h2o Scan Report

Project Name	h2o
Scan Start	Friday, June 21, 2024 11:18:44 PM
Preset	Checkmarx Default
Scan Time	00h:05m:29s
Lines Of Code Scanned	46703
Files Scanned	25
Report Creation Time	Friday, June 21, 2024 11:24:32 PM
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061
Team	CxServer
Checkmarx Version	8.7.0
Scan Type	Full
Source Origin	LocalPath
Density	1/100 (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

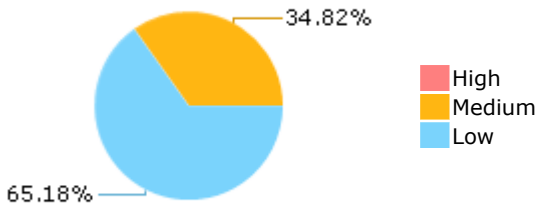
Results Limit

Results limit per query was set to 50

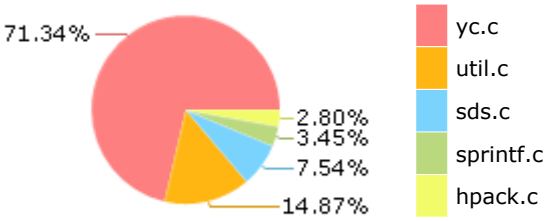
Selected Queries

Selected queries are listed in Result Summary

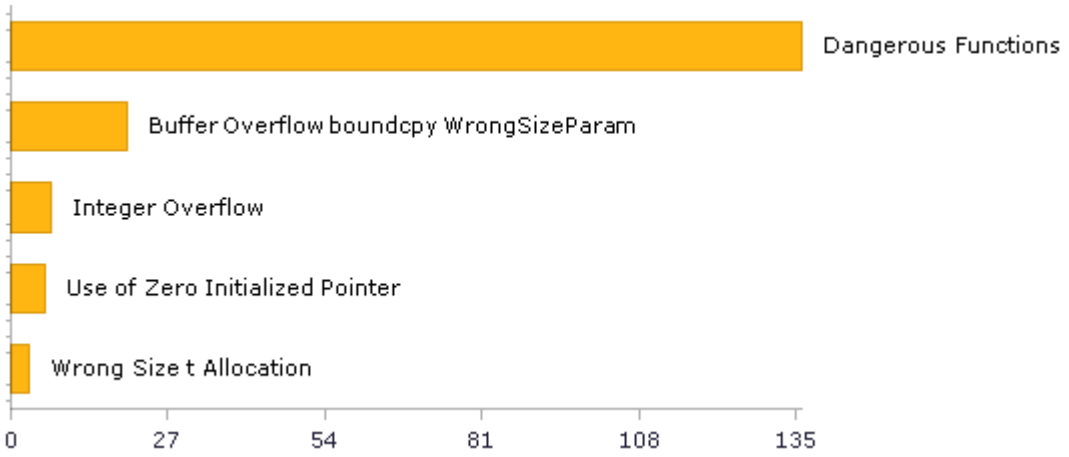
Result Summary



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	23	23
A2-Broken Authentication	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	235	235
A3-Sensitive Data Exposure	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	App. Specific	0	0
A4-XML External Entities (XXE)	App. Specific	AVERAGE	COMMON	EASY	SEVERE	App. Specific	0	0
A5-Broken Access Control*	App. Specific	AVERAGE	COMMON	AVERAGE	SEVERE	App. Specific	1	1
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	136	136
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	1	1
A5-Security Misconfiguration	EXTERNAL, INTERNAL, ADMIN USERS	EASY	COMMON	EASY	MODERATE	ALL DATA AND SYSTEM	0	0
A6-Sensitive Data Exposure	EXTERNAL, INTERNAL, ADMIN USERS, USERS BROWSERS	DIFFICULT	UNCOMMON	AVERAGE	SEVERE	EXPOSED DATA	0	0
A7-Missing Function Level Access Control*	EXTERNAL, INTERNAL USERS	EASY	COMMON	AVERAGE	MODERATE	EXPOSED DATA AND FUNCTIONS	0	0
A8-Cross-Site Request Forgery (CSRF)	USERS BROWSERS	AVERAGE	COMMON	EASY	MODERATE	AFFECTED DATA AND FUNCTIONS	0	0
A9-Using Components with Known Vulnerabilities*	EXTERNAL USERS, AUTOMATED TOOLS	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	AFFECTED DATA AND FUNCTIONS	136	136
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	2	2
PCI DSS (3.2) - 6.5.2 - Buffer overflows	27	27
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.	73	73
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.	236	236
Media Protection	Organizations must: (i) protect information system media, both paper and digital; (ii) limit access to information on information system media to authorized users; and (iii) sanitize or destroy information system media before disposal or release for reuse.	0	0
System And Communications Protection	Organizations must: (i) monitor, control, and protect organizational communications (i.e., information transmitted or received by organizational information systems) at the external boundaries and key internal boundaries of the information systems; and (ii) employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational information systems.	0	0
System And Information Integrity	Organizations must: (i) identify, report, and correct information and information system flaws in a timely manner; (ii) provide protection from malicious code at appropriate locations within organizational information systems; and (iii) monitor information system security alerts and advisories and take appropriate actions in response.	7	7

***** 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)	308	308
AC-4 Information Flow Enforcement (P1)	0	0
AC-6 Least Privilege (P1)	0	0
AU-9 Protection of Audit Information (P1)	0	0
CM-6 Configuration Settings (P2)	0	0
IA-5 Authenticator Management (P1)	0	0
IA-6 Authenticator Feedback (P2)	0	0
IA-8 Identification and Authentication (Non-Organizational Users) (P1)	0	0
SC-12 Cryptographic Key Establishment and Management (P1)	0	0
SC-13 Cryptographic Protection (P1)	0	0
SC-17 Public Key Infrastructure Certificates (P1)	0	0
SC-18 Mobile Code (P2)	0	0
SC-23 Session Authenticity (P1)*	0	0
SC-28 Protection of Information at Rest (P1)	1	1
SC-4 Information in Shared Resources (P1)	0	0
SC-5 Denial of Service Protection (P1)*	7	5
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*	10	10
SI-11 Error Handling (P2)*	2	2
SI-15 Information Output Filtering (P0)	0	0
SI-16 Memory Protection (P1)	2	2

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

Scan Summary - OWASP Mobile Top 10 2016

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

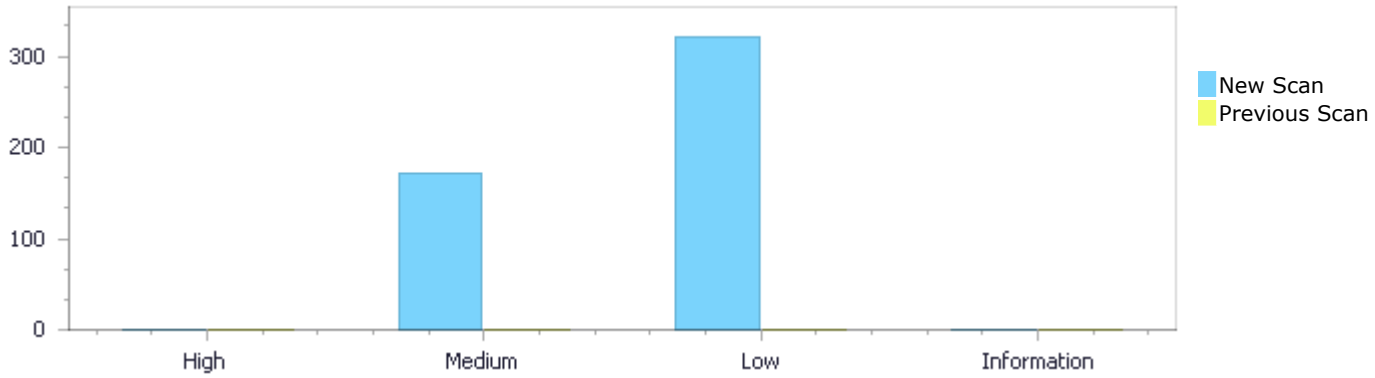
	modify the application's data and resources. This can provide the attacker a direct method of subverting the intended use of the software for personal or monetary gain.		
M9-Reverse Engineering	This category includes analysis of the final core binary to determine its source code, libraries, algorithms,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

Scan Summary - Custom

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

Results Distribution By Status First scan of the project

	High	Medium	Low	Information	Total
New Issues	0	172	322	0	494
Recurrent Issues	0	0	0	0	0
Total	0	172	322	0	494
Fixed Issues	0	0	0	0	0



Results Distribution By State

	High	Medium	Low	Information	Total
Confirmed	0	0	0	0	0
Not Exploitable	0	0	0	0	0
To Verify	0	172	322	0	494
Urgent	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0
Total	0	172	322	0	494

Result Summary

Vulnerability Type	Occurrences	Severity
Dangerous Functions	136	Medium
Buffer Overflow boundcpy WrongSizeParam	20	Medium
Integer Overflow	7	Medium
Use of Zero Initialized Pointer	6	Medium
Wrong Size t Allocation	3	Medium

Improper Resource Access Authorization	235	Low
Exposure of System Data to Unauthorized Control Sphere	73	Low
Unchecked Array Index	3	Low
Potential Off by One Error in Loops	2	Low
Unchecked Return Value	2	Low
Use of Sizeof On a Pointer Type	2	Low
Inconsistent Implementations	1	Low
Information Exposure Through Comments	1	Low
NULL Pointer Dereference	1	Low
Potential Path Traversal	1	Low
TOCTOU	1	Low

10 Most Vulnerable Files

High and Medium Vulnerabilities

File Name	Issues Found
h2o/util.c	67
h2o/sds.c	30
h2o/yc.c	26
h2o/sprintf.c	16
h2o/ffx.c	12
h2o/hpack.c	12
h2o/proc.c	4
h2o/fusion.c	4
h2o/time.c	1

Scan Results Details

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=1050071&projectid=50061&pathid=43
Status	New

The dangerous function, memcpy, was found in use at line 202 in h2o/ffx.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/ffx.c	h2o/ffx.c
Line	206	206
Object	memcpy	memcpy

Code Snippet
File Name h2o/ffx.c
Method static void ffx_init(struct st_ptls_cipher_context_t *_ctx, const void *iv)

```
....
206.      memcpy(ctx->tweaks, iv, 16);
```

Dangerous Functions\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=44
Status	New

The dangerous function, memcpy, was found in use at line 119 in h2o/ffx.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/ffx.c	h2o/ffx.c
Line	125	125

Object	memcpy	memcpy

Code Snippet
File Name h2o/ffx.c
Method static void ptls_ffx_one_pass(ptls_cipher_context_t *enc_ctx, uint8_t *source, size_t source_size, uint8_t *target,

```
....
125.      memcpy(iv, tweaks, 16);
```

Dangerous Functions\Path 3:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=45
Status New

The dangerous function, memcpy, was found in use at line 138 in h2o/ffx.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/ffx.c	h2o/ffx.c
Line	154	154
Object	memcpy	memcpy

Code Snippet
File Name h2o/ffx.c
Method static void ffx_encrypt(ptls_cipher_context_t *_ctx, void *output, const void *input, size_t len)

```
....
154.      memcpy(left, input, ctx->nb_left);
```

Dangerous Functions\Path 4:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=46
Status New

The dangerous function, memcpy, was found in use at line 138 in h2o/ffx.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/ffx.c	h2o/ffx.c
Line	155	155

Object	memcpy	memcpy

Code Snippet
File Name h2o/ffx.c
Method static void ffx_encrypt(ptls_cipher_context_t *_ctx, void *output, const void *input, size_t len)

```
....
155.      memcpy(right, ((uint8_t *)input) + ctx->nb_left, ctx-
>nb_right);
```

Dangerous Functions\Path 5:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=47
Status New

The dangerous function, memcpy, was found in use at line 138 in h2o/ffx.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/ffx.c	h2o/ffx.c
Line	190	190
Object	memcpy	memcpy

Code Snippet
File Name h2o/ffx.c
Method static void ffx_encrypt(ptls_cipher_context_t *_ctx, void *output, const void *input, size_t len)

```
....
190.      memcpy(output, left, ctx->nb_left);
```

Dangerous Functions\Path 6:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=48
Status New

The dangerous function, memcpy, was found in use at line 138 in h2o/ffx.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/ffx.c	h2o/ffx.c
Line	195	195

Object	memcpy	memcpy

Code Snippet
File Name h2o/ffx.c
Method static void ffx_encrypt(ptls_cipher_context_t *_ctx, void *output, const void *input, size_t len)

```
....
195.      memcpy(((uint8_t *)output) + ctx->nb_left, right, ctx->nb_right);
```

Dangerous Functions\Path 7:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=49
Status New

The dangerous function, memcpy, was found in use at line 52 in h2o/fusion.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/fusion.c	h2o/fusion.c
Line	58	58
Object	memcpy	memcpy

Code Snippet
File Name h2o/fusion.c
Method static void test_loadn128(void)

```
....
58.          memcpy(src, "hello world12345", 16);
```

Dangerous Functions\Path 8:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=50
Status New

The dangerous function, memcpy, was found in use at line 200 in h2o/hpack.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/hpack.c	h2o/hpack.c
Line	232	232

Object	memcpy	memcpy

Code Snippet
File Name h2o/hpack.c
Method static h2o_iovec_t *decode_string(h2o_mem_pool_t *pool, unsigned *soft_errors, const uint8_t **src, const uint8_t *src_end,

```
....
232.          memcpy(ret->base, *src, len);
```

Dangerous Functions\Path 9:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=51
Status New

The dangerous function, memcpy, was found in use at line 445 in h2o/hpack.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/hpack.c	h2o/hpack.c
Line	457	457
Object	memcpy	memcpy

Code Snippet
File Name h2o/hpack.c
Method static uint8_t *encode_content_length(uint8_t *dst, size_t value)

```
....
457.      memcpy(dst, p, l);
```

Dangerous Functions\Path 10:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=52
Status New

The dangerous function, memcpy, was found in use at line 714 in h2o/hpack.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/hpack.c	h2o/hpack.c
Line	719	719
Object	memcpy	memcpy

Code Snippet	
File Name	h2o/hpack.c
Method	static size_t encode_as_is(uint8_t *dst, const char *s, size_t len)

```
....
719.      memcpy(dst, s, len);
```

Dangerous Functions\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=53
Status	New

The dangerous function, memcpy, was found in use at line 724 in h2o/hpack.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/hpack.c	h2o/hpack.c
Line	739	739
Object	memcpy	memcpy

Code Snippet	
File Name	h2o/hpack.c
Method	size_t h2o_hpack_encode_string(uint8_t *dst, const char *s, size_t len)

```
....
739.                  memcpy(dst, head, head_len);
```

Dangerous Functions\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=54
Status	New

The dangerous function, memcpy, was found in use at line 766 in h2o/hpack.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/hpack.c	h2o/hpack.c
Line	836	836
Object	memcpy	memcpy

Code Snippet	
File Name	h2o/hpack.c

Method	static uint8_t *do_encode_header(h2o_hpack_header_table_t *header_table, uint8_t *dst, const h2o_iovec_t *name,

```
....
836.                    memcpy(entry->name->base, name->base, name->len);
```

Dangerous Functions\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=55
Status	New

The dangerous function, memcpy, was found in use at line 766 in h2o/hpack.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/hpack.c	h2o/hpack.c
Line	840	840
Object	memcpy	memcpy

Code Snippet	
File Name	h2o/hpack.c
Method	static uint8_t *do_encode_header(h2o_hpack_header_table_t *header_table, uint8_t *dst, const h2o_iovec_t *name,

```
....
840.                memcpy(entry->value->base, value->base, value->len);
```

Dangerous Functions\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=56
Status	New

The dangerous function, memcpy, was found in use at line 81 in h2o/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	132	132
Object	memcpy	memcpy

Code Snippet	
File Name	h2o/sds.c
Method	sds sdsnewlen(const void *init, size_t initlen) {

```
....
132.          memcpy(s, init, initlen);
```

Dangerous Functions\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=57
Status	New

The dangerous function, memcpy, was found in use at line 194 in h2o/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	229	229
Object	memcpy	memcpy

Code Snippet
File Name h2o/sds.c
Method sds sdsMakeRoomFor(sds s, size_t addlen) {

```
....
229.          memcpy((char*)newsh+hdrlen, s, len+1);
```

Dangerous Functions\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=58
Status	New

The dangerous function, memcpy, was found in use at line 245 in h2o/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	261	261
Object	memcpy	memcpy

Code Snippet
File Name h2o/sds.c
Method sds sdsRemoveFreeSpace(sds s) {

```
....
261.          memcpy((char*)newsh+hdrlen, s, len+1);
```

Dangerous Functions\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=59
Status	New

The dangerous function, memcpy, was found in use at line 376 in h2o/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	381	381
Object	memcpy	memcpy

Code Snippet
File Name h2o/sds.c
Method sds sdscatlen(sds s, const void *t, size_t len) {

```
....
381.      memcpy(s+curlen, t, len);
```

Dangerous Functions\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=60
Status	New

The dangerous function, memcpy, was found in use at line 405 in h2o/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	410	410
Object	memcpy	memcpy

Code Snippet
File Name h2o/sds.c
Method sds sdscpylen(sds s, const char *t, size_t len) {

```
....
410.      memcpy(s, t, len);
```

Dangerous Functions\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=61
Status	New

The dangerous function, memcpy, was found in use at line 579 in h2o/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	609	609
Object	memcpy	memcpy

Code Snippet
File Name h2o/sds.c
Method sds sdscatfmt(sds s, char const *fmt, ...) {

```
....
609.                    memcpy(s+i,str,l);
```

Dangerous Functions\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=62
Status	New

The dangerous function, memcpy, was found in use at line 579 in h2o/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	625	625
Object	memcpy	memcpy

Code Snippet
File Name h2o/sds.c
Method sds sdscatfmt(sds s, char const *fmt, ...) {

```
....
625.                        memcpy(s+i,buf,l);
```

Dangerous Functions\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=63
Status	New

The dangerous function, memcpy, was found in use at line 579 in h2o/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	642	642
Object	memcpy	memcpy

Code Snippet
File Name h2o/sds.c
Method sds sdscatfmt(sds s, char const *fmt, ...) {

```
....
642.                        memcpy(s+i,buf,l);
```

Dangerous Functions\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=64
Status	New

The dangerous function, memcpy, was found in use at line 558 in h2o/sprintf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sprintf.c	h2o/sprintf.c
Line	607	607
Object	memcpy	memcpy

Code Snippet
File Name h2o/sprintf.c
Method mrb_str_format(mrb_state *mrb, mrb_int argc, const mrb_value *argv, mrb_value fmt)

```
....
607.      PUSH(p, t - p);
```

Dangerous Functions\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=65
Status	New

The dangerous function, memcpy, was found in use at line 558 in h2o/sprintf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sprintf.c	h2o/sprintf.c
Line	733	733
Object	memcpy	memcpy

Code Snippet
File Name h2o/sprintf.c
Method mrb_str_format(mrb_state *mrb, mrb_int argc, const mrb_value *argv, mrb_value fmt)

```
....
733.          PUSH("%", 1);
```

Dangerous Functions\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=66
Status	New

The dangerous function, memcpy, was found in use at line 558 in h2o/sprintf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sprintf.c	h2o/sprintf.c
Line	773	773
Object	memcpy	memcpy

Code Snippet
File Name h2o/sprintf.c
Method mrb_str_format(mrb_state *mrb, mrb_int argc, const mrb_value *argv, mrb_value fmt)

```
....
773.            PUSH(c, n);
```

Dangerous Functions\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=67
Status	New

The dangerous function, memcpy, was found in use at line 558 in h2o/sprintf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sprintf.c	h2o/sprintf.c
Line	776	776
Object	memcpy	memcpy

Code Snippet

File Name h2o/sprintf.c

Method mrb_str_format(mrb_state *mrb, mrb_int argc, const mrb_value *argv, mrb_value fmt)

```
....
776.            PUSH(c, n);
```

Dangerous Functions\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=68
Status	New

The dangerous function, memcpy, was found in use at line 558 in h2o/sprintf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sprintf.c	h2o/sprintf.c
Line	781	781
Object	memcpy	memcpy

Code Snippet

File Name h2o/sprintf.c

Method mrb_str_format(mrb_state *mrb, mrb_int argc, const mrb_value *argv, mrb_value fmt)

```
....
781.            PUSH(c, n);
```

Dangerous Functions\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=69
Status	New

The dangerous function, memcpy, was found in use at line 558 in h2o/sprintf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sprintf.c	h2o/sprintf.c
Line	821	821
Object	memcpy	memcpy

Code Snippet

File Name h2o/sprintf.c

Method mrb_str_format(mrb_state *mrb, mrb_int argc, const mrb_value *argv, mrb_value fmt)

```
....
821.              PUSH(RSTRING_PTR(str), len);
```

Dangerous Functions\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=70
Status	New

The dangerous function, memcpy, was found in use at line 558 in h2o/sprintf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sprintf.c	h2o/sprintf.c
Line	828	828
Object	memcpy	memcpy

Code Snippet

File Name h2o/sprintf.c

Method mrb_str_format(mrb_state *mrb, mrb_int argc, const mrb_value *argv, mrb_value fmt)

```
....
828.          PUSH(RSTRING_PTR(str), len);
```

Dangerous Functions\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=71
Status	New

The dangerous function, memcpy, was found in use at line 558 in h2o/sprintf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sprintf.c	h2o/sprintf.c
Line	978	978
Object	memcpy	memcpy

Code Snippet

File Name h2o/sprintf.c

Method mrb_str_format(mrb_state *mrb, mrb_int argc, const mrb_value *argv, mrb_value fmt)

```
....
978.          if (sc) PUSH(&sc, 1);
```

Dangerous Functions\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=72
Status	New

The dangerous function, memcpy, was found in use at line 558 in h2o/sprintf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sprintf.c	h2o/sprintf.c
Line	982	982
Object	memcpy	memcpy

Code Snippet

File Name h2o/sprintf.c

Method mrb_str_format(mrb_state *mrb, mrb_int argc, const mrb_value *argv, mrb_value fmt)

```
....
982.            PUSH(prefix, plen);
```

Dangerous Functions\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=73
Status	New

The dangerous function, memcpy, was found in use at line 558 in h2o/sprintf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sprintf.c	h2o/sprintf.c
Line	987	987
Object	memcpy	memcpy

Code Snippet

File Name h2o/sprintf.c

Method mrb_str_format(mrb_state *mrb, mrb_int argc, const mrb_value *argv, mrb_value fmt)

```
....
987.            PUSH("..", 2);
```

Dangerous Functions\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=74
Status	New

The dangerous function, memcpy, was found in use at line 558 in h2o/sprintf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sprintf.c	h2o/sprintf.c
Line	1000	1000
Object	memcpy	memcpy

Code Snippet

File Name h2o/sprintf.c

Method mrb_str_format(mrb_state *mrb, mrb_int argc, const mrb_value *argv, mrb_value fmt)

```
....
1000.           PUSH(s, len);
```

Dangerous Functions\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=75
Status	New

The dangerous function, sprintf, was found in use at line 414 in h2o/hpack.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/hpack.c	h2o/hpack.c
Line	437	437
Object	sprintf	sprintf

Code Snippet
File Name h2o/hpack.c
Method static uint8_t *encode_status(uint8_t *dst, int status)

```
....
437.           sprintf((char *)dst, "%d", status);
```

Dangerous Functions\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=76
Status	New

The dangerous function, strcpy, was found in use at line 26 in h2o/util.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/util.c	h2o/util.c
Line	33	33
Object	strcpy	strcpy

Code Snippet
File Name h2o/util.c
Method static void test_parse_proxy_line(void)

```
....
33.      strcpy(in, "");
```

Dangerous Functions\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=77
Status	New

The dangerous function, strcpy, was found in use at line 26 in h2o/util.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/util.c	h2o/util.c
Line	37	37
Object	strcpy	strcpy

Code Snippet
File Name h2o/util.c
Method static void test_parse_proxy_line(void)

```
....
37.      strcpy(in, "PROXY TCP4 192.168.0.1 192.168.0.11 56324
443\r\nabc");
```

Dangerous Functions\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=78
Status	New

The dangerous function, strcpy, was found in use at line 26 in h2o/util.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/util.c	h2o/util.c
Line	45	45
Object	strcpy	strcpy

Code Snippet
File Name h2o/util.c
Method static void test_parse_proxy_line(void)

```
....
45.      strcpy(in, "PROXY TCP4 192.168.0.1 192.168.0.11 56324 443\r");
```

Dangerous Functions\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=79
Status	New

The dangerous function, strcpy, was found in use at line 26 in h2o/util.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/util.c	h2o/util.c
Line	49	49
Object	strcpy	strcpy

Code Snippet
File Name h2o/util.c
Method static void test_parse_proxy_line(void)

```
....
49.      strcpy(in, "PROXY TCP5");
```

Dangerous Functions\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=80
Status	New

The dangerous function, strcpy, was found in use at line 26 in h2o/util.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/util.c	h2o/util.c
Line	53	53
Object	strcpy	strcpy

Code Snippet
File Name h2o/util.c
Method static void test_parse_proxy_line(void)

```
....
53.      strcpy(in, "PROXY UNKNOWN");
```

Dangerous Functions\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=81
Status	New

The dangerous function, strcpy, was found in use at line 26 in h2o/util.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/util.c	h2o/util.c
Line	57	57
Object	strcpy	strcpy

Code Snippet
File Name h2o/util.c
Method static void test_parse_proxy_line(void)

```
....
57.      strcpy(in, "PROXY UNKNOWN\r\nabc");
```

Dangerous Functions\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=82
Status	New

The dangerous function, strcpy, was found in use at line 26 in h2o/util.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/util.c	h2o/util.c
Line	62	62
Object	strcpy	strcpy

Code Snippet
File Name h2o/util.c
Method static void test_parse_proxy_line(void)

```
....
62.      strcpy(in, "PROXY TCP6 ::1 ::1 56324 443\r\n");
```

Dangerous Functions\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=83
Status	New

The dangerous function, strlen, was found in use at line 144 in h2o/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	145	145
Object	strlen	strlen

Code Snippet
File Name h2o/sds.c
Method sds sdsnew(const char *init) {

```
....
145.      size_t initlen = (init == NULL) ? 0 : strlen(init);
```

Dangerous Functions\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=84
Status	New

The dangerous function, strlen, was found in use at line 174 in h2o/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	175	175
Object	strlen	strlen

Code Snippet
File Name h2o/sds.c
Method void sdsupdatelen(sds s) {

```
....
175.      int reallen = strlen(s);
```

Dangerous Functions\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=85
Status	New

The dangerous function, strlen, was found in use at line 391 in h2o/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	392	392
Object	strlen	strlen

Code Snippet
File Name h2o/sds.c
Method sds sdscat(sds s, const char *t) {

```
....
392.      return sdscatlen(s, t, strlen(t));
```

Dangerous Functions\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=86
Status	New

The dangerous function, strlen, was found in use at line 418 in h2o/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	419	419
Object	strlen	strlen

Code Snippet
File Name h2o/sds.c
Method sds sdscpy(sds s, const char *t) {

```
....
419.      return sdscpylen(s, t, strlen(t));
```

Dangerous Functions\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=87
Status	New

The dangerous function, strlen, was found in use at line 501 in h2o/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	504	504
Object	strlen	strlen

Code Snippet
File Name h2o/sds.c
Method sds sdscatvprintf(sds s, const char *fmt, va_list ap) {

```
....
504.      size_t buflen = strlen(fmt)*2;
```

Dangerous Functions\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=88
Status	New

The dangerous function, strlen, was found in use at line 579 in h2o/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	605	605
Object	strlen	strlen

Code Snippet
File Name h2o/sds.c
Method sds sdscatfmt(sds s, char const *fmt, ...) {

```
....
605.                    l = (next == 's') ? strlen(str) : sdslen(str);
```

Dangerous Functions\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=89
Status	New

The dangerous function, strlen, was found in use at line 22 in h2o/sprintf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sprintf.c	h2o/sprintf.c
Line	33	33
Object	strlen	strlen

Code Snippet
File Name h2o/sprintf.c
Method remove_sign_bits(char *str, int base)

```
....
33.       *t |= EXTENDSIGN(3, strlen(t));
```

Dangerous Functions\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=90
Status	New

The dangerous function, strlen, was found in use at line 558 in h2o/sprintf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sprintf.c	h2o/sprintf.c
Line	923	923
Object	strlen	strlen

Code Snippet
File Name h2o/sprintf.c
Method mrb_str_format(mrb_state *mrb, mrb_int argc, const mrb_value *argv, mrb_value fmt)

```
....
923.            size = strlen(s);
```

Dangerous Functions\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=91
Status	New

The dangerous function, strlen, was found in use at line 558 in h2o/sprintf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sprintf.c	h2o/sprintf.c
Line	955	955
Object	strlen	strlen

Code Snippet

File Name h2o/sprintf.c

Method mrb_str_format(mrb_state *mrb, mrb_int argc, const mrb_value *argv, mrb_value fmt)

```
....
955.            size = strlen(prefix);
```

Dangerous Functions\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=92
Status	New

The dangerous function, strlen, was found in use at line 558 in h2o/sprintf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	h2o/sprintf.c	h2o/sprintf.c
Line	981	981
Object	strlen	strlen

Code Snippet

File Name h2o/sprintf.c

Method mrb_str_format(mrb_state *mrb, mrb_int argc, const mrb_value *argv, mrb_value fmt)

```
....
981.            int plen = (int)strlen(prefix);
```

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=1050071&projectid=50061&pathid=9
Status	New

The size of the buffer used by *ptls_ffx_new in ptls_ffx_context_t, at line 104 of h2o/ffx.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *ptls_ffx_new passes to ptls_ffx_context_t, at line 104 of h2o/ffx.c, to overwrite the target buffer.

	Source	Destination
File	h2o/ffx.c	h2o/ffx.c
Line	109	109
Object	ptls_ffx_context_t	ptls_ffx_context_t

Code Snippet
File Name h2o/ffx.c
Method ptls_cipher_context_t *ptls_ffx_new(ptls_cipher_algorithm_t *algo, int is_enc, int nb_rounds, size_t bit_length, const void *key)

```
....
109.            memset(ctx, 0, sizeof(ptls_ffx_context_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=10
Status	New

The size of the buffer used by mrb_proc_merge_lvar in num, at line 401 of h2o/proc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that mrb_proc_merge_lvar passes to num, at line 401 of h2o/proc.c, to overwrite the target buffer.

	Source	Destination
File	h2o/proc.c	h2o/proc.c

Line	418	418
Object	num	num

Code Snippet
File Name h2o/proc.c
Method mrb_proc_merge_lvar(mrb_state *mrb, mrb_irep *irep, struct REnv *env, int num, const mrb_sym *lv, const mrb_value *stack)

```
....
418.    memmove(destlv, lv, sizeof(mrb_sym) * num);
```

Buffer Overflow boundcpy WrongSizeParam\Path 3:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=11
Status New

The size of the buffer used by mrb_proc_merge_lvar in mrb_sym, at line 401 of h2o/proc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that mrb_proc_merge_lvar passes to mrb_sym, at line 401 of h2o/proc.c, to overwrite the target buffer.

	Source	Destination
File	h2o/proc.c	h2o/proc.c
Line	418	418
Object	mrb_sym	mrb_sym

Code Snippet
File Name h2o/proc.c
Method mrb_proc_merge_lvar(mrb_state *mrb, mrb_irep *irep, struct REnv *env, int num, const mrb_sym *lv, const mrb_value *stack)

```
....
418.    memmove(destlv, lv, sizeof(mrb_sym) * num);
```

Buffer Overflow boundcpy WrongSizeParam\Path 4:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=12
Status New

The size of the buffer used by mrb_proc_merge_lvar in num, at line 401 of h2o/proc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that mrb_proc_merge_lvar passes to num, at line 401 of h2o/proc.c, to overwrite the target buffer.

	Source	Destination
File	h2o/proc.c	h2o/proc.c

Line	420	420
Object	num	num

Code Snippet
File Name h2o/proc.c
Method mrb_proc_merge_lvar(mrb_state *mrb, mrb_irep *irep, struct REnv *env, int num, const mrb_sym *lv, const mrb_value *stack)

```
....
420.      memmove(deststst, stack, sizeof(mrb_value) * num);
```

Buffer Overflow boundcpy WrongSizeParam\Path 5:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=13
Status New

The size of the buffer used by mrb_proc_merge_lvar in mrb_value, at line 401 of h2o/proc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that mrb_proc_merge_lvar passes to mrb_value, at line 401 of h2o/proc.c, to overwrite the target buffer.

	Source	Destination
File	h2o/proc.c	h2o/proc.c
Line	420	420
Object	mrb_value	mrb_value

Code Snippet
File Name h2o/proc.c
Method mrb_proc_merge_lvar(mrb_state *mrb, mrb_irep *irep, struct REnv *env, int num, const mrb_sym *lv, const mrb_value *stack)

```
....
420.      memmove(deststst, stack, sizeof(mrb_value) * num);
```

Buffer Overflow boundcpy WrongSizeParam\Path 6:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=14
Status New

The size of the buffer used by *header_table_add in new_entries, at line 254 of h2o/hpack.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *header_table_add passes to new_entries, at line 254 of h2o/hpack.c, to overwrite the target buffer.

	Source	Destination
File	h2o/hpack.c	h2o/hpack.c

Line	284	284
Object	new_entries	new_entries

Code Snippet
File Name h2o/hpack.c
Method static struct st_h2o_hpack_header_table_entry_t *header_table_add(h2o_hpack_header_table_t *table, size_t size_add,

```
....
284.          memset(new_entries + table->num_entries, 0,
sizeof(*new_entries) * (new_capacity - table->num_entries));
```

Buffer Overflow boundcpy WrongSizeParam\Path 7:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=15
Status New

The size of the buffer used by ffx_encrypt in ctx, at line 138 of h2o/ffx.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ffx_encrypt passes to ctx, at line 138 of h2o/ffx.c, to overwrite the target buffer.

	Source	Destination
File	h2o/ffx.c	h2o/ffx.c
Line	154	154
Object	ctx	ctx

Code Snippet
File Name h2o/ffx.c
Method static void ffx_encrypt(ptls_cipher_context_t *_ctx, void *output, const void *input, size_t len)

```
....
154.      memcpy(left, input, ctx->nb_left);
```

Buffer Overflow boundcpy WrongSizeParam\Path 8:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=16
Status New

The size of the buffer used by ffx_encrypt in ctx, at line 138 of h2o/ffx.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ffx_encrypt passes to ctx, at line 138 of h2o/ffx.c, to overwrite the target buffer.

	Source	Destination
File	h2o/ffx.c	h2o/ffx.c

Line	155	155
Object	ctx	ctx

Code Snippet
File Name h2o/ffx.c
Method static void ffx_encrypt(ptls_cipher_context_t *_ctx, void *output, const void *input, size_t len)

```
....
155.      memcpy(right, ((uint8_t *)input) + ctx->nb_left, ctx->nb_right);
```

Buffer Overflow boundcpy WrongSizeParam\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=17
Status	New

The size of the buffer used by ffx_encrypt in ctx, at line 138 of h2o/ffx.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ffx_encrypt passes to ctx, at line 138 of h2o/ffx.c, to overwrite the target buffer.

	Source	Destination
File	h2o/ffx.c	h2o/ffx.c
Line	190	190
Object	ctx	ctx

Code Snippet
File Name h2o/ffx.c
Method static void ffx_encrypt(ptls_cipher_context_t *_ctx, void *output, const void *input, size_t len)

```
....
190.      memcpy(output, left, ctx->nb_left);
```

Buffer Overflow boundcpy WrongSizeParam\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=18
Status	New

The size of the buffer used by ffx_encrypt in ctx, at line 138 of h2o/ffx.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ffx_encrypt passes to ctx, at line 138 of h2o/ffx.c, to overwrite the target buffer.

	Source	Destination
File	h2o/ffx.c	h2o/ffx.c

Line	195	195
Object	ctx	ctx

Code Snippet
File Name h2o/ffx.c
Method static void ffx_encrypt(ptls_cipher_context_t *_ctx, void *output, const void *input, size_t len)

```
....
195.      memcpy(((uint8_t *)output) + ctx->nb_left, right, ctx->nb_right);
```

Buffer Overflow boundcpy WrongSizeParam\Path 11:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=19
Status New

The size of the buffer used by *encode_content_length in l, at line 445 of h2o/hpack.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *encode_content_length passes to l, at line 445 of h2o/hpack.c, to overwrite the target buffer.

	Source	Destination
File	h2o/hpack.c	h2o/hpack.c
Line	457	457
Object	l	l

Code Snippet
File Name h2o/hpack.c
Method static uint8_t *encode_content_length(uint8_t *dst, size_t value)

```
....
457.      memcpy(dst, p, l);
```

Buffer Overflow boundcpy WrongSizeParam\Path 12:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=20
Status New

The size of the buffer used by *do_encode_header in name, at line 766 of h2o/hpack.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *do_encode_header passes to name, at line 766 of h2o/hpack.c, to overwrite the target buffer.

	Source	Destination
File	h2o/hpack.c	h2o/hpack.c

Line	836	836
Object	name	name

Code Snippet
File Name h2o/hpack.c
Method static uint8_t *do_encode_header(h2o_hpack_header_table_t *header_table, uint8_t *dst, const h2o_iovec_t *name,

```
....
836.                    memcpy(entry->name->base, name->base, name->len);
```

Buffer Overflow boundcpy WrongSizeParam\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=21
Status	New

The size of the buffer used by *do_encode_header in value, at line 766 of h2o/hpack.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *do_encode_header passes to value, at line 766 of h2o/hpack.c, to overwrite the target buffer.

	Source	Destination
File	h2o/hpack.c	h2o/hpack.c
Line	840	840
Object	value	value

Code Snippet
File Name h2o/hpack.c
Method static uint8_t *do_encode_header(h2o_hpack_header_table_t *header_table, uint8_t *dst, const h2o_iovec_t *name,

```
....
840.                memcpy(entry->value->base, value->base, value->len);
```

Buffer Overflow boundcpy WrongSizeParam\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=22
Status	New

The size of the buffer used by sdscatlen in len, at line 376 of h2o/sds.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sdscatlen passes to len, at line 376 of h2o/sds.c, to overwrite the target buffer.

	Source	Destination
File	h2o/sds.c	h2o/sds.c

Line	381	381
Object	len	len

Code Snippet
File Name h2o/sds.c
Method sds sdscatlen(sds s, const void *t, size_t len) {

```
....
381.      memcpy(s+curlen, t, len);
```

Buffer Overflow boundcpy WrongSizeParam\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=23
Status	New

The size of the buffer used by sdscpylen in len, at line 405 of h2o/sds.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sdscpylen passes to len, at line 405 of h2o/sds.c, to overwrite the target buffer.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	410	410
Object	len	len

Code Snippet
File Name h2o/sds.c
Method sds sdscpylen(sds s, const char *t, size_t len) {

```
....
410.      memcpy(s, t, len);
```

Buffer Overflow boundcpy WrongSizeParam\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=24
Status	New

The size of the buffer used by sdscatfmt in l, at line 579 of h2o/sds.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sdscatfmt passes to l, at line 579 of h2o/sds.c, to overwrite the target buffer.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	609	609

Object	l	l

Code Snippet
File Name h2o/sds.c
Method sds sdscatfmt(sds s, char const *fmt, ...) {

```
....
609.                    memcpy(s+i,str,l);
```

Buffer Overflow boundcpy WrongSizeParam\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=25
Status	New

The size of the buffer used by sdscatfmt in l, at line 579 of h2o/sds.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sdscatfmt passes to l, at line 579 of h2o/sds.c, to overwrite the target buffer.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	625	625
Object	l	l

Code Snippet
File Name h2o/sds.c
Method sds sdscatfmt(sds s, char const *fmt, ...) {

```
....
625.                        memcpy(s+i,buf,l);
```

Buffer Overflow boundcpy WrongSizeParam\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=26
Status	New

The size of the buffer used by sdscatfmt in l, at line 579 of h2o/sds.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sdscatfmt passes to l, at line 579 of h2o/sds.c, to overwrite the target buffer.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	642	642
Object	l	l

Code Snippet
File Name h2o/sds.c
Method sds sdscatfmt(sds s, char const *fmt, ...) {

```
....
642.                        memcpy(s+i,buf,l);
```

Buffer Overflow boundcpy WrongSizeParam\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=27
Status	New

The size of the buffer used by fixup_frame_headers in left, at line 920 of h2o/hpack.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that fixup_frame_headers passes to left, at line 920 of h2o/hpack.c, to overwrite the target buffer.

	Source	Destination
File	h2o/hpack.c	h2o/hpack.c
Line	938	938
Object	left	left

Code Snippet
File Name h2o/hpack.c
Method static void fixup_frame_headers(h2o_buffer_t **buf, size_t start_at, uint8_t type, uint32_t stream_id, size_t max_frame_size,

```
....
938.          memmove((*buf)->bytes + off + H2O_HTTP2_FRAME_HEADER_SIZE,
(*buf)->bytes + off, left);
```

Buffer Overflow boundcpy WrongSizeParam\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=28
Status	New

The size of the buffer used by sdscmp in minlen, at line 765 of h2o/sds.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sdscmp passes to minlen, at line 765 of h2o/sds.c, to overwrite the target buffer.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	772	772
Object	minlen	minlen

Code Snippet

File Name	h2o/sds.c
Method	int sdscmp(const sds s1, const sds s2) {

```
....
772.        cmp = memcmp(s1,s2,minlen);
```

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=1050071&projectid=50061&pathid=33
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 579 of h2o/sds.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	611	611
Object	AssignExpr	AssignExpr

Code Snippet	
File Name	h2o/sds.c
Method	sds sdscatfmt(sds s, char const *fmt, ...) {

```
....
611.                    i += l;
```

Integer Overflow\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=34
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 579 of h2o/sds.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	h2o/sds.c	h2o/sds.c

Line	627	627
Object	AssignExpr	AssignExpr

Code Snippet
File Name h2o/sds.c
Method sds sdscatfmt(sds s, char const *fmt, ...) {

```
....
627.                        i += l;
```

Integer Overflow\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=35
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 579 of h2o/sds.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	644	644
Object	AssignExpr	AssignExpr

Code Snippet
File Name h2o/sds.c
Method sds sdscatfmt(sds s, char const *fmt, ...) {

```
....
644.                        i += l;
```

Integer Overflow\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=36
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 712 of h2o/sds.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	717	717
Object	AssignExpr	AssignExpr

Code Snippet
File Name h2o/sds.c
Method void sdsrange(sds s, int start, int end) {

```
....
717.          start = len+start;
```

Integer Overflow\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=37
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 712 of h2o/sds.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	721	721
Object	AssignExpr	AssignExpr

Code Snippet
File Name h2o/sds.c
Method void sdsrange(sds s, int start, int end) {

```
....
721.          end = len+end;
```

Integer Overflow\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=38
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 712 of h2o/sds.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	729	729
Object	AssignExpr	AssignExpr

Code Snippet
File Name h2o/sds.c
Method void sdsrange(sds s, int start, int end) {

```
....
729.              end = len-1;
```

Integer Overflow\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=39
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 558 of h2o/sprintf.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	h2o/sprintf.c	h2o/sprintf.c
Line	958	958
Object	AssignExpr	AssignExpr

Code Snippet
File Name h2o/sprintf.c
Method mrb_str_format(mrb_state *mrb, mrb_int argc, const mrb_value *argv, mrb_value fmt)

```
....
958.            width -= (mrb_int)size;
```

Use of Zero Initialized Pointer

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

Categories

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

Description

Use of Zero Initialized Pointer\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=179
Status	New

The variable declared in supp at h2o/fusion.c in line 289 is not initialized when it is used by supp at h2o/fusion.c in line 289.

	Source	Destination
File	h2o/fusion.c	h2o/fusion.c
Line	293	335

Object	supp	supp

Code Snippet
File Name h2o/fusion.c
Method static void gcm_test_vectors(void)

```
....
293.      ptls_aead_supplementary_encryption_t *supp = NULL;
....
335.          supp = malloc(sizeof(*supp));
```

Use of Zero Initialized Pointer\Path 2:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=180
Status New

The variable declared in supp at h2o/fusion.c in line 289 is not initialized when it is used by supp at h2o/fusion.c in line 289.

	Source	Destination
File	h2o/fusion.c	h2o/fusion.c
Line	293	340
Object	supp	supp

Code Snippet
File Name h2o/fusion.c
Method static void gcm_test_vectors(void)

```
....
293.      ptls_aead_supplementary_encryption_t *supp = NULL;
....
340.      ptls_cipher_free(supp->ctx);
```

Use of Zero Initialized Pointer\Path 3:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=181
Status New

The variable declared in vector at h2o/sds.c in line 931 is not initialized when it is used by vector at h2o/sds.c in line 931.

	Source	Destination
File	h2o/sds.c	h2o/sds.c

Line	934	1021
Object	vector	vector

Code Snippet
File Name h2o/sds.c
Method sds *sdssplitargs(const char *line, int *argc) {

```
....
934.      char **vector = NULL;
....
1021.             vector = s_realloc(vector,((*argc)+1)*sizeof(char*));
```

Use of Zero Initialized Pointer\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=182
Status	New

The variable declared in current at h2o/sds.c in line 931 is not initialized when it is used by vector at h2o/sds.c in line 931.

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	1024	1022
Object	current	vector

Code Snippet
File Name h2o/sds.c
Method sds *sdssplitargs(const char *line, int *argc) {

```
....
1024.             current = NULL;
....
1022.             vector[*argc] = current;
```

Use of Zero Initialized Pointer\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=183
Status	New

The variable declared in vector at h2o/sds.c in line 931 is not initialized when it is used by vector at h2o/sds.c in line 931.

	Source	Destination

File	h2o/sds.c	h2o/sds.c
Line	934	1034
Object	vector	vector

Code Snippet
File Name h2o/sds.c
Method sds *sdssplitargs(const char *line, int *argc) {

```
....
934.      char **vector = NULL;
....
1034.          sdsfree(vector[*argc]);
```

Use of Zero Initialized Pointer\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=184
Status	New

The variable declared in enc_ctx at h2o/ffx.c in line 32 is not initialized when it is used by enc_ctx at h2o/ffx.c in line 32.

	Source	Destination
File	h2o/ffx.c	h2o/ffx.c
Line	37	62
Object	enc_ctx	enc_ctx

Code Snippet
File Name h2o/ffx.c
Method int ptls_ffx_setup_crypto(ptls_cipher_context_t *_ctx, ptls_cipher_algorithm_t *algo, int is_enc, int nb_rounds, size_t bit_length,

```
....
37.      ptls_cipher_context_t *enc_ctx = NULL;
....
62.          ctx->enc_ctx = enc_ctx;
```

Wrong Size t Allocation

Query Path:
CPP\Cx\CPP Integer Overflow\Wrong Size t Allocation Version:0

Description

Wrong Size t Allocation\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=29
Status	New

The function capacity in h2o/yc.c at line 129 assigns an incorrectly calculated size to a buffer, resulting in a mismatch between the value being written and the size of the buffer it is being written into.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	140	140
Object	capacity	capacity

Code Snippet
File Name h2o/yc.c
Method static size_t read_data(const char* filename, char** pdata) {

```
....
140.      *pdata = (char*)malloc(capacity);
```

Wrong Size t Allocation\Path 2:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=30
Status New

The function capacity in h2o/yc.c at line 129 assigns an incorrectly calculated size to a buffer, resulting in a mismatch between the value being written and the size of the buffer it is being written into.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	144	144
Object	capacity	capacity

Code Snippet
File Name h2o/yc.c
Method static size_t read_data(const char* filename, char** pdata) {

```
....
144.              char* new_data = (char*)realloc(*pdata, capacity * 2);
```

Wrong Size t Allocation\Path 3:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=31
Status New

The function ghash_cnt in h2o/fusion.c at line 87 assigns an incorrectly calculated size to a buffer, resulting in a mismatch between the value being written and the size of the buffer it is being written into.

	Source	Destination
File	h2o/fusion.c	h2o/fusion.c
Line	92	92
Object	ghash_cnt	ghash_cnt

Code Snippet
File Name h2o/fusion.c
Method static void test_gfmul(void)

```
....
92.      ctx = malloc(calc_aesgcm_context_size(&ghash_cnt,
ptls_fusion_can_aesni256));
```

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=1050071&projectid=50061&pathid=185
Status New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	153	153
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name h2o/yc.c
Method static size_t read_data(const char* filename, char** pdata) {

```
....
153.          ssize_t n = read(fd, *pdata + data_len, 1 << 20);
```

Improper Resource Access Authorization\Path 2:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=186

Status	New

	Source	Destination
File	h2o/test.c	h2o/test.c
Line	36	36
Object	fprintf	fprintf

Code Snippet
File Name h2o/test.c
Method int main(int argc, char **argv)

```
....
36.          fprintf(stderr, "golombset_encode failed\n");
```

Improper Resource Access Authorization\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=187
Status	New

	Source	Destination
File	h2o/test.c	h2o/test.c
Line	44	44
Object	fprintf	fprintf

Code Snippet
File Name h2o/test.c
Method int main(int argc, char **argv)

```
....
44.          fprintf(stderr, "golombset_decode failed\n");
```

Improper Resource Access Authorization\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=188
Status	New

	Source	Destination
File	h2o/test.c	h2o/test.c
Line	49	49
Object	fprintf	fprintf

Code Snippet
File Name h2o/test.c
Method int main(int argc, char **argv)

```
....
49.          fprintf(stderr, "unexpected number of outputs\n");
```

Improper Resource Access Authorization\Path 5:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=189
Status New

	Source	Destination
File	h2o/test.c	h2o/test.c
Line	53	53
Object	fprintf	fprintf

Code Snippet
File Name h2o/test.c
Method int main(int argc, char **argv)

```
....
53.          fprintf(stderr, "output mismatch\n");
```

Improper Resource Access Authorization\Path 6:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=190
Status New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	1006	1006
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int main(int argc, char** argv) {

```
....
1006.                  fprintf(stderr, "Invalid TCP port.\n");
```

Improper Resource Access Authorization\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=191
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	1014	1014
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int main(int argc, char** argv) {

```
....
1014.                   fprintf(stderr, "Invalid compression
thoreshold.\n");
```

Improper Resource Access Authorization\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=192
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	1050	1050
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int main(int argc, char** argv) {

```
....
1050.       CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=193
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	1050	1050
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int main(int argc, char** argv) {

```
....
1050.       CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 10:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=194
Status New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	1053	1053
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int main(int argc, char** argv) {

```
....
1053.           CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 11:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=195
Status New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	1053	1053
Object	fprintf	fprintf

Code Snippet

File Name	h2o/yc.c
Method	int main(int argc, char** argv) {

```
....
1053.           CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=196
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	1058	1058
Object	fprintf	fprintf

Code Snippet

File Name	h2o/yc.c
Method	int main(int argc, char** argv) {

```
....
1058.           CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=197
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	1058	1058
Object	fprintf	fprintf

Code Snippet

File Name	h2o/yc.c
Method	int main(int argc, char** argv) {

```
....
1058.           CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 14:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=198
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	1094	1094
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int main(int argc, char** argv) {

```
....
1094.       fprintf(stderr, "No such command: %s\n", cmd);
```

Improper Resource Access Authorization\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=199
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	108	108
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method static void print_response(const yrmcds_response* r) {

```
....
108.            fprintf(stderr, "  key:      %.*s (%lu bytes)\n",
```

Improper Resource Access Authorization\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=200
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c

Line	111	111
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method static void print_response(const yrmcds_response* r) {

```
....
111.          fprintf(stderr, "  data:     %.*s (%lu bytes)\n",
```

Improper Resource Access Authorization\Path 17:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=201
Status New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	190	190
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_noop(int argc, char** argv, yrmcds* s) {

```
....
190.      CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 18:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=202
Status New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	190	190
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_noop(int argc, char** argv, yrmcds* s) {

```
....
190.        CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=203
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	192	192
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_noop(int argc, char** argv, yrmcds* s) {

```
....
192.            fprintf(stderr, "request serial = %u\n", serial);
```

Improper Resource Access Authorization\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=204
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	194	194
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_noop(int argc, char** argv, yrmcds* s) {

```
....
194.        CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 21:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=205
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	194	194
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_noop(int argc, char** argv, yrmcds* s) {

```
....
194.        CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=206
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	197	197
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_noop(int argc, char** argv, yrmcds* s) {

```
....
197.        CHECK_RESPONSE(r);
```

Improper Resource Access Authorization\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=207
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	204	204

Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_get(int argc, char** argv, yrmcds* s) {

```
....
204.          fprintf(stderr, "Wrong number of arguments.\n");
```

Improper Resource Access Authorization\Path 24:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=208
Status New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	210	210
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_get(int argc, char** argv, yrmcds* s) {

```
....
210.      CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 25:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=209
Status New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	210	210
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_get(int argc, char** argv, yrmcds* s) {

```
....
210.        CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=210
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	213	213
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_get(int argc, char** argv, yrmcds* s) {

```
....
213.            CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=211
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	213	213
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_get(int argc, char** argv, yrmcds* s) {

```
....
213.            CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 28:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=212
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	216	216
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_get(int argc, char** argv, yrmcds* s) {

```
....
216.            fprintf(stderr, "request serial = %u\n", serial);
```

Improper Resource Access Authorization\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=213
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	219	219
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_get(int argc, char** argv, yrmcds* s) {

```
....
219.            CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=214
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	219	219

Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_get(int argc, char** argv, yrmcds* s) {

```
....
219.          CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 31:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=215
Status New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	222	222
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_get(int argc, char** argv, yrmcds* s) {

```
....
222.          CHECK_RESPONSE(r);
```

Improper Resource Access Authorization\Path 32:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=216
Status New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	232	232
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_getk(int argc, char** argv, yrmcds* s) {

```
....
232.          fprintf(stderr, "Wrong number of arguments.\n");
```

Improper Resource Access Authorization\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=217
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	238	238
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_getk(int argc, char** argv, yrmcds* s) {

```
....
238.      CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=218
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	238	238
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_getk(int argc, char** argv, yrmcds* s) {

```
....
238.      CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 35:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=219
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	241	241
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_getk(int argc, char** argv, yrmcds* s) {

```
....
241.            CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=220
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	241	241
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_getk(int argc, char** argv, yrmcds* s) {

```
....
241.            CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=221
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	244	244

Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_getk(int argc, char** argv, yrmcds* s) {

```
....
244.          fprintf(stderr, "request serial = %u\n", serial);
```

Improper Resource Access Authorization\Path 38:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=222
Status New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	247	247
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_getk(int argc, char** argv, yrmcds* s) {

```
....
247.          CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 39:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=223
Status New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	247	247
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_getk(int argc, char** argv, yrmcds* s) {

```
....
247.            CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=224
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	250	250
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_getk(int argc, char** argv, yrmcds* s) {

```
....
250.            CHECK_RESPONSE(r);
```

Improper Resource Access Authorization\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=225
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	260	260
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_gat(int argc, char** argv, yrmcds* s) {

```
....
260.            fprintf(stderr, "Wrong number of arguments.\n");
```

Improper Resource Access Authorization\Path 42:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=226
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	269	269
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_gat(int argc, char** argv, yrmcds* s) {

```
....
269.        CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=227
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	269	269
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_gat(int argc, char** argv, yrmcds* s) {

```
....
269.        CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=228
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	272	272

Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_gat(int argc, char** argv, yrmcds* s) {

```
....
272.            CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 45:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=229
Status New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	272	272
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_gat(int argc, char** argv, yrmcds* s) {

```
....
272.            CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 46:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=230
Status New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	275	275
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_gat(int argc, char** argv, yrmcds* s) {

```
....
275.          fprintf(stderr, "request serial = %u\n", serial);
```

Improper Resource Access Authorization\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=231
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	278	278
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_gat(int argc, char** argv, yrmcds* s) {

```
....
278.          CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 48:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=232
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	278	278
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_gat(int argc, char** argv, yrmcds* s) {

```
....
278.          CHECK_ERROR(e);
```

Improper Resource Access Authorization\Path 49:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=233
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	281	281
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_gat(int argc, char** argv, yrmcds* s) {

```
....
281.          CHECK_RESPONSE(r);
```

Improper Resource Access Authorization\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=234
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	291	291
Object	fprintf	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_gatk(int argc, char** argv, yrmcds* s) {

```
....
291.          fprintf(stderr, "Wrong number of arguments.\n");
```

Exposure of System Data to Unauthorized Control Sphere

Query Path:
CPP\Cx\CPP Low Visibility\Exposure of System Data to Unauthorized Control Sphere Version:1

Categories

FISMA 2014: Configuration Management
NIST SP 800-53: AC-3 Access Enforcement (P1)

Description

Exposure of System Data to Unauthorized Control Sphere\Path 1:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=420
Status	New

The system data read by main in the file h2o/yc.c at line 989 is potentially exposed by main found in h2o/yc.c at line 989.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	1050	1050
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int main(int argc, char** argv) {

```
....
1050.       CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=421
Status	New

The system data read by main in the file h2o/yc.c at line 989 is potentially exposed by main found in h2o/yc.c at line 989.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	1053	1053
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int main(int argc, char** argv) {

```
....
1053.           CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=422
Status	New

The system data read by main in the file h2o/yc.c at line 989 is potentially exposed by main found in h2o/yc.c at line 989.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	1058	1058
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int main(int argc, char** argv) {

```
....
1058.           CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 4:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=423
Status New

The system data read by cmd_noop in the file h2o/yc.c at line 186 is potentially exposed by cmd_noop found in h2o/yc.c at line 186.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	190	190
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_noop(int argc, char** argv, yrmcds* s) {

```
....
190.       CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 5:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=424
Status New

The system data read by cmd_noop in the file h2o/yc.c at line 186 is potentially exposed by cmd_noop found in h2o/yc.c at line 186.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	194	194
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_noop(int argc, char** argv, yrmcds* s) {

```
....
194.        CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=425
Status	New

The system data read by cmd_get in the file h2o/yc.c at line 202 is potentially exposed by cmd_get found in h2o/yc.c at line 202.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	210	210
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_get(int argc, char** argv, yrmcds* s) {

```
....
210.        CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=426
Status	New

The system data read by cmd_get in the file h2o/yc.c at line 202 is potentially exposed by cmd_get found in h2o/yc.c at line 202.

	Source	Destination
File	h2o/yc.c	h2o/yc.c

Line	213	213
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_get(int argc, char** argv, yrmcds* s) {

```
....
213.            CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=427
Status	New

The system data read by cmd_get in the file h2o/yc.c at line 202 is potentially exposed by cmd_get found in h2o/yc.c at line 202.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	219	219
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_get(int argc, char** argv, yrmcds* s) {

```
....
219.            CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=428
Status	New

The system data read by cmd_getk in the file h2o/yc.c at line 230 is potentially exposed by cmd_getk found in h2o/yc.c at line 230.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	238	238
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_getk(int argc, char** argv, yrmcds* s) {

```
....
238.        CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=429
Status	New

The system data read by cmd_getk in the file h2o/yc.c at line 230 is potentially exposed by cmd_getk found in h2o/yc.c at line 230.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	241	241
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_getk(int argc, char** argv, yrmcds* s) {

```
....
241.            CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=430
Status	New

The system data read by cmd_getk in the file h2o/yc.c at line 230 is potentially exposed by cmd_getk found in h2o/yc.c at line 230.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	247	247
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c

Method	int cmd_getk(int argc, char** argv, yrmcds* s) {

```
....
247.            CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=431
Status	New

The system data read by cmd_gat in the file h2o/yc.c at line 258 is potentially exposed by cmd_gat found in h2o/yc.c at line 258.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	269	269
Object	errno	fprintf

Code Snippet	
File Name	h2o/yc.c
Method	int cmd_gat(int argc, char** argv, yrmcds* s) {

```
....
269.        CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=432
Status	New

The system data read by cmd_gat in the file h2o/yc.c at line 258 is potentially exposed by cmd_gat found in h2o/yc.c at line 258.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	272	272
Object	errno	fprintf

Code Snippet	
File Name	h2o/yc.c
Method	int cmd_gat(int argc, char** argv, yrmcds* s) {

```
....
272.            CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=433
Status	New

The system data read by cmd_gat in the file h2o/yc.c at line 258 is potentially exposed by cmd_gat found in h2o/yc.c at line 258.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	278	278
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_gat(int argc, char** argv, yrmcds* s) {

```
....
278.            CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=434
Status	New

The system data read by cmd_gatk in the file h2o/yc.c at line 289 is potentially exposed by cmd_gatk found in h2o/yc.c at line 289.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	300	300
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_gatk(int argc, char** argv, yrmcds* s) {

```
....
300.        CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=435
Status	New

The system data read by cmd_gatk in the file h2o/yc.c at line 289 is potentially exposed by cmd_gatk found in h2o/yc.c at line 289.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	303	303
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_gatk(int argc, char** argv, yrmcds* s) {

```
....
303.            CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=436
Status	New

The system data read by cmd_gatk in the file h2o/yc.c at line 289 is potentially exposed by cmd_gatk found in h2o/yc.c at line 289.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	309	309
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_gatk(int argc, char** argv, yrmcds* s) {

```
....
309.          CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=437
Status	New

The system data read by cmd_lag in the file h2o/yc.c at line 320 is potentially exposed by cmd_lag found in h2o/yc.c at line 320.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	328	328
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_lag(int argc, char** argv, yrmcds* s) {

```
....
328.      CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=438
Status	New

The system data read by cmd_lag in the file h2o/yc.c at line 320 is potentially exposed by cmd_lag found in h2o/yc.c at line 320.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	331	331
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_lag(int argc, char** argv, yrmcds* s) {

```
....
331.          CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=439
Status	New

The system data read by cmd_lag in the file h2o/yc.c at line 320 is potentially exposed by cmd_lag found in h2o/yc.c at line 320.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	337	337
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_lag(int argc, char** argv, yrmcds* s) {

```
....
337.          CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=440
Status	New

The system data read by cmd_lagk in the file h2o/yc.c at line 350 is potentially exposed by cmd_lagk found in h2o/yc.c at line 350.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	358	358
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_lagk(int argc, char** argv, yrmcds* s) {

```
....
358.        CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=441
Status	New

The system data read by cmd_lagk in the file h2o/yc.c at line 350 is potentially exposed by cmd_lagk found in h2o/yc.c at line 350.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	361	361
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_lagk(int argc, char** argv, yrmcds* s) {

```
....
361.            CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=442
Status	New

The system data read by cmd_lagk in the file h2o/yc.c at line 350 is potentially exposed by cmd_lagk found in h2o/yc.c at line 350.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	367	367
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_lagk(int argc, char** argv, yrmcds* s) {

```
....
367.          CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=443
Status	New

The system data read by cmd_touch in the file h2o/yc.c at line 380 is potentially exposed by cmd_touch found in h2o/yc.c at line 380.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	391	391
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_touch(int argc, char** argv, yrmcds* s) {

```
....
391.      CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=444
Status	New

The system data read by cmd_touch in the file h2o/yc.c at line 380 is potentially exposed by cmd_touch found in h2o/yc.c at line 380.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	394	394
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_touch(int argc, char** argv, yrmcds* s) {

```
....
394.            CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=445
Status	New

The system data read by cmd_touch in the file h2o/yc.c at line 380 is potentially exposed by cmd_touch found in h2o/yc.c at line 380.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	400	400
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_touch(int argc, char** argv, yrmcds* s) {

```
....
400.            CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=446
Status	New

The system data read by cmd_set in the file h2o/yc.c at line 410 is potentially exposed by cmd_set found in h2o/yc.c at line 410.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	438	438
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_set(int argc, char** argv, yrmcds* s) {

```
....
438.        CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=447
Status	New

The system data read by cmd_set in the file h2o/yc.c at line 410 is potentially exposed by cmd_set found in h2o/yc.c at line 410.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	441	441
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_set(int argc, char** argv, yrmcds* s) {

```
....
441.            CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=448
Status	New

The system data read by cmd_set in the file h2o/yc.c at line 410 is potentially exposed by cmd_set found in h2o/yc.c at line 410.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	447	447
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_set(int argc, char** argv, yrmcds* s) {

```
....
447.          CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=449
Status	New

The system data read by cmd_replace in the file h2o/yc.c at line 457 is potentially exposed by cmd_replace found in h2o/yc.c at line 457.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	485	485
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_replace(int argc, char** argv, yrmcds* s) {

```
....
485.      CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=450
Status	New

The system data read by cmd_replace in the file h2o/yc.c at line 457 is potentially exposed by cmd_replace found in h2o/yc.c at line 457.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	488	488
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_replace(int argc, char** argv, yrmcds* s) {

```
....
488.          CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=451
Status	New

The system data read by cmd_replace in the file h2o/yc.c at line 457 is potentially exposed by cmd_replace found in h2o/yc.c at line 457.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	494	494
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_replace(int argc, char** argv, yrmcds* s) {

```
....
494.          CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=452
Status	New

The system data read by cmd_add in the file h2o/yc.c at line 504 is potentially exposed by cmd_add found in h2o/yc.c at line 504.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	532	532
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_add(int argc, char** argv, yrmcds* s) {

```
....
532.        CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=453
Status	New

The system data read by cmd_add in the file h2o/yc.c at line 504 is potentially exposed by cmd_add found in h2o/yc.c at line 504.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	535	535
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_add(int argc, char** argv, yrmcds* s) {

```
....
535.            CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=454
Status	New

The system data read by cmd_add in the file h2o/yc.c at line 504 is potentially exposed by cmd_add found in h2o/yc.c at line 504.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	541	541
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_add(int argc, char** argv, yrmcds* s) {

```
....
541.          CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=455
Status	New

The system data read by cmd_rau in the file h2o/yc.c at line 551 is potentially exposed by cmd_rau found in h2o/yc.c at line 551.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	576	576
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_rau(int argc, char** argv, yrmcds* s) {

```
....
576.      CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=456
Status	New

The system data read by cmd_rau in the file h2o/yc.c at line 551 is potentially exposed by cmd_rau found in h2o/yc.c at line 551.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	579	579
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_rau(int argc, char** argv, yrmcds* s) {

```
....
579.            CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=457
Status	New

The system data read by cmd_rau in the file h2o/yc.c at line 551 is potentially exposed by cmd_rau found in h2o/yc.c at line 551.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	585	585
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_rau(int argc, char** argv, yrmcds* s) {

```
....
585.            CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=458
Status	New

The system data read by cmd_incr in the file h2o/yc.c at line 595 is potentially exposed by cmd_incr found in h2o/yc.c at line 595.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	621	621
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_incr(int argc, char** argv, yrmcds* s) {

```
....
621.        CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=459
Status	New

The system data read by cmd_incr in the file h2o/yc.c at line 595 is potentially exposed by cmd_incr found in h2o/yc.c at line 595.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	624	624
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_incr(int argc, char** argv, yrmcds* s) {

```
....
624.            CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=460
Status	New

The system data read by cmd_incr in the file h2o/yc.c at line 595 is potentially exposed by cmd_incr found in h2o/yc.c at line 595.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	630	630
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_incr(int argc, char** argv, yrmcds* s) {

```
....
630.          CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=461
Status	New

The system data read by cmd_decr in the file h2o/yc.c at line 641 is potentially exposed by cmd_decr found in h2o/yc.c at line 641.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	667	667
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_decr(int argc, char** argv, yrmcds* s) {

```
....
667.      CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=462
Status	New

The system data read by cmd_decr in the file h2o/yc.c at line 641 is potentially exposed by cmd_decr found in h2o/yc.c at line 641.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	670	670
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_decr(int argc, char** argv, yrmcds* s) {

```
....
670.            CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=463
Status	New

The system data read by cmd_decr in the file h2o/yc.c at line 641 is potentially exposed by cmd_decr found in h2o/yc.c at line 641.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	676	676
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_decr(int argc, char** argv, yrmcds* s) {

```
....
676.            CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 45:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=464
Status	New

The system data read by cmd_append in the file h2o/yc.c at line 687 is potentially exposed by cmd_append found in h2o/yc.c at line 687.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	705	705
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_append(int argc, char** argv, yrmcds* s) {

```
....
705.        CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=465
Status	New

The system data read by cmd_append in the file h2o/yc.c at line 687 is potentially exposed by cmd_append found in h2o/yc.c at line 687.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	708	708
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_append(int argc, char** argv, yrmcds* s) {

```
....
708.            CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=466
Status	New

The system data read by cmd_append in the file h2o/yc.c at line 687 is potentially exposed by cmd_append found in h2o/yc.c at line 687.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	714	714
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_append(int argc, char** argv, yrmcds* s) {

```
....
714.          CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 48:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=467
Status	New

The system data read by cmd_prepend in the file h2o/yc.c at line 724 is potentially exposed by cmd_prepend found in h2o/yc.c at line 724.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	742	742
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_prepend(int argc, char** argv, yrmcds* s) {

```
....
742.      CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 49:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=468
Status	New

The system data read by cmd_prepend in the file h2o/yc.c at line 724 is potentially exposed by cmd_prepend found in h2o/yc.c at line 724.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	745	745
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_prepend(int argc, char** argv, yrmcds* s) {

```
....
745.            CHECK_ERROR(e);
```

Exposure of System Data to Unauthorized Control Sphere\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=469
Status	New

The system data read by cmd_prepend in the file h2o/yc.c at line 724 is potentially exposed by cmd_prepend found in h2o/yc.c at line 724.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	751	751
Object	errno	fprintf

Code Snippet
File Name h2o/yc.c
Method int cmd_prepend(int argc, char** argv, yrmcds* s) {

```
....
751.            CHECK_ERROR(e);
```

Unchecked Array Index

Query Path:
CPP\Cx\CPP Low Visibility\Unchecked Array Index Version:1

Categories

NIST SP 800-53: SI-10 Information Input Validation (P1)

Description

Unchecked Array Index\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=40
Status	New

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	350	350
Object	len	len

Code Snippet

File Name	h2o/sds.c
Method	void sdsIncrLen(sds s, int incr) {

```
....
350.        s[len] = '\0';
```

Unchecked Array Index\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=41
Status	New

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	663	663
Object	i	i

Code Snippet	
File Name	h2o/sds.c
Method	sds sdscatfmt(sds s, char const *fmt, ...) {

```
....
663.        s[i] = '\0';
```

Unchecked Array Index\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=42
Status	New

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	691	691
Object	len	len

Code Snippet	
File Name	h2o/sds.c
Method	sds sdstrim(sds s, const char *cset) {

```
....
691.        s[len] = '\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=1050071&projectid=50061&pathid=3
Status	New

The *encode_status method calls the sprintf function, at line 414 of h2o/hpack.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	h2o/hpack.c	h2o/hpack.c
Line	437	437
Object	sprintf	sprintf

Code Snippet
File Name h2o/hpack.c
Method static uint8_t *encode_status(uint8_t *dst, int status)

```
....
437.          sprintf((char *)dst, "%d", status);
```

Unchecked Return Value\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=4
Status	New

The read_data method calls the Pointer function, at line 129 of h2o/yc.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	h2o/yc.c	h2o/yc.c
Line	140	140
Object	Pointer	Pointer

Code Snippet
File Name h2o/yc.c
Method static size_t read_data(const char* filename, char** pdata) {

```
....
140.      *pdata = (char*)malloc(capacity);
```

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=1050071&projectid=50061&pathid=5
Status	New

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	1021	1021
Object	sizeof	sizeof

Code Snippet
File Name h2o/sds.c
Method sds *sdssplitargs(const char *line, int *argc) {

```
....
1021.              vector = s_realloc(vector,((*argc)+1)*sizeof(char*));
```

Use of Sizeof On a Pointer Type\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=6
Status	New

	Source	Destination
File	h2o/sds.c	h2o/sds.c
Line	1027	1027
Object	sizeof	sizeof

Code Snippet
File Name h2o/sds.c
Method sds *sdssplitargs(const char *line, int *argc) {

```
....
1027.              if (vector == NULL) vector = s_malloc(sizeof(void*));
```

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=1050071&projectid=50061&pathid=7
Status	New

The buffer allocated by <= in h2o/util.c at line 155 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	h2o/util.c	h2o/util.c
Line	162	162
Object	<=	<=

Code Snippet
File Name h2o/util.c
Method void test_build_destination(void)

```
....
162.      for (escape = 0; escape <= 1; escape++) {
```

Potential Off by One Error in Loops\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=8
Status	New

The buffer allocated by <= in h2o/util.c at line 205 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	h2o/util.c	h2o/util.c
Line	246	246
Object	<=	<=

Code Snippet

File Name	h2o/util.c
Method	void test_build_destination_escaping(void)

```
....
246.      for (j = 0; j <= 1; j++) {
```

Inconsistent Implementations

Query Path:
CPP\Cx\CPP Low Visibility\Inconsistent Implementations Version:0

Description

Inconsistent Implementations\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=1
Status	New

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	997	997
Object	getopt	getopt

Code Snippet	
File Name	h2o/yc.c
Method	int main(int argc, char** argv) {

```
....
997.          int c = getopt(argc, argv, "s:p:c:dtqvh");
```

Potential Path Traversal

Query Path:
CPP\Cx\CPP Low Visibility\Potential Path Traversal Version:0

Categories

OWASP Top 10 2013: A4-Insecure Direct Object References
OWASP Top 10 2017: A5-Broken Access Control

Description

Potential Path Traversal\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=2
Status	New

Method main at line 989 of h2o/yc.c gets user input from the argv element. This element's value then flows through the code and is eventually used in a file path for local disk access in read_data at line 129 of h2o/yc.c. This may cause a Path Traversal vulnerability.

	Source	Destination
File	h2o/yc.c	h2o/yc.c
Line	989	134
Object	argv	filename

Code Snippet
File Name h2o/yc.c
Method int main(int argc, char** argv) {

```
....
989.  int main(int argc, char** argv) {
```

File Name h2o/yc.c
Method static size_t read_data(const char* filename, char** pdata) {

```
....
134.          fd = open(filename, O_RDONLY);
```

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=1050071&projectid=50061&pathid=32
Status	New

The variable declared in null at h2o/fusion.c in line 289 is not initialized when it is used by supp at h2o/fusion.c in line 289.

	Source	Destination
File	h2o/fusion.c	h2o/fusion.c
Line	293	340
Object	null	supp

Code Snippet
File Name h2o/fusion.c
Method static void gcm_test_vectors(void)

```
....
293.      ptls_aead_supplementary_encryption_t *supp = NULL;
....
340.      ptls_cipher_free(supp->ctx);
```

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=1050071&projectid=50061&pathid=493
Status	New

The read_data method in h2o/yc.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	h2o/yc.c	h2o/yc.c
Line	134	134
Object	open	open

Code Snippet
File Name h2o/yc.c
Method static size_t read_data(const char* filename, char** pdata) {

```
....
134.          fd = open(filename, O_RDONLY);
```

Information Exposure Through Comments

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

Categories

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

Description

Information Exposure Through Comments\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050071&projectid=50061&pathid=494
Status	New

	Source	Destination

File	h2o/cli.c	h2o/cli.c
Line	1363	1363
Object	cipher-	cipher-

Code Snippet
File Name h2o/cli.c
Method /* Amend cipher-suites. Copy the defaults when `-y` option is not used. Otherwise, complain if aes128gcmsha256 is not specified

```
....
1363.      /* Amend cipher-suites. Copy the defaults when `-y` option is
not used. Otherwise, complain if aes128gcmsha256 is not specified
```

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));
    }
}
```

Wrong Size t Allocation

Risk

What might happen

Incorrect allocation of memory may result in unexpected behavior by either overwriting sections of memory with unexpected values. Under certain conditions where both an incorrect allocation of memory and the values being written can be controlled by an attacker, such an issue may result in execution of malicious code.

Cause

How does it happen

Some memory allocation functions require a size value to be provided as a parameter. The allocated size should be derived from the provided value, by providing the length value of the intended source, multiplied by the size of that length. Failure to perform the correct arithmetic to obtain the exact size of the value will likely result in the source overflowing its destination.

General Recommendations

How to avoid it

- Always perform the correct arithmetic to determine size.
- Specifically for memory allocation, calculate the allocation size from the allocation source:
  - Derive the size value from the length of intended source to determine the amount of units to be processed.
  - Always programmatically consider the size of the each unit and their conversion to memory units - for example, by using sizeof() on the unit's type.
  - Memory allocation should be a multiplication of the amount of units being written, times the size of each unit.

Source Code Examples

CPP

Allocating and Assigning Memory without Sizeof Arithmetic

```
int *ptr;
ptr = (int*)malloc(5);
for (int i = 0; i < 5; i++)
{
      ptr[i] = i * 2 + 1;
}
```

Allocating and Assigning Memory with Sizeof Arithmetic

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
```

```
for (int i = 0; i < 5; i++)
{
      ptr[i] = i * 2 + 1;
}
```

Incorrect Arithmetic of Multi-Byte String Allocation

```
wchar_t * dest;
dest = (wchar_t *)malloc(wcslen(source) + 1); // Would not crash for a short "source"
wcscpy((wchar_t *)dest, source);
wprintf(L"Dest: %s\r\n", dest);
```

Correct Arithmetic of Multi-Byte String Allocation

```
wchar_t * dest;
dest = (wchar_t *)malloc((wcslen(source) + 1) * sizeof(wchar_t));
wcscpy((wchar_t *)dest, source);
wprintf(L"Dest: %s\r\n", dest);
```

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;
}
```

Use of Zero Initialized Pointer

Risk

What might happen

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

Cause

How does it happen

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

General Recommendations

How to avoid it

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

Source Code Examples

CPP

Explicit NULL Dereference

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

Implicit NULL Dereference

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

Java

Explicit Null Dereference

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

Use of Function with Inconsistent Implementations

Weakness ID: 474 *(Weakness Base)* **Status:** Draft

Description

Description Summary

The code uses a function that has inconsistent implementations across operating systems and versions, which might cause security-relevant portability problems.

Time of Introduction

- Architecture and Design
- Implementation

Applicable Platforms

Languages

C: *(Often)*

PHP: *(Often)*

All

Potential Mitigations

Do not accept inconsistent behavior from the API specifications when the deviant behavior increase the risk level.

Other Notes

The behavior of functions in this category varies by operating system, and at times, even by operating system version. Implementation differences can include:

- Slight differences in the way parameters are interpreted leading to inconsistent results.
- Some implementations of the function carry significant security risks.
- The function might not be defined on all platforms.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	398	Indicator of Poor Code Quality	**Development Concepts (primary)699 Seven Pernicious Kingdoms (primary)700 Research Concepts (primary)1000**
ParentOf	Weakness Variant	589	Call to Non-ubiquitous API	**Research Concepts (primary)1000**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
7 Pernicious Kingdoms			Inconsistent Implementations

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 Potential Mitigations, Time of Introduction		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Relationships, Other Notes, Taxonomy Mappings		
Previous Entry Names			
Change Date	**Previous Entry Name**		
2008-04-11	Inconsistent Implementations		

BACK TO TOP

Potential Path Traversal

Risk

What might happen

An attacker could define any arbitrary file path for the application to use, potentially leading to:

- Stealing sensitive files, such as configuration or system files
- Overwriting files such as program binaries, configuration files, or system files
- Deleting critical files, causing a denial of service (DoS).

Cause

How does it happen

The application uses user input in the file path for accessing files on the application server's local disk. This enables an attacker to arbitrarily determine the file path.

General Recommendations

How to avoid it

1. Ideally, avoid depending on user input for file selection.
2. Validate all input, regardless of source. Validation should be based on a whitelist: accept only data fitting a specified structure, rather than reject bad patterns. Check for:
   - Data type
   - Size
   - Range
   - Format
   - Expected values
3. Accept user input only for the filename, not for the path and folders.
4. Ensure that file path is fully canonicalized.
5. Explicitly limit the application to using a designated folder that separate from the applications binary folder.
6. Restrict the privileges of the application's OS user to necessary files and folders. The application should not be able to write to the application binary folder, and should not read anything outside of the application folder and data folder.

Source Code Examples

CSharp

Using unvalidated user input as the file name may enable the user to access arbitrary files on the server local disk

```
public class PathTraversal
{
        private void foo(TextBox textbox1)
    {
                string fileNum = textbox1.Text;
                string path = "c:\files\file" + fileNum;
                FileStream f = new FileStream(path, FileMode.Open);
                byte[] output = new byte[10];
                f.Read(output,0, 10);
```

```
        }
}
```

Potentially hazardous characters are removed from the user input before use

```
public class PathTraversalFixed
{
        private void foo(TextBox textbox1)
    {
                string fileNum = textbox1.Text.Replace("\", "").Replace("..", "");
         string path = "c:\files\file" + fileNum;
                FileStream f = new FileStream(path, FileMode.Open);
                byte[] output = new byte[10];
                f.Read(output,0, 10);
        }
}
```

Java

Using unvalidated user input as the file name may enable the user to access arbitrary files on the server local disk

```
public class Absolute_Path_Traversal {
    public static void main(String[] args) {
        Scanner userInputScanner = new Scanner(System.in);
        System.out.print("\nEnter file name: ");
        String name = userInputScanner.nextLine();
        String path = "c:\files\file" + name;
        try {
            BufferedReader reader = new BufferedReader(new FileReader(path));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Potentially hazardous characters are removed from the user input before use

```
public class Absolute_Path_Traversal_Fixed {
    public static void main(String[] args) {
        Scanner userInputScanner = new Scanner(System.in);
        System.out.print("\nEnter file name: ");
        String name = userInputScanner.nextLine();
        name = name.replace("/", "").replace("..", "");
        String path = "c:\files\file" + name;
        try {
            BufferedReader reader = new BufferedReader(new FileReader(path));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

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

Improper Validation of Array Index

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

Description

Description Summary

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

Alternate Terms

out-of-bounds array index

index-out-of-range

array index underflow

Time of Introduction

- Implementation

Applicable Platforms

Languages

C: *(Often)*

C++: *(Often)*

Language-independent

Common Consequences

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

Likelihood of Exploit

High

Detection Methods

Automated Static Analysis

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

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

Effectiveness: High

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

Automated Dynamic Analysis

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

Black Box

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

Demonstrative Examples

Example 1

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

(Bad Code)

Example Language: **C**

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

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

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

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

(Good Code)

Example Language: **C**

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

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

// continue read from socket until buf only contains '.'
if (DOTLINE(buf))
```

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

Example 2

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

(Bad Code)

Example Language: **Java**

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

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

Example 3

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

(Bad Code)

Example Language: **Java**

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

String productSummary = new String("");

try {
String productSummary = getProductSummary(index);

} catch (Exception ex) {...}

return productSummary;
}

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

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

(Good Code)

Example Language: **Java**

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

String productSummary = new String("");
```

```
try {
String productSummary = getProductSummary(index);

} catch (Exception ex) {...}

return productSummary;
}

public String getProductSummary(int index) {
String productSummary = "";

if ((index >= 0) && (index < MAX_PRODUCTS)) {
productSummary = products[index];
}
else {
System.err.println("index is out of bounds");
throw new IndexOutOfBoundsException();
}

return productSummary;
}
```

An alternative in Java would be to use one of the collection objects such as ArrayList that will automatically generate an exception if an attempt is made to access an array index that is out of bounds.

(Good Code)

Example Language: **Java**

```
ArrayList productArray = new ArrayList(MAX_PRODUCTS);
...
try {
productSummary = (String) productArray.get(index);
} catch (IndexOutOfBoundsException ex) {...}
```

Observed Examples

Reference	Description
CVE-2005-0369	large ID in packet used as array index
CVE-2001-1009	negative array index as argument to POP LIST command
CVE-2003-0721	Integer signedness error leads to negative array index
CVE-2004-1189	product does not properly track a count and a maximum number, which can lead to resultant array index overflow.
CVE-2007-5756	chain: device driver for packet-capturing software allows access to an unintended IOCTL with resultant array index error.

Potential Mitigations

Phase: Architecture and Design

Strategies: Input Validation; Libraries or Frameworks

Use an input validation framework such as Struts or the OWASP ESAPI Validation API. If you use Struts, be mindful of weaknesses covered by the CWE-101 category.

Phase: Architecture and Design

For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side, in order to avoid CWE-602. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the server.

Even though client-side checks provide minimal benefits with respect to server-side security, they are still useful. First, they can support intrusion detection. If the server receives input that should have been rejected by the client, then it may be an indication of an attack. Second, client-side error-checking can provide helpful feedback to the user about the expectations for valid input. Third, there may be a reduction in server-side processing time for accidental input errors, although this is typically a small savings.

Phase: Requirements

Strategy: Language Selection

Use a language with features that can automatically mitigate or eliminate out-of-bounds indexing errors.

For example, Ada allows the programmer to constrain the values of a variable and languages such as Java and Ruby will allow the programmer to handle exceptions when an out-of-bounds index is accessed.

Phase: Implementation

Strategy: Input Validation

Assume all input is malicious. Use an "accept known good" input validation strategy (i.e., use a whitelist). Reject any input that does not strictly conform to specifications, or transform it into something that does. Use a blacklist to reject any unexpected inputs and detect potential attacks.

When accessing a user-controlled array index, use a stringent range of values that are within the target array. Make sure that you do not allow negative values to be used. That is, verify the minimum as well as the maximum of the range of acceptable values.

Phase: Implementation

Be especially careful to validate your input when you invoke code that crosses language boundaries, such as from an interpreted language to native code. This could create an unexpected interaction between the language boundaries. Ensure that you are not violating any of the expectations of the language with which you are interfacing. For example, even though Java may not be susceptible to buffer overflows, providing a large argument in a call to native code might trigger an overflow.

Weakness Ordinalities

Ordinality	Description
Resultant	The most common condition situation leading to unchecked array indexing is the use of loop index variables as buffer indexes. If the end condition for the loop is subject to a flaw, the index can grow or shrink unbounded, therefore causing a buffer overflow or underflow. Another common situation leading to this condition is the use of a function's return value, or the resulting value of a calculation directly as an index in to a buffer.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	20	Improper Input Validation	**Development Concepts (primary)699 Research Concepts (primary)1000**
ChildOf	Category	189	Numeric Errors	Development Concepts699
ChildOf	Category	633	Weaknesses that Affect Memory	**Resource-specific Weaknesses (primary)631**
ChildOf	Category	738	CERT C Secure Coding Section 04 - Integers (INT)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
ChildOf	Category	740	CERT C Secure Coding Section 06 - Arrays (ARR)	Weaknesses Addressed by the CERT C Secure Coding Standard734
ChildOf	Category	802	2010 Top 25 - Risky Resource Management	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
CanPrecede	Weakness Class	119	Failure to Constrain Operations within the Bounds of a Memory Buffer	Research Concepts1000
CanPrecede	Weakness Variant	789	Uncontrolled Memory Allocation	Research Concepts1000
PeerOf	Weakness Base	124	Buffer Underwrite ('Buffer Underflow')	Research Concepts1000

Theoretical Notes

An improperly validated array index might lead directly to the always-incorrect behavior of "access of array using out-of-bounds index."

Affected Resources

- Memory

f Causal Nature

Explicit

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Unchecked array indexing
PLOVER			INDEX - Array index overflow
CERT C Secure Coding	ARR00-C		Understand how arrays work
CERT C Secure Coding	ARR30-C		Guarantee that array indices are within the valid range
CERT C Secure Coding	ARR38-C		Do not add or subtract an integer to a pointer if the resulting value does not refer to a valid array element
CERT C Secure Coding	INT32-C		Ensure that operations on signed integers do not result in overflow

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
100	Overflow Buffers	

References

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 5, "Array Indexing Errors" Page 144. 2nd Edition. Microsoft. 2002.

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	CLASP		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Sean Eidemiller	Cigital	External
	added/updated demonstrative examples		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Alternate Terms, Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Description, Name, Relationships		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Observed Examples, Other Notes, Potential Mitigations, Theoretical Notes, Weakness Ordinalities		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Demonstrative Examples, Detection Factors, Likelihood of Exploit, Potential Mitigations, References, Related Attack Patterns, Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Related Attack Patterns		
Previous Entry Names			
Change Date	**Previous Entry Name**		
2009-10-29	Unchecked Array Indexing		

BACK TO TOP

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

Exposure of System Data to Unauthorized Control Sphere

Risk

What might happen

System data can provide attackers with valuable insights on systems and services they are targeting - any type of system data, from service version to operating system fingerprints, can assist attackers to hone their attack, correlate data with known vulnerabilities or focus efforts on developing new attacks against specific technologies.

Cause

How does it happen

System data is read and subsequently exposed where it might be read by untrusted entities.

General Recommendations

How to avoid it

Consider the implications of exposure of the specified input, and expected level of access to the specified output. If not required, consider removing this code, or modifying exposed information to exclude potentially sensitive system data.

Source Code Examples

Java

Leaking Environment Variables in JSP Web-Page

```
String envVarValue = System.getenv(envVar);
if (envVarValue == null) {
    out.println("Environment variable is not defined:");
    out.println(System.getenv());
} else {
    //[..]
};
```

TOCTOU

Risk

What might happen

At best, a Race Condition may cause errors in accuracy, overidden values or unexpected behavior that may result in denial-of-service. At worst, it may allow attackers to retrieve data or bypass security processes by replaying a controllable Race Condition until it plays out in their favor.

Cause

How does it happen

Race Conditions occur when a public, single instance of a resource is used by multiple concurrent logical processes. If the these logical processes attempt to retrieve and update the resource without a timely management system, such as a lock, a Race Condition will occur.

An example for when a Race Condition occurs is a resource that may return a certain value to a process for further editing, and then updated by a second process, resulting in the original process' data no longer being valid. Once the original process edits and updates the incorrect value back into the resource, the second process' update has been overwritten and lost.

General Recommendations

How to avoid it

When sharing resources between concurrent processes across the application ensure that these resources are either thread-safe, or implement a locking mechanism to ensure expected concurrent activity.

Source Code Examples

Java

Different Threads Increment and Decrement The Same Counter Repeatedly, Resulting in a Race Condition

```
	public static int counter = 0;
	public static void start() throws InterruptedException {
		incrementCounter ic;
		decrementCounter dc;
		while(counter == 0) {
			counter = 0;
			ic = new incrementCounter();
			dc = new decrementCounter();
			ic.start();
			dc.start();
			ic.join();
			dc.join();
		}
		System.out.println(counter); //Will stop and return either -1 or 1 due to race
condition over counter
	}

	public static class incrementCounter extends Thread {
	    public void run() {
		counter++;
	    }
```

```
    }

    public static class decrementCounter extends Thread {
        public void run() {
           counter--;
        }
    }
```

Different Threads Increment and Decrement The Same Thread-Safe Counter Repeatedly, Never Resulting in a Race Condition

```
    public static int counter = 0;
    public static Object lock = new Object();

    public static void start() throws InterruptedException {
           incrementCounter ic;
           decrementCounter dc;
           while(counter == 0) { // because of proper locking, this condition is never false
                  counter = 0;
                  ic = new incrementCounter();
                  dc = new decrementCounter();
                  ic.start();
                  dc.start();
                  ic.join();
                  dc.join();
           }
           System.out.println(counter); // Never reached
    }

    public static class incrementCounter extends Thread {
        public void run() {
           synchronized (lock) {
                  counter++;
           }
        }
    }

    public static class decrementCounter extends Thread {
        public void run() {
           synchronized (lock) {
                  counter--;
           }
        }
    }
```

Information Leak Through Comments

Weakness ID: 615 *(Weakness Variant)* **Status:** Incomplete

Description

Description Summary

While adding general comments is very useful, some programmers tend to leave important data, such as: filenames related to the web application, old links or links which were not meant to be browsed by users, old code fragments, etc.

Extended Description

An attacker who finds these comments can map the application's structure and files, expose hidden parts of the site, and study the fragments of code to reverse engineer the application, which may help develop further attacks against the site.

Time of Introduction

- Implementation

Demonstrative Examples

Example 1

The following comment, embedded in a JSP, will be displayed in the resulting HTML output.

(Bad Code)

Example Languages: **HTML and JSP**

```
<!-- FIXME: calling this with more than 30 args kills the JDBC server -->
```

Observed Examples

Reference	Description
CVE-2007-6197	Version numbers and internal hostnames leaked in HTML comments.
CVE-2007-4072	CMS places full pathname of server in HTML comment.
CVE-2009-2431	blog software leaks real username in HTML comment.

Potential Mitigations

Remove comments which have sensitive information about the design/implementation of the application. Some of the comments may be exposed to the user and affect the security posture of the application.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Variant	540	Information Leak Through Source Code	**Development Concepts (primary)699 Research Concepts (primary)1000**

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	Anonymous Tool Vendor (under NDA)		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Sean Eidemiller	Cigital	External
	added/updated demonstrative examples		
2008-07-01	Eric Dalci	Cigital	External
	updated Potential Mitigations, Time of Introduction		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2008-10-14	CWE Content Team	MITRE	Internal
	updated Description		
2009-03-10	CWE Content Team	MITRE	Internal

	updated Demonstrative Examples		
2009-07-27	CWE Content Team	MITRE	Internal
	updated Observed Examples, Taxonomy Mappings		

BACK TO TOP

Scanned Languages

Language	Hash Number	Change Date
CPP	4541647240435660	6/19/2024
Common	0105849645654507	6/19/2024