in newitem at awtk/cJSON.c in line 2570 is not initialized when it is used by string at awtk/cJSON.c in line 2570.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2572	2602
Object	newitem	string

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(cJSON *) cJSON_Duplicate(const cJSON *item, cJSON_bool recurse)

```
....
2572.       cJSON *newitem = NULL;
....
2602.           newitem->string = (item->type&cJSON_StringIsConst) ?
item->string : (char*)cJSON_strdup((unsigned char*)item->string,
&global_hooks);
```

NULL Pointer Dereference\Path 17:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=57
Status	New

The variable declared in newitem at awtk/cJSON.c in line 2570 is not initialized when it is used by string at awtk/cJSON.c in line 2570.

	Source	Destination
File	awtk/cJSON.c	awtk/cJSON.c
Line	2572	2603
Object	newitem	string

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(cJSON *) cJSON_Duplicate(const cJSON *item, cJSON_bool recurse)

```
....
2572.       cJSON *newitem = NULL;
....
2603.           if (!newitem->string)
```

NULL Pointer Dereference\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=58
Status	New

The variable declared in format at awtk/SDL_wave.c in line 448 is not initialized when it is used by encoding at awtk/SDL_wave.c in line 448.

	Source	Destination
File	awtk/SDL_wave.c	awtk/SDL_wave.c
Line	464	563
Object	format	encoding

Code Snippet
File Name awtk/SDL_wave.c
Method SDL_LoadWAV_RW(SDL_RWops * src, int freesrc,

```
....
464.      WaveFMT *format = NULL;
....
563.                          SDL_SwapLE16(format->encoding));
```

NULL Pointer Dereference\Path 19:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=59
Status	New

The variable declared in format at awtk/SDL_wave.c in line 448 is not initialized when it is used by SDL_SwapLE16 at awtk/SDL_wave.c in line 448.

	Source	Destination
File	awtk/SDL_wave.c	awtk/SDL_wave.c
Line	464	515
Object	format	SDL_SwapLE16

Code Snippet
File Name awtk/SDL_wave.c
Method SDL_LoadWAV_RW(SDL_RWops * src, int freesrc,

```
....
464.      WaveFMT *format = NULL;
....
515.      switch (SDL_SwapLE16(format->encoding)) {
```

NULL Pointer Dereference\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=60
Status	New

The variable declared in format at awtk/SDL_wave.c in line 448 is not initialized when it is used by frequency at awtk/SDL_wave.c in line 448.

	Source	Destination
File	awtk/SDL_wave.c	awtk/SDL_wave.c
Line	464	568
Object	format	frequency

Code Snippet
File Name awtk/SDL_wave.c
Method SDL_LoadWAV_RW(SDL_RWops * src, int freesrc,

```
....
464.      WaveFMT *format = NULL;
....
568.      spec->freq = SDL_SwapLE32(format->frequency);
```

NULL Pointer Dereference\Path 21:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=61
Status	New

The variable declared in format at awtk/SDL_wave.c in line 448 is not initialized when it is used by bitspersample at awtk/SDL_wave.c in line 448.

	Source	Destination
File	awtk/SDL_wave.c	awtk/SDL_wave.c
Line	464	571
Object	format	bitspersample

Code Snippet
File Name awtk/SDL_wave.c
Method SDL_LoadWAV_RW(SDL_RWops * src, int freesrc,

```
....
464.      WaveFMT *format = NULL;
....
571.          if ((SDL_SwapLE16(format->bitspersample)) != 32) {
```

NULL Pointer Dereference\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=62
Status	New

The variable declared in format at awtk/SDL_wave.c in line 448 is not initialized when it is used by SDL_SwapLE16 at awtk/SDL_wave.c in line 448.

	Source	Destination
File	awtk/SDL_wave.c	awtk/SDL_wave.c
Line	464	577
Object	format	SDL_SwapLE16

Code Snippet
File Name awtk/SDL_wave.c
Method SDL_LoadWAV_RW(SDL_RWops * src, int freesrc,

```
....
464.      WaveFMT *format = NULL;
....
577.          switch (SDL_SwapLE16(format->bitspersample)) {
```

NULL Pointer Dereference\Path 23:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=63
Status	New

The variable declared in format at awtk/SDL_wave.c in line 448 is not initialized when it is used by bitspersample at awtk/SDL_wave.c in line 448.

	Source	Destination
File	awtk/SDL_wave.c	awtk/SDL_wave.c
Line	464	605
Object	format	bitspersample

Code Snippet
File Name awtk/SDL_wave.c
Method SDL_LoadWAV_RW(SDL_RWops * src, int freesrc,

```
....
464.      WaveFMT *format = NULL;
....
605.                        SDL_SwapLE16(format->bitspersample));
```

NULL Pointer Dereference\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=64
Status	New

The variable declared in format at awtk/SDL_wave.c in line 448 is not initialized when it is used by channels at awtk/SDL_wave.c in line 448.

	Source	Destination
File	awtk/SDL_wave.c	awtk/SDL_wave.c
Line	464	608
Object	format	channels

Code Snippet
File Name awtk/SDL_wave.c
Method SDL_LoadWAV_RW(SDL_RWops * src, int freesrc,

```
....
464.      WaveFMT *format = NULL;
....
608.      spec->channels = (Uint8) SDL_SwapLE16(format->channels);
```

NULL Pointer Dereference\Path 25:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=65
Status	New

The variable declared in format at awtk/SDL_wave.c in line 448 is not initialized when it is used by bitspersample at awtk/SDL_wave.c in line 448.

	Source	Destination
File	awtk/SDL_wave.c	awtk/SDL_wave.c
Line	464	641
Object	format	bitspersample

Code Snippet
File Name awtk/SDL_wave.c
Method SDL_LoadWAV_RW(SDL_RWops * src, int freesrc,

```
....
464.      WaveFMT *format = NULL;
....
641.      if (SDL_SwapLE16(format->bitspersample) == 24) {
```

NULL Pointer Dereference\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=66
Status	New

The variable declared in ext at awtk/SDL_wave.c in line 448 is not initialized when it is used by subformat at awtk/SDL_wave.c in line 448.

	Source	Destination
File	awtk/SDL_wave.c	awtk/SDL_wave.c
Line	465	550
Object	ext	subformat

Code Snippet
File Name awtk/SDL_wave.c
Method SDL_LoadWAV_RW(SDL_RWops * src, int freesrc,

```
....
465.      WaveExtensibleFMT *ext = NULL;
....
550.          if (SDL_memcmp(ext->subformat, extensible_pcm_guid, 16) ==
0) {
```

NULL Pointer Dereference\Path 27:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=67
Status	New

The variable declared in ext at awtk/SDL_wave.c in line 448 is not initialized when it is used by size at awtk/SDL_wave.c in line 448.

	Source	Destination
File	awtk/SDL_wave.c	awtk/SDL_wave.c
Line	465	545
Object	ext	size

Code Snippet
File Name awtk/SDL_wave.c
Method SDL_LoadWAV_RW(SDL_RWops * src, int freesrc,

```
....
465.      WaveExtensibleFMT *ext = NULL;
....
545.          if (SDL_SwapLE16(ext->size) < 22) {
```

NULL Pointer Dereference\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=68
Status	New

The variable declared in ext at awtk/SDL_wave.c in line 448 is not initialized when it is used by subformat at awtk/SDL_wave.c in line 448.

	Source	Destination
File	awtk/SDL_wave.c	awtk/SDL_wave.c
Line	465	552
Object	ext	subformat

Code Snippet
File Name awtk/SDL_wave.c
Method SDL_LoadWAV_RW(SDL_RWops * src, int freesrc,

```
....
465.      WaveExtensibleFMT *ext = NULL;
....
552.          } else if (SDL_memcmp(ext->subformat,
extensible_ieee_guid, 16) == 0) {
```

Unchecked Return Value

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

Categories

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

Description

Unchecked Return Value\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=26
Status	New

The CJSON_PUBLIC method calls the sprintf function, at line 95 of awtk/cJSON.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	awtk/cJSON.c	awtk/cJSON.c
Line	98	98
Object	sprintf	sprintf

Code Snippet
File Name awtk/cJSON.c
Method CJSON_PUBLIC(const char*) cJSON_Version(void)

```
....
98.      sprintf(version, "%i.%i.%i", CJSON_VERSION_MAJOR,
CJSON_VERSION_MINOR, CJSON_VERSION_PATCH);
```

Unchecked Return Value\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=27
Status	New

The print_string_ptr method calls the sprintf function, at line 843 of awtk/cJSON.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	awtk/cJSON.c	awtk/cJSON.c
Line	952	952
Object	sprintf	sprintf

Code Snippet

File Name	awtk/cJSON.c
Method	static cJSON_bool print_string_ptr(const unsigned char * const input, printbuffer * const output_buffer)

```
....
952.                        sprintf((char*)output_pointer, "u%04x",
*input_pointer);
```

Unchecked Return Value\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=28
Status	New

The network_interface_linux_enable method calls the snprintf function, at line 43 of awtk/network_interface_linux.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	awtk/network_interface_linux.c	awtk/network_interface_linux.c
Line	47	47
Object	snprintf	snprintf

Code Snippet	
File Name	awtk/network_interface_linux.c
Method	static ret_t network_interface_linux_enable(network_interface_t* interface) {

```
....
47.    snprintf(command, sizeof(command), "ifconfig %s up", interface-
>interface_name);
```

Unchecked Return Value\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=29
Status	New

The network_interface_linux_disable method calls the snprintf function, at line 53 of awtk/network_interface_linux.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	awtk/network_interface_linux.c	awtk/network_interface_linux.c
Line	57	57
Object	snprintf	snprintf

Code Snippet
File Name awtk/network_interface_linux.c
Method static ret_t network_interface_linux_disable(network_interface_t* interface) {

```
....
57.    snprintf(command, sizeof(command), "ifconfig %s down", interface->interface_name);
```

Unchecked Return Value\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=30
Status	New

The network_interface_linux_get_ipaddr method calls the snprintf function, at line 63 of awtk/network_interface_linux.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	awtk/network_interface_linux.c	awtk/network_interface_linux.c
Line	79	79
Object	snprintf	snprintf

Code Snippet
File Name awtk/network_interface_linux.c
Method static char* network_interface_linux_get_ipaddr(network_interface_t* interface) {

```
....
79.    snprintf(ipstr, sizeof(ipstr), "%d.%d.%d.%d", ipaddr[0], ipaddr[1], ipaddr[2], ipaddr[3]);
```

Unchecked Return Value\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=31
Status	New

The network_interface_linux_get_macaddr method calls the snprintf function, at line 86 of awtk/network_interface_linux.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	awtk/network_interface_linux.c	awtk/network_interface_linux.c
Line	101	101

Object	snprintf	snprintf

Code Snippet
File Name awtk/network_interface_linux.c
Method static char* network_interface_linux_get_macaddr(network_interface_t* interface) {

```
....
101.    snprintf(macstr, sizeof(macstr),
"%02x:%02x:%02x:%02x:%02x:%02x", m[0], m[1], m[2], m[3], m[4],
```

Unchecked Return Value\Path 7:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=32
Status New

The network_interface_linux_eth_get_status method calls the snprintf function, at line 109 of awtk/network_interface_linux.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	awtk/network_interface_linux.c	awtk/network_interface_linux.c
Line	113	113
Object	snprintf	snprintf

Code Snippet
File Name awtk/network_interface_linux.c
Method static int network_interface_linux_eth_get_status(network_interface_t* interface) {

```
....
113.    snprintf(carrier_path, sizeof(carrier_path),
"/sys/class/net/%s/carrier",
```

Unchecked Return Value\Path 8:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=33
Status New

The network_interface_linux_eth_get_quality method calls the snprintf function, at line 128 of awtk/network_interface_linux.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	awtk/network_interface_linux.c	awtk/network_interface_linux.c
Line	133	133
Object	snprintf	snprintf

Code Snippet
File Name awtk/network_interface_linux.c
Method static int network_interface_linux_eth_get_quality(network_interface_t* interface) {

```
....
133.    snprintf(speed_path, sizeof(speed_path),
"/sys/class/net/%s/speed", interface->interface_name);
```

Unchecked Return Value\Path 9:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=34
Status New

The network_interface_linux_set_ipaddr method calls the snprintf function, at line 145 of awtk/network_interface_linux.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	awtk/network_interface_linux.c	awtk/network_interface_linux.c
Line	150	150
Object	snprintf	snprintf

Code Snippet
File Name awtk/network_interface_linux.c
Method static ret_t network_interface_linux_set_ipaddr(network_interface_t* interface, const char* ipaddr,

```
....
150.    snprintf(command, sizeof(command), "ifconfig %s  %s netmask %s",
interface->interface_name,
```

Unchecked Return Value\Path 10:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=35
Status New

The network_interface_linux_set_dns method calls the snprintf function, at line 157 of awtk/network_interface_linux.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	awtk/network_interface_linux.c	awtk/network_interface_linux.c
Line	161	161
Object	snprintf	snprintf

Code Snippet
File Name awtk/network_interface_linux.c
Method static ret_t network_interface_linux_set_dns(network_interface_t* interface, const char* dns) {

```
....
161.    snprintf(command, sizeof(command), "echo \"nameserver %s\" >
/etc/resolv.conf", dns);
```

Unchecked Return Value\Path 11:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=36
Status New

The network_interface_linux_set_dhcp method calls the snprintf function, at line 167 of awtk/network_interface_linux.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	awtk/network_interface_linux.c	awtk/network_interface_linux.c
Line	171	171
Object	snprintf	snprintf

Code Snippet
File Name awtk/network_interface_linux.c
Method static ret_t network_interface_linux_set_dhcp(network_interface_t* interface) {

```
....
171.    snprintf(command, sizeof(command), "udhcpc -i %s &", interface-
>interface_name);
```

Use of Sizeof On a Pointer Type

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

Description

Use of Sizeof On a Pointer Type\Path 1:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=37

Status	New

	Source	Destination
File	awtk/testgles.c	awtk/testgles.c
Line	28	176
Object	context	sizeof

Code Snippet
File Name awtk/testgles.c
Method static SDL_GLContext *context = NULL;

```
....
28.  static SDL_GLContext *context = NULL;
```

File Name awtk/testgles.c
Method main(int argc, char *argv[])

```
....
176.      context = (SDL_GLContext *)SDL_calloc(state->num_windows,
sizeof(context));
```

Use of Sizeof On a Pointer Type\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=38
Status	New

	Source	Destination
File	awtk/harness_argparser.c	awtk/harness_argparser.c
Line	234	234
Object	sizeof	sizeof

Code Snippet
File Name awtk/harness_argparser.c
Method ParseConfig(char* file, SDLVisualTest_HarnessState* state)

```
....
234.          argv = (char**)SDL_malloc((num_params + 1) *
sizeof(char*));
```

Use of Sizeof On a Pointer Type\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=100

	04&pathid=39
Status	New

	Source	Destination
File	awtk/rsa_internal.c	awtk/rsa_internal.c
Line	86	86
Object	sizeof	sizeof

Code Snippet
File Name awtk/rsa_internal.c
Method int mbedtls_rsa_deduce_primes(mbedtls_mpi const *N,

```
....
86.      const size_t num_primes = sizeof( primes ) / sizeof( *primes );
```

Improper Resource Access Authorization

Query Path:
CPP\Cx\CPP Low Visibility\Improper Resource Access Authorization Version:1

Categories

FISMA 2014: Identification And Authentication
NIST SP 800-53: AC-3 Access Enforcement (P1)
OWASP Top 10 2017: A2-Broken Authentication

Description

Improper Resource Access Authorization\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=167
Status	New

	Source	Destination
File	awtk/network_interface_linux.c	awtk/network_interface_linux.c
Line	117	117
Object	Address	Address

Code Snippet
File Name awtk/network_interface_linux.c
Method static int network_interface_linux_eth_get_status(network_interface_t* interface) {

```
....
117.    if (read(fd, &carrier, 1) <= 0) {
```

Improper Resource Access Authorization\Path 2:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=168
Status	New

	Source	Destination
File	awtk/network_interface_linux.c	awtk/network_interface_linux.c
Line	136	136
Object	speed	speed

Code Snippet
File Name awtk/network_interface_linux.c
Method static int network_interface_linux_eth_get_quality(network_interface_t* interface) {

```
....
136.    if (read(fd, speed, sizeof(speed)) <= 0) {
```

TOCTOU

Query Path:
CPP\Cx\CPP Low Visibility\TOCTOU Version:1

Description

TOCTOU\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=169
Status	New

The network_interface_linux_eth_get_status method in awtk/network_interface_linux.c file utilizes open that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	awtk/network_interface_linux.c	awtk/network_interface_linux.c
Line	115	115
Object	open	open

Code Snippet
File Name awtk/network_interface_linux.c
Method static int network_interface_linux_eth_get_status(network_interface_t* interface) {

```
....
115.    fd = open(carrier_path, O_RDONLY);
```

TOCTOU\Path 2:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=170
Status	New

The network_interface_linux_eth_get_quality method in awtk/network_interface_linux.c file utilizes open that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	awtk/network_interface_linux.c	awtk/network_interface_linux.c
Line	134	134
Object	open	open

Code Snippet
File Name awtk/network_interface_linux.c
Method static int network_interface_linux_eth_get_quality(network_interface_t* interface) {

```
....
134.    fd = open(speed_path, O_RDONLY);
```

Potential Off by One Error in Loops

Query Path:
CPP\Cx\CPP Heuristic\Potential Off by One Error in Loops Version:1

Categories

PCI DSS v3.2: PCI DSS (3.2) - 6.5.1 - Injection flaws - particularly SQL injection
NIST SP 800-53: SI-16 Memory Protection (P1)
OWASP Top 10 2017: A1-Injection

Description
Potential Off by One Error in Loops\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=40
Status	New

The buffer allocated by <= in awtk/miniz_tinfl.c at line 174 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	awtk/miniz_tinfl.c	awtk/miniz_tinfl.c
Line	279	279
Object	<=	<=

Code Snippet

File Name	awtk/miniz_tinfl.c
Method	tinfl_status tinfl_decompress(tinfl_decompressor *r, const mz_uint8 *pIn_buf_next, size_t *pIn_buf_size, mz_uint8 *pOut_buf_start, mz_uint8 *pOut_buf_next, size_t *pOut_buf_size, const mz_uint32 decomp_flags)

```
....
279.                    for (i = 0; i <= 143; ++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-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1010003&projectid=10004&pathid=70
Status	New

	Source	Destination
File	awtk/testgles.c	awtk/testgles.c
Line	176	176
Object	context	sizeof

Code Snippet	
File Name	awtk/testgles.c
Method	main(int argc, char *argv[])

```
....
176.      context = (SDL_GLContext *)SDL_calloc(state->num_windows,
sizeof(context));
```

Buffer Overflow StrcpyStrcat

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

Buffer Overflow IndexFromInput

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

Buffer Overflow boundcpy WrongSizeParam

Risk

What might happen

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

Cause

How does it happen

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

General Recommendations

How to avoid it

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

Source Code Examples

CPP

Overflowing Buffers

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

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

Checked Buffers

```
const int BUFFER_SIZE = 10;
const int MAX_INPUT_SIZE = 256;
```

```
char buffer[BUFFER_SIZE];

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

Integer Overflow

Risk

What might happen

Assigning large data types into smaller data types, without proper checks and explicit casting, will lead to undefined behavior and unintentional effects, such as data corruption (e.g. value wraparound, wherein maximum values become minimum values); system crashes; infinite loops; logic errors, such as bypassing of security mechanisms; or even buffer overflows leading to arbitrary code execution.

Cause

How does it happen

This flaw can occur when implicitly casting numerical data types of a larger size, into a variable with a data type of a smaller size. This forces the program to discard some bits of information from the number. Depending on how the numerical data types are stored in memory, this is often the bits with the highest value, causing substantial corruption of the stored number. Alternatively, the sign bit of a signed integer could be lost, completely reversing the intention of the number.

General Recommendations

How to avoid it

- Avoid casting larger data types to smaller types.
- Prefer promoting the target variable to a large enough data type.
- If downcasting is necessary, always check that values are valid and in range of the target type, before casting

---

Source Code Examples

CPP

Unsafe Downsize Casting

```
int unsafe_addition(short op1, int op2) {

    // op2 gets forced from int into a short
    short total = op1 + op2;

    return total;
}
```

Safer Use of Proper Data Types

```
int safe_addition(short op1, int op2) {

    // total variable is of type int, the largest type that is needed
    int total = 0;

    // check if total will overflow available integer size
    if (INT_MAX - abs(op2) > op1)
```

```
    {
        total = op1 + op2;
    }
    else
    {
        // instead of overflow, saturate (but this is not always a good thing)
        total = INT_MAX
    }

    return total;
}
```

Dangerous Functions

Risk

What might happen

Use of dangerous functions may expose varying risks associated with each particular function, with potential impact of improper usage of these functions varying significantly. The presence of such functions indicates a flaw in code maintenance policies and adherence to secure coding practices, in a way that has allowed introducing known dangerous code into the application.

Cause

How does it happen

A dangerous function has been identified within the code. Functions are often deemed dangerous to use for numerous reasons, as there are different sets of vulnerabilities associated with usage of such functions. For example, some string copy and concatenation functions are vulnerable to Buffer Overflow, Memory Disclosure, Denial of Service and more. Use of these functions is not recommended.

General Recommendations

How to avoid it

- Deploy a secure and recommended alternative to any functions that were identified as dangerous.
    - If no secure alternative is found, conduct further researching and testing to identify whether current usage successfully sanitizes and verifies values, and thus successfully avoids the use-cases for whom the function is indeed dangerous
- Conduct a periodical review of methods that are in use, to ensure that all external libraries and built-in functions are up-to-date and whose use has not been excluded from best secure coding practices.

Source Code Examples

CPP

Buffer Overflow in gets()

```
int main()
{
	char buf[10];

	printf("Please enter your name: ");
	gets(buf); // veryveryverylongname
	if (buf == ACCEPTED_NAME)
	{
		// Do something
	}
	return 0;
}
```

Safe reading from user

```
int main()
{
	char buf[10];

	printf("Please enter your name: ");
	fgets(buf, sizeof(buf), stdin); //setting the amount of bytes to read
	if (buf == ACCEPTED_NAME)
	{
		//Do something
	}
	return 0;
}
```

Unsafe function for string copy

```
int main(int argc, char* argv[])
{
	char buf[10];
	strcpy(buf, argv[1]); // overflow occurs when len(argv[1]) > 10 bytes

	return 0;
}
```

Safe string copy

```
int main(int argc, char* argv[])
{
	char buf[10];
	strncpy(buf, argv[1], sizeof(buf));
	buf[9]= '\0'; //strncpy doesn't NULL terminates

	return 0;
}
```

Unsafe format string

```
int main(int argc, char* argv[])
{
	printf(argv[1]); // If argv[1] contains a format token, such as %s,%x or %d, will cause
an access violation
	return 0;
}
```

Safe format string

```
int main(int argc, char* argv[])
{
      printf("%s", argv[1]); // Second parameter is not a formattable string
      return 0;
}
```

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

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

Potential Off by One Error in Loops

Risk

What might happen

An off by one error may result in overwriting or over-reading of unintended memory; in most cases, this can result in unexpected behavior and even application crashes. In other cases, where allocation can be controlled by an attacker, a combination of variable assignment and an off by one error can result in execution of malicious code.

Cause

How does it happen

Often when designating variables to memory, a calculation error may occur when determining size or length that is off by one.

For example in loops, when allocating an array of size 2, its cells are counted as 0,1 - therefore, if a For loop iterator on the array is incorrectly set with the start condition i=0 and the continuation condition i<=2, three cells will be accessed instead of 2, and an attempt will be made to write or read cell [2], which was not originally allocated, resulting in potential corruption of memory outside the bounds of the originally assigned array.

Another example occurs when a null-byte terminated string, in the form of a character array, is copied without its terminating null-byte. Without the null-byte, the string representation is unterminated, resulting in certain functions to over-read memory as they expect the missing null terminator.

General Recommendations

How to avoid it

- Always ensure that a given iteration boundary is correct:
  - With array iterations, consider that arrays begin with cell 0 and end with cell n-1, for a size n array.
  - With character arrays and null-byte terminated string representations, consider that the null byte is required and should not be overwritten or ignored; ensure functions in use are not vulnerable to off-by-one, specifically for instances where null-bytes are automatically appended after the buffer, instead of in place of its last character.
- Where possible, use safe functions that manage memory and are not prone to off-by-one errors.

Source Code Examples

CPP

Off-By-One in For Loop

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
for (int i = 0; i <= 5; i++)
{
      ptr[i] = i * 2 + 1; // ptr[5] will be set, but is out of bounds
```

```
}
```

Proper Iteration in For Loop

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
for (int i = 0; i < 5; i++)
{
      ptr[i] = i * 2 + 1; // ptr[0-4] are well defined
}
```

Off-By-One in strncat

```
strncat(buf, input, sizeof(buf) - strlen(buf)); // actual value should be sizeof(buf)-
strlen(buf)-1 - this form will overwrite the terminating nullbyte
```

NULL Pointer Dereference

Risk

What might happen

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

Cause

How does it happen

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

General Recommendations

How to avoid it

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

Source Code Examples

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());
```

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 Access Control (Authorization)

Weakness ID: 285 *(Weakness Class)* **Status:** Draft

Description

Description Summary

The software does not perform or incorrectly performs access control checks across all potential execution paths.

Extended Description

When access control checks are not applied consistently - or not at all - users are able to access data or perform actions that they should not be allowed to perform. This can lead to a wide range of problems, including information leaks, denial of service, and arbitrary code execution.

Alternate Terms

AuthZ:	"AuthZ" is typically used as an abbreviation of "authorization" within the web application security community. It is also distinct from "AuthC," which is an abbreviation of "authentication." The use of "Auth" as an abbreviation is discouraged, since it could be used for either authentication or authorization.

Time of Introduction

- Architecture and Design
- Implementation
- Operation

Applicable Platforms

Languages

Language-independent

Technology Classes

Web-Server: *(Often)*

Database-Server: *(Often)*

Modes of Introduction

A developer may introduce authorization weaknesses because of a lack of understanding about the underlying technologies. For example, a developer may assume that attackers cannot modify certain inputs such as headers or cookies.

Authorization weaknesses may arise when a single-user application is ported to a multi-user environment.

Common Consequences

Scope	Effect
Confidentiality	An attacker could read sensitive data, either by reading the data directly from a data store that is not properly restricted, or by accessing insufficiently-protected, privileged functionality to read the data.
Integrity	An attacker could modify sensitive data, either by writing the data directly to a data store that is not properly restricted, or by accessing insufficiently-protected, privileged functionality to write the data.
Integrity	An attacker could gain privileges by modifying or reading critical data directly, or by accessing insufficiently-protected, privileged functionality.

Likelihood of Exploit

High

Detection Methods

Automated Static Analysis

Automated static analysis is useful for detecting commonly-used idioms for authorization. A tool may be able to analyze related configuration files, such as .htaccess in Apache web servers, or detect the usage of commonly-used authorization libraries.

Generally, automated static analysis tools have difficulty detecting custom authorization schemes. In addition, the software's design may include some functionality that is accessible to any user and does not require an authorization check; an automated technique that detects the absence of authorization may report false positives.

Effectiveness: Limited

Automated Dynamic Analysis

Automated dynamic analysis may find many or all possible interfaces that do not require authorization, but manual analysis is required to determine if the lack of authorization violates business logic

Manual Analysis

This weakness can be detected using tools and techniques that require manual (human) analysis, such as penetration testing, threat modeling, and interactive tools that allow the tester to record and modify an active session.

Specifically, manual static analysis is useful for evaluating the correctness of custom authorization mechanisms.

Effectiveness: Moderate

These may be more effective than strictly automated techniques. This is especially the case with weaknesses that are related to design and business rules. However, manual efforts might not achieve desired code coverage within limited time constraints.

Demonstrative Examples

Example 1

The following program could be part of a bulletin board system that allows users to send private messages to each other. This program intends to authenticate the user before deciding whether a private message should be displayed. Assume that LookupMessageObject() ensures that the $id argument is numeric, constructs a filename based on that id, and reads the message details from that file. Also assume that the program stores all private messages for all users in the same directory.

(Bad Code)

Example Language: **Perl**

```
sub DisplayPrivateMessage {
my($id) = @_;
my $Message = LookupMessageObject($id);
print "From: " . encodeHTML($Message->{from}) . "<br>\n";
print "Subject: " . encodeHTML($Message->{subject}) . "\n";
print "<hr>\n";
print "Body: " . encodeHTML($Message->{body}) . "\n";
}

my $q = new CGI;
# For purposes of this example, assume that CWE-309 and
# CWE-523 do not apply.
if (! AuthenticateUser($q->param('username'), $q->param('password'))) {
ExitError("invalid username or password");
}

my $id = $q->param('id');
DisplayPrivateMessage($id);
```

While the program properly exits if authentication fails, it does not ensure that the message is addressed to the user. As a result, an authenticated attacker could provide any arbitrary identifier and read private messages that were intended for other users.

One way to avoid this problem would be to ensure that the "to" field in the message object matches the username of the authenticated user.

Observed Examples

Reference	Description
CVE-2009-3168	Web application does not restrict access to admin scripts, allowing authenticated users to reset administrative passwords.

CVE-2009-2960	Web application does not restrict access to admin scripts, allowing authenticated users to modify passwords of other users.
CVE-2009-3597	Web application stores database file under the web root with insufficient access control (CWE-219), allowing direct request.
CVE-2009-2282	Terminal server does not check authorization for guest access.
CVE-2009-3230	Database server does not use appropriate privileges for certain sensitive operations.
CVE-2009-2213	Gateway uses default "Allow" configuration for its authorization settings.
CVE-2009-0034	Chain: product does not properly interpret a configuration option for a system group, allowing users to gain privileges.
CVE-2008-6123	Chain: SNMP product does not properly parse a configuration option for which hosts are allowed to connect, allowing unauthorized IP addresses to connect.
CVE-2008-5027	System monitoring software allows users to bypass authorization by creating custom forms.
CVE-2008-7109	Chain: reliance on client-side security (CWE-602) allows attackers to bypass authorization using a custom client.
CVE-2008-3424	Chain: product does not properly handle wildcards in an authorization policy list, allowing unintended access.
CVE-2009-3781	Content management system does not check access permissions for private files, allowing others to view those files.
CVE-2008-4577	ACL-based protection mechanism treats negative access rights as if they are positive, allowing bypass of intended restrictions.
CVE-2008-6548	Product does not check the ACL of a page accessed using an "include" directive, allowing attackers to read unauthorized files.
CVE-2007-2925	Default ACL list for a DNS server does not set certain ACLs, allowing unauthorized DNS queries.
CVE-2006-6679	Product relies on the X-Forwarded-For HTTP header for authorization, allowing unintended access by spoofing the header.
CVE-2005-3623	OS kernel does not check for a certain privilege before setting ACLs for files.
CVE-2005-2801	Chain: file-system code performs an incorrect comparison (CWE-697), preventing defauls ACLs from being properly applied.
CVE-2001-1155	Chain: product does not properly check the result of a reverse DNS lookup because of operator precedence (CWE-783), allowing bypass of DNS-based access restrictions.

Potential Mitigations

Phase: Architecture and Design

Divide your application into anonymous, normal, privileged, and administrative areas. Reduce the attack surface by carefully mapping roles with data and functionality. Use role-based access control (RBAC) to enforce the roles at the appropriate boundaries.

Note that this approach may not protect against horizontal authorization, i.e., it will not protect a user from attacking others with the same role.

Phase: Architecture and Design

Ensure that you perform access control checks related to your business logic. These checks may be different than the access control checks that you apply to more generic resources such as files, connections, processes, memory, and database records. For example, a database may restrict access for medical records to a specific database user, but each record might only be intended to be accessible to the patient and the patient's doctor.

Phase: Architecture and Design

Strategy: Libraries or Frameworks

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness

easier to avoid.

For example, consider using authorization frameworks such as the JAAS Authorization Framework and the OWASP ESAPI Access Control feature.

Phase: Architecture and Design

For web applications, make sure that the access control mechanism is enforced correctly at the server side on every page. Users should not be able to access any unauthorized functionality or information by simply requesting direct access to that page.

One way to do this is to ensure that all pages containing sensitive information are not cached, and that all such pages restrict access to requests that are accompanied by an active and authenticated session token associated with a user who has the required permissions to access that page.

Phases: System Configuration; Installation

Use the access control capabilities of your operating system and server environment and define your access control lists accordingly. Use a "default deny" policy when defining these ACLs.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	254	Security Features	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Weakness Class	284	Access Control (Authorization) Issues	**Development Concepts (primary)699 Research Concepts (primary)1000**
ChildOf	Category	721	OWASP Top Ten 2007 Category A10 - Failure to Restrict URL Access	**Weaknesses in OWASP Top Ten (2007) (primary)629**
ChildOf	Category	723	OWASP Top Ten 2004 Category A2 - Broken Access Control	**Weaknesses in OWASP Top Ten (2004) (primary)711**
ChildOf	Category	753	2009 Top 25 - Porous Defenses	**Weaknesses in the 2009 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)750**
ChildOf	Category	803	2010 Top 25 - Porous Defenses	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
ParentOf	Weakness Variant	219	Sensitive Data Under Web Root	**Research Concepts (primary)1000**
ParentOf	Weakness Base	551	Incorrect Behavior Order: Authorization Before Parsing and Canonicalization	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Class	638	Failure to Use Complete Mediation	Research Concepts1000
ParentOf	Weakness Base	804	Guessable CAPTCHA	**Development Concepts (primary)699 Research Concepts (primary)1000**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
7 Pernicious Kingdoms			Missing Access Control
OWASP Top Ten 2007	A10	CWE More Specific	Failure to Restrict URL Access
OWASP Top Ten 2004	A2	CWE More Specific	Broken Access Control

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
1	Accessing Functionality Not Properly Constrained by ACLs	
13	Subverting Environment Variable Values	

17	Accessing, Modifying or Executing Executable Files
87	Forceful Browsing
39	Manipulating Opaque Client-based Data Tokens
45	Buffer Overflow via Symbolic Links
51	Poison Web Service Registry
59	Session Credential Falsification through Prediction
60	Reusing Session IDs (aka Session Replay)
77	Manipulating User-Controlled Variables
76	Manipulating Input to File System Calls
104	Cross Zone Scripting

References

NIST. "Role Based Access Control and Role Based Security". <http://csrc.nist.gov/groups/SNS/rbac/>.

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 4, "Authorization" Page 114; Chapter 6, "Determining Appropriate Access Control" Page 171. 2nd Edition. Microsoft. 2002.

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	7 Pernicious Kingdoms		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-15		Veracode	External
	Suggested OWASP Top Ten 2004 mapping		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Relationships, Other Notes, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences, Description, Likelihood of Exploit, Name, Other Notes, Potential Mitigations, References, Relationships		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Potential Mitigations		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Description, Related Attack Patterns		
2009-07-27	CWE Content Team	MITRE	Internal
	updated Relationships		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Type		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Demonstrative Examples, Detection Factors, Modes of Introduction, Observed Examples, Relationships		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Alternate Terms, Detection Factors, Potential Mitigations, References, Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Potential Mitigations		
Previous Entry Names			
Change Date	**Previous Entry Name**		
2009-01-12	Missing or Inconsistent Access Control		

BACK TO TOP

TOCTOU

Risk

What might happen

At best, a Race Condition may cause errors in accuracy, overidden values or unexpected behavior that may result in denial-of-service. At worst, it may allow attackers to retrieve data or bypass security processes by replaying a controllable Race Condition until it plays out in their favor.

Cause

How does it happen

Race Conditions occur when a public, single instance of a resource is used by multiple concurrent logical processes. If the these logical processes attempt to retrieve and update the resource without a timely management system, such as a lock, a Race Condition will occur.

An example for when a Race Condition occurs is a resource that may return a certain value to a process for further editing, and then updated by a second process, resulting in the original process' data no longer being valid. Once the original process edits and updates the incorrect value back into the resource, the second process' update has been overwritten and lost.

General Recommendations

How to avoid it

When sharing resources between concurrent processes across the application ensure that these resources are either thread-safe, or implement a locking mechanism to ensure expected concurrent activity.

Source Code Examples

Java

Different Threads Increment and Decrement The Same Counter Repeatedly, Resulting in a Race Condition

```
	public static int counter = 0;
	public static void start() throws InterruptedException {
		incrementCounter ic;
		decrementCounter dc;
		while(counter == 0) {
			counter = 0;
			ic = new incrementCounter();
			dc = new decrementCounter();
			ic.start();
			dc.start();
			ic.join();
			dc.join();
		}
		System.out.println(counter); //Will stop and return either -1 or 1 due to race
condition over counter
	}

	public static class incrementCounter extends Thread {
	    public void run() {
		counter++;
	    }
```

```
    }

    public static class decrementCounter extends Thread {
        public void run() {
           counter--;
        }
    }
```

Different Threads Increment and Decrement The Same Thread-Safe Counter Repeatedly, Never Resulting in a Race Condition

```
    public static int counter = 0;
    public static Object lock = new Object();

    public static void start() throws InterruptedException {
           incrementCounter ic;
           decrementCounter dc;
           while(counter == 0) { // because of proper locking, this condition is never false
                  counter = 0;
                  ic = new incrementCounter();
                  dc = new decrementCounter();
                  ic.start();
                  dc.start();
                  ic.join();
                  dc.join();
           }
           System.out.println(counter); // Never reached
    }

    public static class incrementCounter extends Thread {
        public void run() {
           synchronized (lock) {
                  counter++;
           }
        }
    }

    public static class decrementCounter extends Thread {
        public void run() {
           synchronized (lock) {
                  counter--;
           }
        }
    }
```

Scanned Languages

Language	Hash Number	Change Date
CPP	4541647240435660	6/19/2024
Common	0105849645654507	6/19/2024