ossec-hids Scan Report

Project Name	ossec-hids
Scan Start	Friday, June 21, 2024 10:49:43 PM
Preset	Checkmarx Default
Scan Time	00h:04m:36s
Lines Of Code Scanned	9588
Files Scanned	13
Report Creation Time	Friday, June 21, 2024 10:57:20 PM
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049
Team	CxServer
Checkmarx Version	8.7.0
Scan Type	Full
Source Origin	LocalPath
Density	2/100 (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

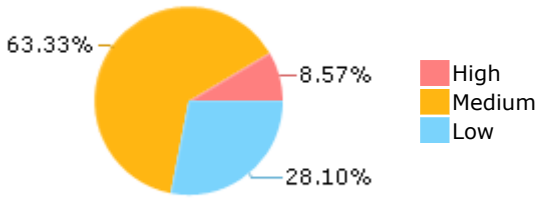
Results Limit

Results limit per query was set to 50

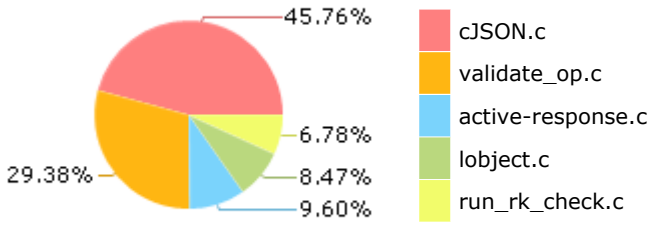
Selected Queries

Selected queries are listed in Result Summary

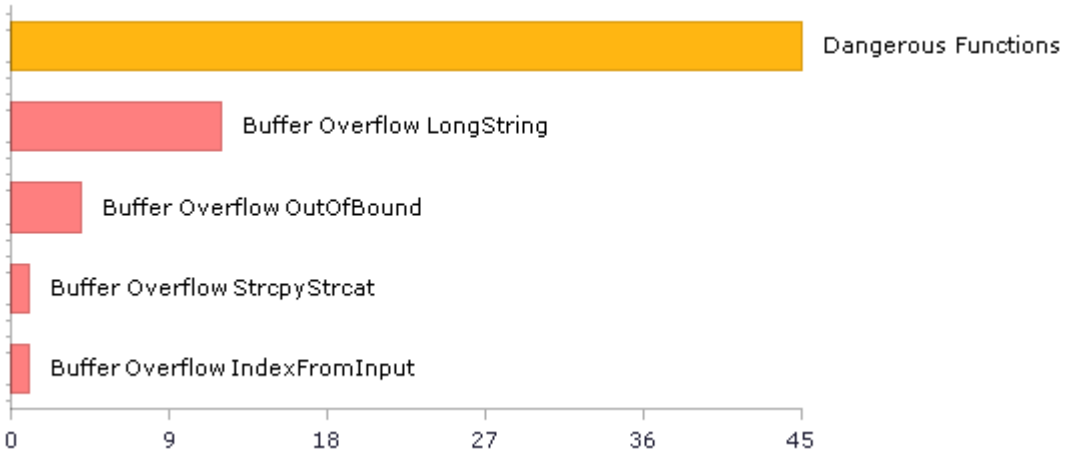
Result Summary



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	57	40
A2-Broken Authentication	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	15	15
A3-Sensitive Data Exposure	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	App. Specific	0	0
A4-XML External Entities (XXE)	App. Specific	AVERAGE	COMMON	EASY	SEVERE	App. Specific	0	0
A5-Broken Access Control*	App. Specific	AVERAGE	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A6-Security Misconfiguration	App. Specific	EASY	WIDESPREAD	EASY	MODERATE	App. Specific	0	0
A7-Cross-Site Scripting (XSS)	App. Specific	EASY	WIDESPREAD	EASY	MODERATE	App. Specific	0	0
A8-Insecure Deserialization	App. Specific	DIFFICULT	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A9-Using Components with Known Vulnerabilities*	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	MODERATE	App. Specific	45	45
A10-Insufficient Logging & Monitoring	App. Specific	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	App. Specific	0	0

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

Scan Summary - OWASP Top 10 2013

Further details and elaboration about vulnerabilities and risks can be found at: OWASP Top 10 2013

Category	Threat Agent	Attack Vectors	Weakness Prevalence	Weakness Detectability	Technical Impact	Business Impact	Issues Found	Best Fix Locations
A1-Injection	EXTERNAL, INTERNAL, ADMIN USERS	EASY	COMMON	AVERAGE	SEVERE	ALL DATA	0	0
A2-Broken Authentication and Session Management	EXTERNAL, INTERNAL USERS	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	AFFECTED DATA AND FUNCTIONS	0	0
A3-Cross-Site Scripting (XSS)	EXTERNAL, INTERNAL, ADMIN USERS	AVERAGE	VERY WIDESPREAD	EASY	MODERATE	AFFECTED DATA AND SYSTEM	0	0
A4-Insecure Direct Object References	SYSTEM USERS	EASY	COMMON	EASY	MODERATE	EXPOSED DATA	0	0
A5-Security Misconfiguration	EXTERNAL, INTERNAL, ADMIN USERS	EASY	COMMON	EASY	MODERATE	ALL DATA AND SYSTEM	0	0
A6-Sensitive Data Exposure	EXTERNAL, INTERNAL, ADMIN USERS, USERS BROWSERS	DIFFICULT	UNCOMMON	AVERAGE	SEVERE	EXPOSED DATA	0	0
A7-Missing Function Level Access Control*	EXTERNAL, INTERNAL USERS	EASY	COMMON	AVERAGE	MODERATE	EXPOSED DATA AND FUNCTIONS	0	0
A8-Cross-Site Request Forgery (CSRF)	USERS BROWSERS	AVERAGE	COMMON	EASY	MODERATE	AFFECTED DATA AND FUNCTIONS	0	0
A9-Using Components with Known Vulnerabilities*	EXTERNAL USERS, AUTOMATED TOOLS	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	AFFECTED DATA AND FUNCTIONS	45	45
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	4	4
PCI DSS (3.2) - 6.5.2 - Buffer overflows	41	32
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.	8	8
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.	1	1
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.	8	8
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.	3	3

***** 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)	16	16
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)*	1	1
SC-28 Protection of Information at Rest (P1)	0	0
SC-4 Information in Shared Resources (P1)	0	0
SC-5 Denial of Service Protection (P1)*	62	22
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*	20	11
SI-11 Error Handling (P2)*	15	15
SI-15 Information Output Filtering (P0)	0	0
SI-16 Memory Protection (P1)	4	4

***** 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

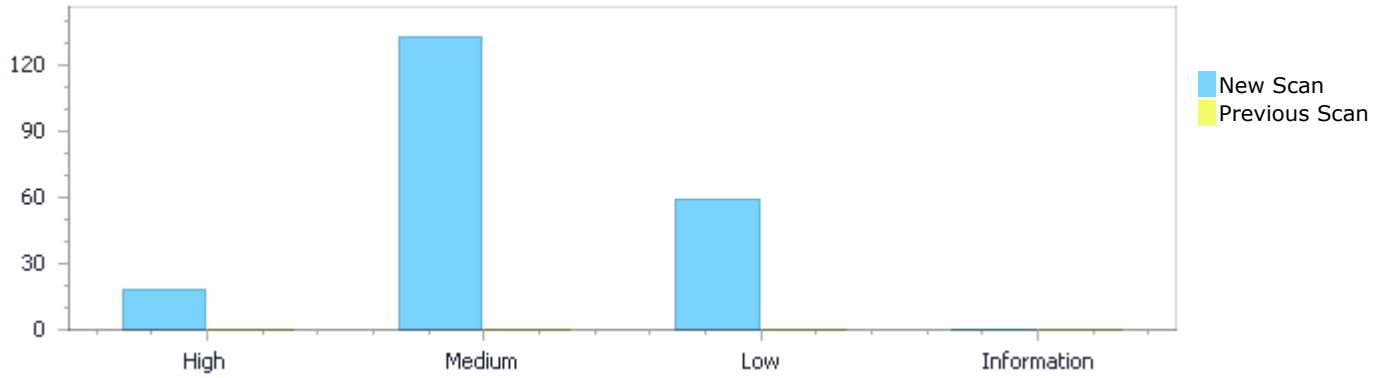
Scan Summary - Custom

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	18	133	59	0	210
Recurrent Issues	0	0	0	0	0
Total	18	133	59	0	210
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	18	133	59	0	210
Urgent	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0
Total	18	133	59	0	210

Result Summary

Vulnerability Type	Occurrences	Severity
Buffer Overflow LongString	12	High
Buffer Overflow OutOfBound	4	High
Buffer Overflow IndexFromInput	1	High
Buffer Overflow StrcpyStrcat	1	High
Dangerous Functions	45	Medium

Use of Zero Initialized Pointer	41	Medium
Buffer Overflow boundcpy WrongSizeParam	21	Medium
MemoryFree on StackVariable	14	Medium
Memory Leak	7	Medium
Short Overflow	3	Medium
Use After Free	1	Medium
Wrong Size t Allocation	1	Medium
Unchecked Return Value	15	Low
NULL Pointer Dereference	13	Low
Incorrect Permission Assignment For Critical Resources	8	Low
Improper Resource Access Authorization	7	Low
TOCTOU	7	Low
Potential Off by One Error in Loops	4	Low
Use of Sizeof On a Pointer Type	3	Low
Exposure of System Data to Unauthorized Control Sphere	1	Low
Reliance on DNS Lookups in a Decision	1	Low

10 Most Vulnerable Files

High and Medium Vulnerabilities

File Name	Issues Found
ossec-hids/cJSON.c	66
ossec-hids/validate_op.c	38
ossec-hids/lobject.c	14
ossec-hids/active-response.c	9
ossec-hids/imsg.c	8
ossec-hids/puff.c	6
ossec-hids/lvm.c	4
ossec-hids/b64.c	3
ossec-hids/blast.c	2
ossec-hids/run_rk_check.c	1

Scan Results Details

Buffer Overflow LongString

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

Categories

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

Description

Buffer Overflow LongString\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=1
Status	New

The size of the buffer used by *OS_IsValidTime in first_hour, at line 537 of ossec-hids/validate_op.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *__gethour passes to "%02d:%02d", at line 454 of ossec-hids/validate_op.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	505	597
Object	"%02d:%02d"	first_hour

Code Snippet
File Name ossec-hids/validate_op.c
Method static const char *__gethour(const char *str, char *ossec_hour)

```
....
505.              snprintf(ossec_hour, 6, "%02d:%02d", chour, cmin);
```

File Name ossec-hids/validate_op.c
Method char *OS_IsValidTime(const char *time_str)

```
....
597.      if (strcmp(first_hour, second_hour) > 0) {
```

Buffer Overflow LongString\Path 2:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=2

Status	New

The size of the buffer used by *OS_IsValidTime in first_hour, at line 537 of ossec-hids/validate_op.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *__gethour passes to "%02d:%02d", at line 454 of ossec-hids/validate_op.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	505	566
Object	"%02d:%02d"	first_hour

Code Snippet
File Name ossec-hids/validate_op.c
Method static const char *__gethour(const char *str, char *ossec_hour)

```
....
505.              snprintf(ossec_hour, 6, "%02d:%02d", chour, cmin);
```

File Name ossec-hids/validate_op.c
Method char *OS_IsValidTime(const char *time_str)

```
....
566.      time_str = __gethour(time_str, first_hour);
```

Buffer Overflow LongString\Path 3:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=3
Status	New

The size of the buffer used by *OS_IsValidTime in second_hour, at line 537 of ossec-hids/validate_op.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *__gethour passes to "%02d:%02d", at line 454 of ossec-hids/validate_op.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	505	584
Object	"%02d:%02d"	second_hour

Code Snippet
File Name ossec-hids/validate_op.c
Method static const char *__gethour(const char *str, char *ossec_hour)

```
....
505.              snprintf(ossec_hour, 6, "%02d:%02d", chour, cmin);
```

File Name ossec-hids/validate_op.c

Method char *OS_IsValidTime(const char *time_str)

```
....
584.      time_str = __gethour(time_str, second_hour);
```

Buffer Overflow LongString\Path 4:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=4
Status	New

The size of the buffer used by *OS_IsValidTime in second_hour, at line 537 of ossec-hids/validate_op.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *__gethour passes to "%02d:%02d", at line 454 of ossec-hids/validate_op.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	505	597
Object	"%02d:%02d"	second_hour

Code Snippet

File Name ossec-hids/validate_op.c

Method static const char *__gethour(const char *str, char *ossec_hour)

```
....
505.              snprintf(ossec_hour, 6, "%02d:%02d", chour, cmin);
```

File Name ossec-hids/validate_op.c

Method char *OS_IsValidTime(const char *time_str)

```
....
597.      if (strcmp(first_hour, second_hour) > 0) {
```

Buffer Overflow LongString\Path 5:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=5
Status	New

The size of the buffer used by *OS_IsValidTime in first_hour, at line 537 of ossec-hids/validate_op.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *__gethour passes to "%02d:%02d", at line 454 of ossec-hids/validate_op.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	522	597
Object	"%02d:%02d"	first_hour

Code Snippet
File Name ossec-hids/validate_op.c
Method static const char *__gethour(const char *str, char *ossec_hour)

```
....
522.              snprintf(ossec_hour, 6, "%02d:%02d", chour, cmin);
```

File Name ossec-hids/validate_op.c
Method char *OS_IsValidTime(const char *time_str)

```
....
597.      if (strcmp(first_hour, second_hour) > 0) {
```

Buffer Overflow LongString\Path 6:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=6
Status	New

The size of the buffer used by *OS_IsValidTime in first_hour, at line 537 of ossec-hids/validate_op.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *__gethour passes to "%02d:%02d", at line 454 of ossec-hids/validate_op.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	522	566
Object	"%02d:%02d"	first_hour

Code Snippet
File Name ossec-hids/validate_op.c
Method static const char *__gethour(const char *str, char *ossec_hour)

```
....
522.              snprintf(ossec_hour, 6, "%02d:%02d", chour, cmin);
```

File Name ossec-hids/validate_op.c

Method char *OS_IsValidTime(const char *time_str)

```
....
566.      time_str = __gethour(time_str, first_hour);
```

Buffer Overflow LongString\Path 7:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=7
Status	New

The size of the buffer used by *OS_IsValidTime in second_hour, at line 537 of ossec-hids/validate_op.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *__gethour passes to "%02d:%02d", at line 454 of ossec-hids/validate_op.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	522	584
Object	"%02d:%02d"	second_hour

Code Snippet

File Name ossec-hids/validate_op.c

Method static const char *__gethour(const char *str, char *ossec_hour)

```
....
522.              snprintf(ossec_hour, 6, "%02d:%02d", chour, cmin);
```

File Name ossec-hids/validate_op.c

Method char *OS_IsValidTime(const char *time_str)

```
....
584.      time_str = __gethour(time_str, second_hour);
```

Buffer Overflow LongString\Path 8:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=8
Status	New

The size of the buffer used by *OS_IsValidTime in second_hour, at line 537 of ossec-hids/validate_op.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source

buffer that *__gethour passes to "%02d:%02d", at line 454 of ossec-hids/validate_op.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	522	597
Object	"%02d:%02d"	second_hour

Code Snippet
File Name ossec-hids/validate_op.c
Method static const char *__gethour(const char *str, char *ossec_hour)

```
....
522.              snprintf(ossec_hour, 6, "%02d:%02d", chour, cmin);
```

File Name ossec-hids/validate_op.c
Method char *OS_IsValidTime(const char *time_str)

```
....
597.      if (strcmp(first_hour, second_hour) > 0) {
```

Buffer Overflow LongString\Path 9:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=9
Status	New

The size of the buffer used by *OS_IsValidTime in first_hour, at line 537 of ossec-hids/validate_op.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *__gethour passes to "%02d:%02d", at line 454 of ossec-hids/validate_op.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	528	566
Object	"%02d:%02d"	first_hour

Code Snippet
File Name ossec-hids/validate_op.c
Method static const char *__gethour(const char *str, char *ossec_hour)

```
....
528.          snprintf(ossec_hour, 6, "%02d:%02d", chour, cmin);
```

File Name ossec-hids/validate_op.c

Method	char *OS_IsValidTime(const char *time_str)

```
....
566.      time_str = __gethour(time_str, first_hour);
```

Buffer Overflow LongString\Path 10:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=10
Status	New

The size of the buffer used by *OS_IsValidTime in first_hour, at line 537 of ossec-hids/validate_op.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *__gethour passes to "%02d:%02d", at line 454 of ossec-hids/validate_op.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	528	597
Object	"%02d:%02d"	first_hour

Code Snippet

File Name	ossec-hids/validate_op.c
Method	static const char *__gethour(const char *str, char *ossec_hour)

```
....
528.          snprintf(ossec_hour, 6, "%02d:%02d", chour, cmin);
```

File Name	ossec-hids/validate_op.c
Method	char *OS_IsValidTime(const char *time_str)

```
....
597.      if (strcmp(first_hour, second_hour) > 0) {
```

Buffer Overflow LongString\Path 11:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=11
Status	New

The size of the buffer used by *OS_IsValidTime in second_hour, at line 537 of ossec-hids/validate_op.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *__gethour passes to "%02d:%02d", at line 454 of ossec-hids/validate_op.c, to overwrite the target buffer.

	Source	Destination

File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	528	597
Object	"%02d:%02d"	second_hour

Code Snippet
File Name ossec-hids/validate_op.c
Method static const char *__gethour(const char *str, char *ossec_hour)

```
....
528.          snprintf(ossec_hour, 6, "%02d:%02d", chour, cmin);
```

File Name ossec-hids/validate_op.c
Method char *OS_IsValidTime(const char *time_str)

```
....
597.      if (strcmp(first_hour, second_hour) > 0) {
```

Buffer Overflow LongString\Path 12:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=12
Status	New

The size of the buffer used by *OS_IsValidTime in second_hour, at line 537 of ossec-hids/validate_op.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *__gethour passes to "%02d:%02d", at line 454 of ossec-hids/validate_op.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	528	584
Object	"%02d:%02d"	second_hour

Code Snippet
File Name ossec-hids/validate_op.c
Method static const char *__gethour(const char *str, char *ossec_hour)

```
....
528.          snprintf(ossec_hour, 6, "%02d:%02d", chour, cmin);
```

File Name ossec-hids/validate_op.c
Method char *OS_IsValidTime(const char *time_str)

```
....
584.      time_str = __gethour(time_str, second_hour);
```

Buffer Overflow OutOfBound

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

Categories

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

Description

Buffer Overflow OutOfBound\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=13
Status	New

The size of the buffer used by codes in symbol, at line 436 of ossec-hids/puff.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that codes passes to lens, at line 436 of ossec-hids/puff.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/puff.c	ossec-hids/puff.c
Line	443	477
Object	lens	symbol

Code Snippet
File Name ossec-hids/puff.c
Method local int codes(struct state *s,

```
....
443.      static const short lens[29] = { /* Size base for length codes
257..285 */
....
477.              len = lens[symbol] + bits(s, lext[symbol]);
```

Buffer Overflow OutOfBound\Path 2:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=14
Status	New

The size of the buffer used by codes in symbol, at line 436 of ossec-hids/puff.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that codes passes to lext, at line 436 of ossec-hids/puff.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/puff.c	ossec-hids/puff.c
Line	446	477
Object	lext	symbol

Code Snippet
File Name ossec-hids/puff.c
Method local int codes(struct state *s,

```
....
446.      static const short lext[29] = { /* Extra bits for length codes
257..285 */
....
477.              len = lens[symbol] + bits(s, lext[symbol]);
```

Buffer Overflow OutOfBound\Path 3:

Severity High
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=15
Status New

The size of the buffer used by codes in symbol, at line 436 of ossec-hids/puff.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that codes passes to dists, at line 436 of ossec-hids/puff.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/puff.c	ossec-hids/puff.c
Line	449	483
Object	dists	symbol

Code Snippet
File Name ossec-hids/puff.c
Method local int codes(struct state *s,

```
....
449.      static const short dists[30] = { /* Offset base for distance
codes 0..29 */
....
483.              dist = dists[symbol] + bits(s, dext[symbol]);
```

Buffer Overflow OutOfBound\Path 4:

Severity High
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=16
Status New

The size of the buffer used by codes in symbol, at line 436 of ossec-hids/puff.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that codes passes to dext, at line 436 of ossec-hids/puff.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/puff.c	ossec-hids/puff.c
Line	453	483
Object	dext	symbol

Code Snippet
File Name ossec-hids/puff.c
Method local int codes(struct state *s,

```
....
453.      static const short dext[30] = { /* Extra bits for distance
codes 0..29 */
....
483.              dist = dists[symbol] + bits(s, dext[symbol]);
```

Buffer Overflow StrcpyStrcat

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

Categories

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

Description
Buffer Overflow StrcpyStrcat\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=76
Status	New

The size of the buffer used by print_string_ptr in output, at line 828 of ossec-hids/cJSON.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that print_string_ptr passes to input, at line 828 of ossec-hids/cJSON.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	828	850
Object	input	output

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

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

Buffer Overflow IndexFromInput

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

Categories

OWASP Top 10 2017: A1-Injection

Description

Buffer Overflow IndexFromInput\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=210
Status	New

The size of the buffer used by decomp in PostfixExpr, at line 282 of ossec-hids/blast.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to stdin, at line 446 of ossec-hids/blast.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/blast.c	ossec-hids/blast.c
Line	453	371
Object	stdin	PostfixExpr

Code Snippet
File Name ossec-hids/blast.c
Method int main(void)

```
....
453.      ret = blast(inf, stdin, outf, stdout, &left, NULL);
```

File Name ossec-hids/blast.c

Method local int decomp(struct state *s)

```
....
371.              s->out[s->next++] = symbol;
```

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=1050059&projectid=50049&pathid=93
Status	New

The dangerous function, memcpy, was found in use at line 150 in ossec-hids/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	166	166
Object	memcpy	memcpy

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

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

Dangerous Functions\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=94
Status	New

The dangerous function, memcpy, was found in use at line 374 in ossec-hids/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	451	451
Object	memcpy	memcpy

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

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

Dangerous Functions\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=95
Status	New

The dangerous function, memcpy, was found in use at line 828 in ossec-hids/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	891	891
Object	memcpy	memcpy

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

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

Dangerous Functions\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=96
Status	New

The dangerous function, memcpy, was found in use at line 1088 in ossec-hids/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	1129	1129
Object	memcpy	memcpy

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

```
....
1129.          memcpy(printed, buffer->buffer, cjson_min(buffer->length,
buffer->offset + 1));
```

Dangerous Functions\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=97
Status	New

The dangerous function, memcpy, was found in use at line 1268 in ossec-hids/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	1327	1327
Object	memcpy	memcpy

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

```
....
1327.              memcpy(output, item->valuestring, raw_length);
```

Dangerous Functions\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=98
Status	New

The dangerous function, memcpy, was found in use at line 1828 in ossec-hids/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	1842	1842
Object	memcpy	memcpy

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

```
....
1842.       memcpy(reference, item, sizeof(cJSON));
```

Dangerous Functions\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=99
Status	New

The dangerous function, memcpy, was found in use at line 156 in ossec-hids/imsg.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/imsg.c	ossec-hids/imsg.c
Line	165	165
Object	memcpy	memcpy

Code Snippet
File Name ossec-hids/imsg.c
Method imsg_get(struct imsgbuf *ibuf, struct imsg *imsg)

```
....
165.        memcpy(&imsg->hdr, ibuf->r.buf, sizeof(imsg->hdr));
```

Dangerous Functions\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=100
Status	New

The dangerous function, memcpy, was found in use at line 156 in ossec-hids/imsg.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/imsg.c	ossec-hids/imsg.c
Line	183	183
Object	memcpy	memcpy

Code Snippet
File Name ossec-hids/imsg.c
Method imsg_get(struct imsgbuf *ibuf, struct imsg *imsg)

```
....
183.        memcpy(imsg->data, ibuf->r.rptr, datalen);
```

Dangerous Functions\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=101
Status	New

The dangerous function, memcpy, was found in use at line 252 in ossec-hids/lobject.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/lobject.c	ossec-hids/lobject.c
Line	256	256
Object	memcpy	memcpy

Code Snippet
File Name ossec-hids/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
256.        memcpy(out, source + 1, l * sizeof(char));
```

Dangerous Functions\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=102
Status	New

The dangerous function, memcpy, was found in use at line 252 in ossec-hids/lobject.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/lobject.c	ossec-hids/lobject.c
Line	264	264
Object	memcpy	memcpy

Code Snippet
File Name ossec-hids/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
264.        memcpy(out, source + 1, l * sizeof(char));
```

Dangerous Functions\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=103
Status	New

The dangerous function, memcpy, was found in use at line 252 in ossec-hids/lobject.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/lobject.c	ossec-hids/lobject.c
Line	268	268
Object	memcpy	memcpy

Code Snippet
File Name ossec-hids/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
268.        memcpy(out, source + 1 + l - bufflen, bufflen *
sizeof(char));
```

Dangerous Functions\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=104
Status	New

The dangerous function, memcpy, was found in use at line 252 in ossec-hids/lobject.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/lobject.c	ossec-hids/lobject.c
Line	284	284
Object	memcpy	memcpy

Code Snippet
File Name ossec-hids/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
284.      memcpy(out, POS, (LL(POS) + 1) * sizeof(char));
```

Dangerous Functions\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=105
Status	New

The dangerous function, memcpy, was found in use at line 293 in ossec-hids/lvm.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/lvm.c	ossec-hids/lvm.c
Line	324	324
Object	memcpy	memcpy

Code Snippet
File Name ossec-hids/lvm.c
Method void luaV_concat (lua_State *L, int total) {

```
....
324.          memcpy(buffer+tl, svalue(top-i), l * sizeof(char));
```

Dangerous Functions\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=106
Status	New

The dangerous function, memcpy, was found in use at line 241 in ossec-hids/validate_op.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	335	335
Object	memcpy	memcpy

Code Snippet
File Name ossec-hids/validate_op.c
Method int OS_IsValidIP(const char *in_address, os_ip *final_ip)

```
....
335.          memcpy(&(final_ip->ss), result->ai_addr, result-
>ai_addrlen);
```

Dangerous Functions\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=107
Status	New

The dangerous function, sprintf, was found in use at line 89 in ossec-hids/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	92	92
Object	sprintf	sprintf

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

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

Dangerous Functions\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=108
Status	New

The dangerous function, sprintf, was found in use at line 475 in ossec-hids/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	493	493
Object	sprintf	sprintf

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

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

Dangerous Functions\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=109
Status	New

The dangerous function, sprintf, was found in use at line 475 in ossec-hids/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	498	498
Object	sprintf	sprintf

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

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

Dangerous Functions\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=110
Status	New

The dangerous function, sprintf, was found in use at line 475 in ossec-hids/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	504	504
Object	sprintf	sprintf

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

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

Dangerous Functions\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=111
Status	New

The dangerous function, sprintf, was found in use at line 828 in ossec-hids/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	937	937
Object	sprintf	sprintf

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

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

Dangerous Functions\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=112
Status	New

The dangerous function, sscanf, was found in use at line 475 in ossec-hids/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	501	501
Object	sscanf	sscanf

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

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

Dangerous Functions\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=113
Status	New

The dangerous function, strcpy, was found in use at line 828 in ossec-hids/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	850	850
Object	strcpy	strcpy

Code Snippet

File Name ossec-hids/cJSON.c

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

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

Dangerous Functions\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=114
Status	New

The dangerous function, strcpy, was found in use at line 1268 in ossec-hids/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	1285	1285
Object	strcpy	strcpy

Code Snippet

File Name ossec-hids/cJSON.c

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

```
....
1285.               strcpy((char*)output, "null");
```

Dangerous Functions\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=115
Status	New

The dangerous function, strcpy, was found in use at line 1268 in ossec-hids/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	1294	1294
Object	strcpy	strcpy

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

```
....
1294.               strcpy((char*)output, "false");
```

Dangerous Functions\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=116
Status	New

The dangerous function, strcpy, was found in use at line 1268 in ossec-hids/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	1303	1303
Object	strcpy	strcpy

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

```
....
1303.               strcpy((char*)output, "true");
```

Dangerous Functions\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=117
Status	New

The dangerous function, strlen, was found in use at line 45 in ossec-hids/b64.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/b64.c	ossec-hids/b64.c
Line	55	55
Object	strlen	strlen

Code Snippet
File Name ossec-hids/b64.c
Method char *encode_base64(int size, char *src)

```
....
55.           size = strlen((char *)src);
```

Dangerous Functions\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=118
Status	New

The dangerous function, strlen, was found in use at line 106 in ossec-hids/b64.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/b64.c	ossec-hids/b64.c
Line	111	111
Object	strlen	strlen

Code Snippet
File Name ossec-hids/b64.c
Method char *decode_base64(const char *src)

```
....
111.          int k, l = strlen(src) + 1;
```

Dangerous Functions\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=119
Status	New

The dangerous function, strlen, was found in use at line 150 in ossec-hids/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	160	160
Object	strlen	strlen

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

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

Dangerous Functions\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=120
Status	New

The dangerous function, strlen, was found in use at line 462 in ossec-hids/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	471	471
Object	strlen	strlen

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

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

Dangerous Functions\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=121
Status	New

The dangerous function, strlen, was found in use at line 1001 in ossec-hids/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	1016	1016
Object	strlen	strlen

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

```
....
1016.      buffer.length = strlen((const char*)value) + sizeof("");
```

Dangerous Functions\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=122
Status	New

The dangerous function, strlen, was found in use at line 1268 in ossec-hids/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	1321	1321
Object	strlen	strlen

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

```
....
1321.              raw_length = strlen(item->valuestring) + sizeof("");
```

Dangerous Functions\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=123
Status	New

The dangerous function, strlen, was found in use at line 252 in ossec-hids/lobject.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/lobject.c	ossec-hids/lobject.c
Line	253	253
Object	strlen	strlen

Code Snippet
File Name ossec-hids/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
253.    size_t l = strlen(source);
```

Dangerous Functions\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=124
Status	New

The dangerous function, strlen, was found in use at line 179 in ossec-hids/lobject.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/lobject.c	ossec-hids/lobject.c
Line	190	190
Object	strlen	strlen

Code Snippet
File Name ossec-hids/lobject.c
Method const char *luaO_pushvfstring (lua_State *L, const char *fmt, va_list argp) {

```
....
190.          pushstr(L, s, strlen(s));
```

Dangerous Functions\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=125
Status	New

The dangerous function, strlen, was found in use at line 179 in ossec-hids/lobject.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/lobject.c	ossec-hids/lobject.c
Line	227	227
Object	strlen	strlen

Code Snippet
File Name ossec-hids/lobject.c
Method const char *luaO_pushvfstring (lua_State *L, const char *fmt, va_list argp) {

```
....
227.    pushstr(L, fmt, strlen(fmt));
```

Dangerous Functions\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=126
Status	New

The dangerous function, strlen, was found in use at line 209 in ossec-hids/lvm.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/lvm.c	ossec-hids/lvm.c
Line	218	218
Object	strlen	strlen

Code Snippet
File Name ossec-hids/lvm.c
Method static int l_strcmp (const TString *ls, const TString *rs) {

```
....
218.        size_t len = strlen(l);  /* index of first `\0' in both
strings */
```

Dangerous Functions\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=127
Status	New

The dangerous function, strlen, was found in use at line 57 in ossec-hids/run_rk_check.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/run_rk_check.c	ossec-hids/run_rk_check.c
Line	70	70
Object	strlen	strlen

Code Snippet
File Name ossec-hids/run_rk_check.c
Method void run_rk_check()

```
....
70.      i = strlen(basedir);
```

Dangerous Functions\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=128
Status	New

The dangerous function, strlen, was found in use at line 674 in ossec-hids/validate_op.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	702	702
Object	strlen	strlen

Code Snippet
File Name ossec-hids/validate_op.c
Method char *OS_IsValidDay(const char *day_str)

```
....
702.              if (strncasecmp(day_str, days[i], strlen(days[i])) ==
0) {
```

Dangerous Functions\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=129
Status	New

The dangerous function, strlen, was found in use at line 674 in ossec-hids/validate_op.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	728	728
Object	strlen	strlen

Code Snippet
File Name ossec-hids/validate_op.c
Method char *OS_IsValidDay(const char *day_str)

```
....
728.          day_str += strlen(days[i]);
```

Dangerous Functions\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=130
Status	New

The dangerous function, strlen, was found in use at line 241 in ossec-hids/validate_op.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	288	288
Object	strlen	strlen

Code Snippet
File Name ossec-hids/validate_op.c
Method int OS_IsValidIP(const char *in_address, os_ip *final_ip)

```
....
288.          if(strlen(tmp_str) <= 3) {
```

Dangerous Functions\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=131
Status	New

The dangerous function, atoi, was found in use at line 26 in ossec-hids/active-response.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/active-response.c	ossec-hids/active-response.c
Line	131	131
Object	atoi	atoi

Code Snippet
File Name ossec-hids/active-response.c
Method int ReadActiveResponses(XML_NODE node, void *d1, void *d2)

```
....
131.              tmp_ar->level = atoi(node[i]->content);
```

Dangerous Functions\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=132
Status	New

The dangerous function, atoi, was found in use at line 26 in ossec-hids/active-response.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/active-response.c	ossec-hids/active-response.c
Line	139	139
Object	atoi	atoi

Code Snippet
File Name ossec-hids/active-response.c
Method int ReadActiveResponses(XML_NODE node, void *d1, void *d2)

```
....
139.              tmp_ar->timeout = atoi(node[i]->content);
```

Dangerous Functions\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=133
Status	New

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

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	140	140
Object	realloc	realloc

Code Snippet
File Name ossec-hids/cJSON.c
Method static void *internal_realloc(void *pointer, size_t size)

```
....
140.      return realloc(pointer, size);
```

Dangerous Functions\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=134
Status	New

The dangerous function, atoi, was found in use at line 114 in ossec-hids/validate_op.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	137	137
Object	atoi	atoi

Code Snippet
File Name ossec-hids/validate_op.c
Method int getDefine_Int(const char *high_name, const char *low_name, int min, int max)

```
....
137.      ret = atoi(value);
```

Dangerous Functions\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=135
Status	New

The dangerous function, atoi, was found in use at line 241 in ossec-hids/validate_op.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	289	289
Object	atoi	atoi

Code Snippet
File Name ossec-hids/validate_op.c
Method int OS_IsValidIP(const char *in_address, os_ip *final_ip)

```
....
289.              cidr = atoi(tmp_str);
```

Dangerous Functions\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=136
Status	New

The dangerous function, atoi, was found in use at line 454 in ossec-hids/validate_op.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	466	466
Object	atoi	atoi

Code Snippet
File Name ossec-hids/validate_op.c
Method static const char *__gethour(const char *str, char *ossec_hour)

```
....
466.      chour = atoi(str);
```

Dangerous Functions\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=137
Status	New

The dangerous function, atoi, was found in use at line 454 in ossec-hids/validate_op.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	495	495
Object	atoi	atoi

Code Snippet
File Name ossec-hids/validate_op.c
Method static const char *__gethour(const char *str, char *ossec_hour)

```
....
495.          cmin = atoi(str);
```

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=1050059&projectid=50049&pathid=146
Status	New

The variable declared in after_end at ossec-hids/cJSON.c in line 267 is not initialized when it is used by after_end at ossec-hids/cJSON.c in line 267.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	270	339

Object	after_end	after_end

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

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

Use of Zero Initialized Pointer\Path 2:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=147
Status New

The variable declared in current_item at ossec-hids/cJSON.c in line 1346 is not initialized when it is used by prev at ossec-hids/cJSON.c in line 1346.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	1349	1400
Object	current_item	prev

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

```
....
1349.      cJSON *current_item = NULL;
....
1400.              new_item->prev = current_item;
```

Use of Zero Initialized Pointer\Path 3:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=148
Status New

The variable declared in current_item at ossec-hids/cJSON.c in line 1502 is not initialized when it is used by prev at ossec-hids/cJSON.c in line 1502.

	Source	Destination

File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	1505	1554
Object	current_item	prev

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

```
....
1505.       cJSON *current_item = NULL;
....
1554.               new_item->prev = current_item;
```

Use of Zero Initialized Pointer\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=149
Status	New

The variable declared in p at ossec-hids/cJSON.c in line 2477 is not initialized when it is used by prev at ossec-hids/cJSON.c in line 1821.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2481	1824
Object	p	prev

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

```
....
2481.       cJSON *p = NULL;
```

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

```
....
1824.       item->prev = prev;
```

Use of Zero Initialized Pointer\Path 5:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=150
Status	New

The variable declared in p at ossec-hids/cJSON.c in line 2406 is not initialized when it is used by prev at ossec-hids/cJSON.c in line 1821.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2410	1824
Object	p	prev

Code Snippet

File Name ossec-hids/cJSON.c

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

```
....
2410.       cJSON *p = NULL;
```

File Name ossec-hids/cJSON.c

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

```
....
1824.       item->prev = prev;
```

Use of Zero Initialized Pointer\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=151
Status	New

The variable declared in p at ossec-hids/cJSON.c in line 2441 is not initialized when it is used by prev at ossec-hids/cJSON.c in line 1821.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2445	1824
Object	p	prev

Code Snippet

File Name ossec-hids/cJSON.c

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

```
....
2445.       cJSON *p = NULL;
```

File Name ossec-hids/cJSON.c

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

```
....
1824.       item->prev = prev;
```

Use of Zero Initialized Pointer\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=152
Status	New

The variable declared in p at ossec-hids/cJSON.c in line 2513 is not initialized when it is used by prev at ossec-hids/cJSON.c in line 1821.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2517	1824
Object	p	prev

Code Snippet

File Name ossec-hids/cJSON.c

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

```
....
2517.       cJSON *p = NULL;
```

File Name ossec-hids/cJSON.c

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

```
....
1824.       item->prev = prev;
```

Use of Zero Initialized Pointer\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=153
Status	New

The variable declared in p at ossec-hids/cJSON.c in line 2406 is not initialized when it is used by p at ossec-hids/cJSON.c in line 2406.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2410	2435
Object	p	p

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

```
....
2410.       cJSON *p = NULL;
....
2435.           p = n;
```

Use of Zero Initialized Pointer\Path 9:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=154
Status New

The variable declared in p at ossec-hids/cJSON.c in line 2477 is not initialized when it is used by next at ossec-hids/cJSON.c in line 1821.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2481	1823
Object	p	next

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

```
....
2481.       cJSON *p = NULL;
```

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

```
....
1823.       prev->next = item;
```

Use of Zero Initialized Pointer\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=155
Status	New

The variable declared in p at ossec-hids/cJSON.c in line 2441 is not initialized when it is used by next at ossec-hids/cJSON.c in line 1821.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2445	1823
Object	p	next

Code Snippet

File Name ossec-hids/cJSON.c

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

```
....
2445.       cJSON *p = NULL;
```

File Name ossec-hids/cJSON.c

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

```
....
1823.       prev->next = item;
```

Use of Zero Initialized Pointer\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=156
Status	New

The variable declared in p at ossec-hids/cJSON.c in line 2406 is not initialized when it is used by next at ossec-hids/cJSON.c in line 1821.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2410	1823
Object	p	next

Code Snippet

File Name	ossec-hids/cJSON.c
Method	CJSON_PUBLIC(cJSON *) cJSON_CreateIntArray(const int *numbers, int count)

```
....
2410.       cJSON *p = NULL;
```

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

```
....
1823.       prev->next = item;
```

Use of Zero Initialized Pointer\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=157
Status	New

The variable declared in p at ossec-hids/cJSON.c in line 2513 is not initialized when it is used by next at ossec-hids/cJSON.c in line 1821.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2517	1823
Object	p	next

Code Snippet

File Name	ossec-hids/cJSON.c
Method	CJSON_PUBLIC(cJSON *) cJSON_CreateStringArray(const char **strings, int count)

```
....
2517.       cJSON *p = NULL;
```

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

```
....
1823.       prev->next = item;
```

Use of Zero Initialized Pointer\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=500

	49&pathid=158
Status	New

The variable declared in p at ossec-hids/cJSON.c in line 2441 is not initialized when it is used by p at ossec-hids/cJSON.c in line 2441.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2445	2471
Object	p	p

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

```
....
2445.       cJSON *p = NULL;
....
2471.           p = n;
```

Use of Zero Initialized Pointer\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=159
Status	New

The variable declared in p at ossec-hids/cJSON.c in line 2477 is not initialized when it is used by p at ossec-hids/cJSON.c in line 2477.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2481	2507
Object	p	p

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

```
....
2481.       cJSON *p = NULL;
....
2507.           p = n;
```

Use of Zero Initialized Pointer\Path 15:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=160
Status	New

The variable declared in p at ossec-hids/cJSON.c in line 2513 is not initialized when it is used by p at ossec-hids/cJSON.c in line 2513.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2517	2543
Object	p	p

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

```
....
2517.       cJSON *p = NULL;
....
2543.           p = n;
```

Use of Zero Initialized Pointer\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=161
Status	New

The variable declared in a_element at ossec-hids/cJSON.c in line 2799 is not initialized when it is used by a_element at ossec-hids/cJSON.c in line 2799.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2883	2888
Object	a_element	a_element

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

```
....
2883.               cJSON *a_element = NULL;
....
2888.                   b_element = get_object_item(b, a_element->string,
case_sensitive);
```

Use of Zero Initialized Pointer\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=162
Status	New

The variable declared in ar_cmd at ossec-hids/active-response.c in line 26 is not initialized when it is used by ar_cmd at ossec-hids/active-response.c in line 26.

	Source	Destination
File	ossec-hids/active-response.c	ossec-hids/active-response.c
Line	97	283
Object	ar_cmd	ar_cmd

Code Snippet
File Name ossec-hids/active-response.c
Method int ReadActiveResponses(XML_NODE node, void *d1, void *d2)

```
....
97.      tmp_ar->ar_cmd = NULL;
....
283.              tmp_ar->ar_cmd->executable,
```

Use of Zero Initialized Pointer\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=163
Status	New

The variable declared in ar_cmd at ossec-hids/active-response.c in line 26 is not initialized when it is used by ar_cmd at ossec-hids/active-response.c in line 26.

	Source	Destination
File	ossec-hids/active-response.c	ossec-hids/active-response.c
Line	97	276
Object	ar_cmd	ar_cmd

Code Snippet
File Name ossec-hids/active-response.c
Method int ReadActiveResponses(XML_NODE node, void *d1, void *d2)

```
....
97.      tmp_ar->ar_cmd = NULL;
....
276.              tmp_ar->ar_cmd->name,
```

Use of Zero Initialized Pointer\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=164
Status	New

The variable declared in buffer at ossec-hids/cJSON.c in line 374 is not initialized when it is used by newbuffer at ossec-hids/cJSON.c in line 374.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	445	427
Object	buffer	newbuffer

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

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

Use of Zero Initialized Pointer\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=165
Status	New

The variable declared in buffer at ossec-hids/cJSON.c in line 374 is not initialized when it is used by newbuffer at ossec-hids/cJSON.c in line 374.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	432	427
Object	buffer	newbuffer

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

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

Use of Zero Initialized Pointer\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=166
Status	New

The variable declared in buffer at ossec-hids/cJSON.c in line 374 is not initialized when it is used by output_pointer at ossec-hids/cJSON.c in line 1611.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	432	1643
Object	buffer	output_pointer

Code Snippet

File Name ossec-hids/cJSON.c

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

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

File Name ossec-hids/cJSON.c

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

```
....
1643.              output_pointer = ensure(output_buffer, output_buffer-
>depth);
```

Use of Zero Initialized Pointer\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=167
Status	New

The variable declared in buffer at ossec-hids/cJSON.c in line 374 is not initialized when it is used by output_pointer at ossec-hids/cJSON.c in line 1611.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	445	1643
Object	buffer	output_pointer

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

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

File Name ossec-hids/cJSON.c

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

```
....
1643.                output_pointer = ensure(output_buffer, output_buffer->depth);
```

Use of Zero Initialized Pointer\Path 23:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=168
Status New

The variable declared in buffer at ossec-hids/cJSON.c in line 374 is not initialized when it is used by output at ossec-hids/cJSON.c in line 828.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	432	881
Object	buffer	output

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

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

File Name ossec-hids/cJSON.c

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

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

Use of Zero Initialized Pointer\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=169
Status	New

The variable declared in buffer at ossec-hids/cJSON.c in line 374 is not initialized when it is used by newbuffer at ossec-hids/cJSON.c in line 374.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	432	440
Object	buffer	newbuffer

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

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

Use of Zero Initialized Pointer\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=170
Status	New

The variable declared in buffer at ossec-hids/cJSON.c in line 374 is not initialized when it is used by newbuffer at ossec-hids/cJSON.c in line 374.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	445	440
Object	buffer	newbuffer

Code Snippet

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

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

Use of Zero Initialized Pointer\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=171
Status	New

The variable declared in buffer at ossec-hids/cJSON.c in line 374 is not initialized when it is used by output_pointer at ossec-hids/cJSON.c in line 1611.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	445	1704
Object	buffer	output_pointer

Code Snippet

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

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

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

```
....
1704.       output_pointer = ensure(output_buffer, output_buffer->format
? (output_buffer->depth + 1) : 2);
```

Use of Zero Initialized Pointer\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=172
Status	New

The variable declared in buffer at ossec-hids/cJSON.c in line 374 is not initialized when it is used by output_pointer at ossec-hids/cJSON.c in line 1611.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	432	1704
Object	buffer	output_pointer

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

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

File Name ossec-hids/cJSON.c

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

```
....
1704.       output_pointer = ensure(output_buffer, output_buffer->format
? (output_buffer->depth + 1) : 2);
```

Use of Zero Initialized Pointer\Path 28:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=173
Status New

The variable declared in buffer at ossec-hids/cJSON.c in line 374 is not initialized when it is used by output_pointer at ossec-hids/cJSON.c in line 1611.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	432	1684
Object	buffer	output_pointer

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

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

File Name ossec-hids/cJSON.c

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

```
....
1684.          output_pointer = ensure(output_buffer, length + 1);
```

Use of Zero Initialized Pointer\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=174
Status	New

The variable declared in buffer at ossec-hids/cJSON.c in line 374 is not initialized when it is used by output_pointer at ossec-hids/cJSON.c in line 1611.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	445	1684
Object	buffer	output_pointer

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

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

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

```
....
1684.          output_pointer = ensure(output_buffer, length + 1);
```

Use of Zero Initialized Pointer\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=175
Status	New

The variable declared in buffer at ossec-hids/cJSON.c in line 374 is not initialized when it is used by buffer at ossec-hids/cJSON.c in line 462.

	Source	Destination

File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	432	469
Object	buffer	buffer

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

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

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

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

Use of Zero Initialized Pointer\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=176
Status	New

The variable declared in buffer at ossec-hids/cJSON.c in line 374 is not initialized when it is used by buffer at ossec-hids/cJSON.c in line 462.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	445	469
Object	buffer	buffer

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

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

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

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

Use of Zero Initialized Pointer\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=177
Status	New

The variable declared in buffer at ossec-hids/cJSON.c in line 374 is not initialized when it is used by output_pointer at ossec-hids/cJSON.c in line 1611.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	445	1663
Object	buffer	output_pointer

Code Snippet

File Name ossec-hids/cJSON.c

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

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

File Name ossec-hids/cJSON.c

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

```
....
1663.           output_pointer = ensure(output_buffer, length);
```

Use of Zero Initialized Pointer\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=178
Status	New

The variable declared in buffer at ossec-hids/cJSON.c in line 374 is not initialized when it is used by output_pointer at ossec-hids/cJSON.c in line 1611.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	432	1663
Object	buffer	output_pointer

Code Snippet

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

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

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

```
....
1663.           output_pointer = ensure(output_buffer, length);
```

Use of Zero Initialized Pointer\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=179
Status	New

The variable declared in buffer at ossec-hids/cJSON.c in line 374 is not initialized when it is used by output_pointer at ossec-hids/cJSON.c in line 1440.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	445	1489
Object	buffer	output_pointer

Code Snippet

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

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

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

```
....
1489.       output_pointer = ensure(output_buffer, 2);
```

Use of Zero Initialized Pointer\Path 35:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=180
Status	New

The variable declared in buffer at ossec-hids/cJSON.c in line 374 is not initialized when it is used by output_pointer at ossec-hids/cJSON.c in line 1440.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	432	1489
Object	buffer	output_pointer

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

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

File Name ossec-hids/cJSON.c

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

```
....
1489.      output_pointer = ensure(output_buffer, 2);
```

Use of Zero Initialized Pointer\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=181
Status	New

The variable declared in buffer at ossec-hids/cJSON.c in line 374 is not initialized when it is used by output_pointer at ossec-hids/cJSON.c in line 1440.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	432	1473
Object	buffer	output_pointer

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

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

File Name ossec-hids/cJSON.c

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

```
....
1473.              output_pointer = ensure(output_buffer, length + 1);
```

Use of Zero Initialized Pointer\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=182
Status	New

The variable declared in buffer at ossec-hids/cJSON.c in line 374 is not initialized when it is used by output_pointer at ossec-hids/cJSON.c in line 1440.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	445	1473
Object	buffer	output_pointer

Code Snippet

File Name ossec-hids/cJSON.c

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

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

File Name ossec-hids/cJSON.c

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

```
....
1473.              output_pointer = ensure(output_buffer, length + 1);
```

Use of Zero Initialized Pointer\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=183

Status	New

The variable declared in valuestring at ossec-hids/cJSON.c in line 1502 is not initialized when it is used by current_item at ossec-hids/cJSON.c in line 1502.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	1569	1568
Object	valuestring	current_item

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

```
....
1569.           current_item->valuestring = NULL;
....
1568.           current_item->string = current_item->valuestring;
```

Use of Zero Initialized Pointer\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=184
Status	New

The variable declared in prev at ossec-hids/cJSON.c in line 2170 is not initialized when it is used by next at ossec-hids/cJSON.c in line 215.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2199	220
Object	prev	next

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

```
....
2199.       item->prev = NULL;
```

File Name ossec-hids/cJSON.c
Method CJSON_PUBLIC(void) cJSON_Delete(cJSON *item)

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

Use of Zero Initialized Pointer\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=185
Status	New

The variable declared in next at ossec-hids/cJSON.c in line 2071 is not initialized when it is used by next at ossec-hids/cJSON.c in line 215.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2096	220
Object	next	next

Code Snippet

File Name ossec-hids/cJSON.c

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

```
....
2096.      item->next = NULL;
```

File Name ossec-hids/cJSON.c

Method CJSON_PUBLIC(void) cJSON_Delete(cJSON *item)

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

Use of Zero Initialized Pointer\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=186
Status	New

The variable declared in prev at ossec-hids/cJSON.c in line 2071 is not initialized when it is used by next at ossec-hids/cJSON.c in line 215.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c

Line	2095	220
Object	prev	next

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

```
....
2095.       item->prev = NULL;
```

File Name ossec-hids/cJSON.c
Method CJSON_PUBLIC(void) cJSON_Delete(cJSON *item)

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

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=1050059&projectid=50049&pathid=39
Status	New

The size of the buffer used by *create_reference in cJSON, at line 1828 of ossec-hids/cJSON.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *create_reference passes to cJSON, at line 1828 of ossec-hids/cJSON.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	1842	1842
Object	cJSON	cJSON

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

```
....
1842.       memcpy(reference, item, sizeof(cJSON));
```

Buffer Overflow boundcpy WrongSizeParam\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=40
Status	New

The size of the buffer used by imsg_get in ->, at line 156 of ossec-hids/imsg.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that imsg_get passes to ->, at line 156 of ossec-hids/imsg.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/imsg.c	ossec-hids/imsg.c
Line	165	165
Object	->	->

Code Snippet
File Name ossec-hids/imsg.c
Method imsg_get(struct imsgbuf *ibuf, struct imsg *imsg)

```
....
165.        memcpy(&imsg->hdr, ibuf->r.buf, sizeof(imsg->hdr));
```

Buffer Overflow boundcpy WrongSizeParam\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=41
Status	New

The size of the buffer used by *cJSON_New_Item in cJSON, at line 203 of ossec-hids/cJSON.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *cJSON_New_Item passes to cJSON, at line 203 of ossec-hids/cJSON.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	208	208
Object	cJSON	cJSON

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

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

Buffer Overflow boundcpy WrongSizeParam\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=42
Status	New

The size of the buffer used by OS_IPFound in _os_ip, at line 151 of ossec-hids/validate_op.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that OS_IPFound passes to _os_ip, at line 151 of ossec-hids/validate_op.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	170	170
Object	_os_ip	_os_ip

Code Snippet
File Name ossec-hids/validate_op.c
Method int OS_IPFound(const char *ip_address, const os_ip *that_ip)

```
....
170.      memset(&temp_ip, 0, sizeof(struct _os_ip));
```

Buffer Overflow boundcpy WrongSizeParam\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=43
Status	New

The size of the buffer used by OS_IPFoundList in _os_ip, at line 193 of ossec-hids/validate_op.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that OS_IPFoundList passes to _os_ip, at line 193 of ossec-hids/validate_op.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	198	198
Object	_os_ip	_os_ip

Code Snippet
File Name ossec-hids/validate_op.c
Method int OS_IPFoundList(const char *ip_address, os_ip **list_of_ips)

```
....
198.      memset(&temp_ip, 0, sizeof(struct _os_ip));
```

Buffer Overflow boundcpy WrongSizeParam\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=44
Status	New

The size of the buffer used by OS_IsValidIP in addrinfo, at line 241 of ossec-hids/validate_op.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that OS_IsValidIP passes to addrinfo, at line 241 of ossec-hids/validate_op.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	297	297
Object	addrinfo	addrinfo

Code Snippet
File Name ossec-hids/validate_op.c
Method int OS_IsValidIP(const char *in_address, os_ip *final_ip)

```
....
297.      memset(&hints, 0, sizeof(struct addrinfo));
```

Buffer Overflow boundcpy WrongSizeParam\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=45
Status	New

The size of the buffer used by luaO_chunkid in l, at line 252 of ossec-hids/lobject.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that luaO_chunkid passes to l, at line 252 of ossec-hids/lobject.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/lobject.c	ossec-hids/lobject.c
Line	256	256
Object	l	l

Code Snippet
File Name ossec-hids/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
256.        memcpy(out, source + 1, l * sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=46
Status	New

The size of the buffer used by luaO_chunkid in char, at line 252 of ossec-hids/lobject.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that luaO_chunkid passes to char, at line 252 of ossec-hids/lobject.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/lobject.c	ossec-hids/lobject.c
Line	256	256
Object	char	char

Code Snippet
File Name ossec-hids/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
256.        memcpy(out, source + 1, l * sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=47
Status	New

The size of the buffer used by luaO_chunkid in l, at line 252 of ossec-hids/lobject.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that luaO_chunkid passes to l, at line 252 of ossec-hids/lobject.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/lobject.c	ossec-hids/lobject.c
Line	264	264
Object	l	l

Code Snippet
File Name ossec-hids/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
264.        memcpy(out, source + 1, l * sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=48
Status	New

The size of the buffer used by luaO_chunkid in char, at line 252 of ossec-hids/lobject.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that luaO_chunkid passes to char, at line 252 of ossec-hids/lobject.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/lobject.c	ossec-hids/lobject.c
Line	264	264
Object	char	char

Code Snippet
File Name ossec-hids/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
264.        memcpy(out, source + 1, l * sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=49
Status	New

The size of the buffer used by luaO_chunkid in bufflen, at line 252 of ossec-hids/lobject.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that luaO_chunkid passes to bufflen, at line 252 of ossec-hids/lobject.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/lobject.c	ossec-hids/lobject.c
Line	268	268
Object	bufflen	bufflen

Code Snippet
File Name ossec-hids/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
268.        memcpy(out, source + 1 + l - bufflen, bufflen *
sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=50
Status	New

The size of the buffer used by luaO_chunkid in char, at line 252 of ossec-hids/lobject.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that luaO_chunkid passes to char, at line 252 of ossec-hids/lobject.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/lobject.c	ossec-hids/lobject.c
Line	268	268
Object	char	char

Code Snippet
File Name ossec-hids/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
268.        memcpy(out, source + 1 + l - bufflen, bufflen *
sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=51
Status	New

The size of the buffer used by luaO_chunkid in char, at line 252 of ossec-hids/lobject.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that luaO_chunkid passes to char, at line 252 of ossec-hids/lobject.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/lobject.c	ossec-hids/lobject.c
Line	284	284
Object	char	char

Code Snippet
File Name ossec-hids/lobject.c
Method void luaO_chunkid (char *out, const char *source, size_t bufflen) {

```
....
284.      memcpy(out, POS, (LL(POS) + 1) * sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=52
Status	New

The size of the buffer used by luaV_concat in l, at line 293 of ossec-hids/lvm.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that luaV_concat passes to l, at line 293 of ossec-hids/lvm.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/lvm.c	ossec-hids/lvm.c
Line	324	324
Object	l	l

Code Snippet
File Name ossec-hids/lvm.c
Method void luaV_concat (lua_State *L, int total) {

```
....
324.          memcpy(buffer+tl, svalue(top-i), l * sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=53
Status	New

The size of the buffer used by luaV_concat in char, at line 293 of ossec-hids/lvm.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that luaV_concat passes to char, at line 293 of ossec-hids/lvm.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/lvm.c	ossec-hids/lvm.c
Line	324	324
Object	char	char

Code Snippet
File Name ossec-hids/lvm.c
Method void luaV_concat (lua_State *L, int total) {

```
....
324.          memcpy(buffer+tl, svalue(top-i), l * sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=54
Status	New

The size of the buffer used by cJSON_strdup in length, at line 150 of ossec-hids/cJSON.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that cJSON_strdup passes to length, at line 150 of ossec-hids/cJSON.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	166	166
Object	length	length

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

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

Buffer Overflow boundcpy WrongSizeParam\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=55
Status	New

The size of the buffer used by print_string_ptr in output_length, at line 828 of ossec-hids/cJSON.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that print_string_ptr passes to output_length, at line 828 of ossec-hids/cJSON.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	891	891
Object	output_length	output_length

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

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

Buffer Overflow boundcpy WrongSizeParam\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=56
Status	New

The size of the buffer used by print_value in raw_length, at line 1268 of ossec-hids/cJSON.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that print_value passes to raw_length, at line 1268 of ossec-hids/cJSON.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	1327	1327
Object	raw_length	raw_length

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

```
....
1327.              memcpy(output, item->valuestring, raw_length);
```

Buffer Overflow boundcpy WrongSizeParam\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=57
Status	New

The size of the buffer used by imsg_get in datalen, at line 156 of ossec-hids/imsg.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that imsg_get passes to datalen, at line 156 of ossec-hids/imsg.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/imsg.c	ossec-hids/imsg.c
Line	183	183
Object	datalen	datalen

Code Snippet
File Name ossec-hids/imsg.c
Method imsg_get(struct imsgbuf *ibuf, struct imsg *imsg)

```
....
183.        memcpy(imsg->data, ibuf->r.rptr, datalen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=58
Status	New

The size of the buffer used by OS_IsValidIP in result, at line 241 of ossec-hids/validate_op.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that OS_IsValidIP passes to result, at line 241 of ossec-hids/validate_op.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	335	335
Object	result	result

Code Snippet
File Name ossec-hids/validate_op.c
Method int OS_IsValidIP(const char *in_address, os_ip *final_ip)

```
....
335.          memcpy(&(final_ip->ss), result->ai_addr, result-
>ai_addrlen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=59
Status	New

The size of the buffer used by imsg_get in left, at line 156 of ossec-hids/imsg.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that imsg_get passes to left, at line 156 of ossec-hids/imsg.c, to overwrite the target buffer.

	Source	Destination
File	ossec-hids/imsg.c	ossec-hids/imsg.c
Line	187	187
Object	left	left

Code Snippet
File Name ossec-hids/imsg.c
Method imsg_get(struct imsgbuf *ibuf, struct imsg *imsg)

```
....
187.                memmove(&ibuf->r.buf, ibuf->r.buf + imsg->hdr.len,
left);
```

MemoryFree on StackVariable

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

Description

MemoryFree on StackVariable\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=60
Status	New

Calling free() (line 300) on a variable that was not dynamically allocated (line 300) in file ossec-hids/imsg.c may result with a crash.

	Source	Destination
File	ossec-hids/imsg.c	ossec-hids/imsg.c
Line	310	310
Object	ifd	ifd

Code Snippet
File Name ossec-hids/imsg.c
Method imsg_get_fd(struct imsgbuf *ibuf)

```
....
310.        free(ifd);
```

MemoryFree on StackVariable\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=61
Status	New

Calling free() (line 674) on a variable that was not dynamically allocated (line 674) in file ossec-hids/validate_op.c may result with a crash.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	759	759
Object	ret	ret

Code Snippet

File Name	ossec-hids/validate_op.c
Method	char *OS_IsValidDay(const char *day_str)

```
....
759.            free(ret);
```

MemoryFree on StackVariable\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=62
Status	New

Calling free() (line 114) on a variable that was not dynamically allocated (line 114) in file ossec-hids/validate_op.c may result with a crash.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	143	143
Object	value	value

Code Snippet	
File Name	ossec-hids/validate_op.c
Method	int getDefine_Int(const char *high_name, const char *low_name, int min, int max)

```
....
143.        free(value);
```

MemoryFree on StackVariable\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=63
Status	New

Calling free() (line 241) on a variable that was not dynamically allocated (line 241) in file ossec-hids/validate_op.c may result with a crash.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	266	266
Object	ip_address	ip_address

Code Snippet	
File Name	ossec-hids/validate_op.c
Method	int OS_IsValidIP(const char *in_address, os_ip *final_ip)

```
....
266.              free(ip_address);
```

MemoryFree on StackVariable\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=64
Status	New

Calling free() (line 241) on a variable that was not dynamically allocated (line 241) in file ossec-hids/validate_op.c may result with a crash.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	277	277
Object	ip_address	ip_address

Code Snippet
File Name ossec-hids/validate_op.c
Method int OS_IsValidIP(const char *in_address, os_ip *final_ip)

```
....
277.          free(ip_address);    // Free the old value before writing
the new one?
```

MemoryFree on StackVariable\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=65
Status	New

Calling free() (line 241) on a variable that was not dynamically allocated (line 241) in file ossec-hids/validate_op.c may result with a crash.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	291	291
Object	ip_address	ip_address

Code Snippet
File Name ossec-hids/validate_op.c
Method int OS_IsValidIP(const char *in_address, os_ip *final_ip)

```
....
291.              free(ip_address);
```

MemoryFree on StackVariable\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=66
Status	New

Calling free() (line 241) on a variable that was not dynamically allocated (line 241) in file ossec-hids/validate_op.c may result with a crash.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	300	300
Object	ip_address	ip_address

Code Snippet
File Name ossec-hids/validate_op.c
Method int OS_IsValidIP(const char *in_address, os_ip *final_ip)

```
....
300.          free(ip_address);
```

MemoryFree on StackVariable\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=67
Status	New

Calling free() (line 241) on a variable that was not dynamically allocated (line 241) in file ossec-hids/validate_op.c may result with a crash.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	314	314
Object	ip_address	ip_address

Code Snippet
File Name ossec-hids/validate_op.c
Method int OS_IsValidIP(const char *in_address, os_ip *final_ip)

```
....
314.          free(ip_address);
```

MemoryFree on StackVariable\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=68
Status	New

Calling free() (line 241) on a variable that was not dynamically allocated (line 241) in file ossec-hids/validate_op.c may result with a crash.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	315	315
Object	result	result

Code Snippet
File Name ossec-hids/validate_op.c
Method int OS_IsValidIP(const char *in_address, os_ip *final_ip)

```
....
315.          free(result);
```

MemoryFree on StackVariable\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=69
Status	New

Calling free() (line 241) on a variable that was not dynamically allocated (line 241) in file ossec-hids/validate_op.c may result with a crash.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	325	325
Object	ip_address	ip_address

Code Snippet
File Name ossec-hids/validate_op.c
Method int OS_IsValidIP(const char *in_address, os_ip *final_ip)

```
....
325.          free(ip_address);
```

MemoryFree on StackVariable\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=70
Status	New

Calling free() (line 241) on a variable that was not dynamically allocated (line 241) in file ossec-hids/validate_op.c may result with a crash.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	326	326
Object	result	result

Code Snippet
File Name ossec-hids/validate_op.c
Method int OS_IsValidIP(const char *in_address, os_ip *final_ip)

```
....
326.          free(result);
```

MemoryFree on StackVariable\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=71
Status	New

Calling free() (line 241) on a variable that was not dynamically allocated (line 241) in file ossec-hids/validate_op.c may result with a crash.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	329	329
Object	ip_address	ip_address

Code Snippet
File Name ossec-hids/validate_op.c
Method int OS_IsValidIP(const char *in_address, os_ip *final_ip)

```
....
329.            free(ip_address);
```

MemoryFree on StackVariable\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=72
Status	New

Calling free() (line 241) on a variable that was not dynamically allocated (line 241) in file ossec-hids/validate_op.c may result with a crash.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	330	330
Object	result	result

Code Snippet
File Name ossec-hids/validate_op.c
Method int OS_IsValidIP(const char *in_address, os_ip *final_ip)

```
....
330.            free(result);
```

MemoryFree on StackVariable\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=73
Status	New

Calling free() (line 241) on a variable that was not dynamically allocated (line 241) in file ossec-hids/validate_op.c may result with a crash.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	341	341
Object	ip_address	ip_address

Code Snippet
File Name ossec-hids/validate_op.c
Method int OS_IsValidIP(const char *in_address, os_ip *final_ip)

```
....
341.      free(ip_address);
```

Memory Leak

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

Categories

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

Description

Memory Leak\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=138
Status	New

	Source	Destination
File	ossec-hids/active-response.c	ossec-hids/active-response.c
Line	112	112
Object	command	command

Code Snippet
File Name ossec-hids/active-response.c
Method int ReadActiveResponses(XML_NODE node, void *d1, void *d2)

```
....
112.              tmp_ar->command = strdup(node[i]->content);
```

Memory Leak\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=139
Status	New

	Source	Destination
File	ossec-hids/active-response.c	ossec-hids/active-response.c
Line	119	119
Object	agent_id	agent_id

Code Snippet
File Name ossec-hids/active-response.c
Method int ReadActiveResponses(XML_NODE node, void *d1, void *d2)

```
....
119.            tmp_ar->agent_id = strdup(node[i]->content);
```

Memory Leak\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=140
Status	New

	Source	Destination
File	ossec-hids/active-response.c	ossec-hids/active-response.c
Line	121	121
Object	rules_id	rules_id

Code Snippet
File Name ossec-hids/active-response.c
Method int ReadActiveResponses(XML_NODE node, void *d1, void *d2)

```
....
121.            tmp_ar->rules_id = strdup(node[i]->content);
```

Memory Leak\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=141
Status	New

	Source	Destination
File	ossec-hids/active-response.c	ossec-hids/active-response.c
Line	123	123
Object	rules_group	rules_group

Code Snippet
File Name ossec-hids/active-response.c
Method int ReadActiveResponses(XML_NODE node, void *d1, void *d2)

```
....
123.            tmp_ar->rules_group = strdup(node[i]->content);
```

Memory Leak\Path 5:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=142
Status	New

	Source	Destination
File	ossec-hids/active-response.c	ossec-hids/active-response.c
Line	271	271
Object	name	name

Code Snippet
File Name ossec-hids/active-response.c
Method int ReadActiveResponses(XML_NODE node, void *d1, void *d2)

```
....
271.      tmp_ar->name = (char *) calloc(OS_FLSIZE + 1, sizeof(char));
```

Memory Leak\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=143
Status	New

	Source	Destination
File	ossec-hids/b64.c	ossec-hids/b64.c
Line	58	58
Object	out	out

Code Snippet
File Name ossec-hids/b64.c
Method char *encode_base64(int size, char *src)

```
....
58.      out = (char *)calloc(sizeof(char), size * 4 / 3 + 4);
```

Memory Leak\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=144
Status	New

	Source	Destination
File	ossec-hids/imsg.c	ossec-hids/imsg.c
Line	175	175

Object	data	data

Code Snippet
File Name ossec-hids/imsg.c
Method imsg_get(struct imsgbuf *ibuf, struct imsg *imsg)

```
....
175.        if ((imsg->data = malloc(datalen)) == NULL)
```

Short Overflow

Query Path:
CPP\Cx\CPP Integer Overflow\Short Overflow Version:1

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

Short Overflow\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=90
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 191 of ossec-hids/blast.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	ossec-hids/blast.c	ossec-hids/blast.c
Line	206	206
Object	AssignExpr	AssignExpr

Code Snippet
File Name ossec-hids/blast.c
Method local int construct(struct huffman *h, const unsigned char *rep, int n)

```
....
206.            length[symbol++] = len;
```

Short Overflow\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=91
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 665 of ossec-hids/puff.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	ossec-hids/puff.c	ossec-hids/puff.c
Line	711	711
Object	AssignExpr	AssignExpr

Code Snippet
File Name ossec-hids/puff.c
Method local int dynamic(struct state *s)

```
....
711.              lengths[index++] = symbol;
```

Short Overflow\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=92
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 665 of ossec-hids/puff.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	ossec-hids/puff.c	ossec-hids/puff.c
Line	727	727
Object	AssignExpr	AssignExpr

Code Snippet
File Name ossec-hids/puff.c
Method local int dynamic(struct state *s)

```
....
727.                  lengths[index++] = len;
```

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=1050059&projectid=50049&pathid=75
Status	New

The function datalen in ossec-hids/imsg.c at line 156 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	ossec-hids/imsg.c	ossec-hids/imsg.c
Line	175	175
Object	datalen	datalen

Code Snippet
File Name ossec-hids/imsg.c
Method imsg_get(struct imsgbuf *ibuf, struct imsg *imsg)

```
....
175.        if ((imsg->data = malloc(datalen)) == NULL)
```

Use After Free

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

Categories

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

Description
Use After Free\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=145
Status	New

The pointer ip_address at ossec-hids/validate_op.c in line 241 is being used after it has been freed.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	266	268
Object	ip_address	ip_address

Code Snippet
File Name ossec-hids/validate_op.c
Method int OS_IsValidIP(const char *in_address, os_ip *final_ip)

```
....
266.                free(ip_address);
....
268.            os_strdup(in_address+1, ip_address);
```

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=1050059&projectid=50049&pathid=17
Status	New

The ReadActiveResponses method calls the snprintf function, at line 26 of ossec-hids/active-response.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	ossec-hids/active-response.c	ossec-hids/active-response.c
Line	275	275
Object	snprintf	snprintf

Code Snippet
File Name ossec-hids/active-response.c
Method int ReadActiveResponses(XML_NODE node, void *d1, void *d2)

```
....
275.      snprintf(tmp_ar->name, OS_FLSIZE, "%s%d",
```

Unchecked Return Value\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=18
Status	New

The CJSON_PUBLIC method calls the sprintf function, at line 89 of ossec-hids/cJSON.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	92	92
Object	sprintf	sprintf

Code Snippet
File Name ossec-hids/cJSON.c

Method	CJSON_PUBLIC(const char*) cJSON_Version(void)

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

Unchecked Return Value\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=19
Status	New

The *internal_malloc method calls the malloc function, at line 130 of ossec-hids/cJSON.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	132	132
Object	malloc	malloc

Code Snippet	
File Name	ossec-hids/cJSON.c
Method	static void *internal_malloc(size_t size)

```
....
132.      return malloc(size);
```

Unchecked Return Value\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=20
Status	New

The *internal_realloc method calls the realloc function, at line 138 of ossec-hids/cJSON.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	140	140
Object	realloc	realloc

Code Snippet	
File Name	ossec-hids/cJSON.c

Method	static void *internal_realloc(void *pointer, size_t size)

```
....
140.      return realloc(pointer, size);
```

Unchecked Return Value\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=21
Status	New

The print_string_ptr method calls the sprintf function, at line 828 of ossec-hids/cJSON.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	937	937
Object	sprintf	sprintf

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

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

Unchecked Return Value\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=22
Status	New

The *_read_file method calls the snprintf function, at line 20 of ossec-hids/validate_op.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	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	31	31
Object	snprintf	snprintf

Code Snippet

File Name	ossec-hids/validate_op.c
Method	static char *_read_file(const char *high_name, const char *low_name, const char *defines_file)

```
....
31.          snprintf(def_file, OS_FLSIZE, "%s", defines_file);
```

Unchecked Return Value\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=23
Status	New

The *_read_file method calls the snprintf function, at line 20 of ossec-hids/validate_op.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	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	33	33
Object	snprintf	snprintf

Code Snippet	
File Name	ossec-hids/validate_op.c
Method	static char *_read_file(const char *high_name, const char *low_name, const char *defines_file)

```
....
33.          snprintf(def_file, OS_FLSIZE, "%s%s", DEFAULTDIR,
defines_file);
```

Unchecked Return Value\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=24
Status	New

The *__gethour method calls the snprintf function, at line 454 of ossec-hids/validate_op.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	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	505	505
Object	snprintf	snprintf

Code Snippet
File Name ossec-hids/validate_op.c
Method static const char *__gethour(const char *str, char *ossec_hour)

```
....
505.              snprintf(ossec_hour, 6, "%02d:%02d", chour, cmin);
```

Unchecked Return Value\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=25
Status	New

The *__gethour method calls the snprintf function, at line 454 of ossec-hids/validate_op.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	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	522	522
Object	snprintf	snprintf

Code Snippet
File Name ossec-hids/validate_op.c
Method static const char *__gethour(const char *str, char *ossec_hour)

```
....
522.              snprintf(ossec_hour, 6, "%02d:%02d", chour, cmin);
```

Unchecked Return Value\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=26
Status	New

The *__gethour method calls the snprintf function, at line 454 of ossec-hids/validate_op.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	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	528	528
Object	snprintf	snprintf

Code Snippet
File Name ossec-hids/validate_op.c
Method static const char *__gethour(const char *str, char *ossec_hour)

```
....
528.          snprintf(ossec_hour, 6, "%02d:%02d", chour, cmin);
```

Unchecked Return Value\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=27
Status	New

The *OS_IsValidTime method calls the snprintf function, at line 537 of ossec-hids/validate_op.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	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	598	598
Object	snprintf	snprintf

Code Snippet
File Name ossec-hids/validate_op.c
Method char *OS_IsValidTime(const char *time_str)

```
....
598.          snprintf(ret, 12, "!%s%s", second_hour, first_hour);
```

Unchecked Return Value\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=28
Status	New

The *OS_IsValidTime method calls the snprintf function, at line 537 of ossec-hids/validate_op.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	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	603	603
Object	snprintf	snprintf

Code Snippet

File Name	ossec-hids/validate_op.c
Method	char *OS_IsValidTime(const char *time_str)

```
....
603.      snprintf(ret, 12, "%c%s%s", ng == 0 ? '.' : '!', first_hour,
second_hour);
```

Unchecked Return Value\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=29
Status	New

The *OS_IsValidUniqueTime method calls the snprintf function, at line 626 of ossec-hids/validate_op.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	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	635	635
Object	snprintf	snprintf

Code Snippet	
File Name	ossec-hids/validate_op.c
Method	char *OS_IsValidUniqueTime(const char *time_str)

```
....
635.      snprintf(mytime, 128, "%s-%s", time_str, time_str);
```

Unchecked Return Value\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=30
Status	New

The ReadActiveResponses method calls the rules_id function, at line 26 of ossec-hids/active-response.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	ossec-hids/active-response.c	ossec-hids/active-response.c
Line	121	121
Object	rules_id	rules_id

Code Snippet

File Name	ossec-hids/active-response.c
Method	int ReadActiveResponses(XML_NODE node, void *d1, void *d2)

```
....
121.            tmp_ar->rules_id = strdup(node[i]->content);
```

Unchecked Return Value\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=31
Status	New

The ReadActiveResponses method calls the rules_group function, at line 26 of ossec-hids/active-response.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	ossec-hids/active-response.c	ossec-hids/active-response.c
Line	123	123
Object	rules_group	rules_group

Code Snippet

File Name	ossec-hids/active-response.c
Method	int ReadActiveResponses(XML_NODE node, void *d1, void *d2)

```
....
123.            tmp_ar->rules_group = strdup(node[i]->content);
```

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=1050059&projectid=50049&pathid=77
Status	New

The variable declared in null at ossec-hids/blast.c in line 446 is not initialized when it is used by in at ossec-hids/blast.c in line 383.

	Source	Destination

File	ossec-hids/blast.c	ossec-hids/blast.c
Line	453	394
Object	null	in

Code Snippet
File Name ossec-hids/blast.c
Method int main(void)

```
....
453.      ret = blast(inf, stdin, outf, stdout, &left, NULL);
```

File Name ossec-hids/blast.c

Method int blast(blast_in infun, void *inhow, blast_out outfun, void *outhow,

```
....
394.          s.in = *in;
```

NULL Pointer Dereference\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=78
Status	New

The variable declared in null at ossec-hids/blast.c in line 446 is not initialized when it is used by in at ossec-hids/blast.c in line 383.

	Source	Destination
File	ossec-hids/blast.c	ossec-hids/blast.c
Line	453	394
Object	null	in

Code Snippet
File Name ossec-hids/blast.c
Method int main(void)

```
....
453.      ret = blast(inf, stdin, outf, stdout, &left, NULL);
```

File Name ossec-hids/blast.c

Method int blast(blast_in infun, void *inhow, blast_out outfun, void *outhow,

```
....
394.          s.in = *in;
```

NULL Pointer Dereference\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=79
Status	New

The variable declared in null at ossec-hids/cJSON.c in line 985 is not initialized when it is used by content at ossec-hids/cJSON.c in line 964.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	989	966
Object	null	content

Code Snippet

File Name ossec-hids/cJSON.c

Method static parse_buffer *skip_utf8_bom(parse_buffer * const buffer)

```
....
989.            return NULL;
```

File Name ossec-hids/cJSON.c

Method static parse_buffer *buffer_skip_whitespace(parse_buffer * const buffer)

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

NULL Pointer Dereference\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=80
Status	New

The variable declared in null at ossec-hids/cJSON.c in line 964 is not initialized when it is used by content at ossec-hids/cJSON.c in line 964.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	968	966
Object	null	content

Code Snippet

File Name ossec-hids/cJSON.c

Method	static parse_buffer *buffer_skip_whitespace(parse_buffer * const buffer)

```
....
968.            return NULL;
....
966.        if ((buffer == NULL) || (buffer->content == NULL))
```

NULL Pointer Dereference\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=81
Status	New

The variable declared in null at ossec-hids/cJSON.c in line 2799 is not initialized when it is used by valuestring at ossec-hids/cJSON.c in line 2799.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2883	2850
Object	null	valuestring

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

```
....
2883.               cJSON *a_element = NULL;
....
2850.               if (strcmp(a->valuestring, b->valuestring) == 0)
```

NULL Pointer Dereference\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=82
Status	New

The variable declared in null at ossec-hids/cJSON.c in line 2799 is not initialized when it is used by valuestring at ossec-hids/cJSON.c in line 2799.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2883	2846
Object	null	valuestring

Code Snippet

File Name	ossec-hids/cJSON.c
Method	CJSON_PUBLIC(cJSON_bool) cJSON_Compare(const cJSON * const a, const cJSON * const b, const cJSON_bool case_sensitive)

```
....
2883.               cJSON *a_element = NULL;
....
2846.               if ((a->valuestring == NULL) || (b->valuestring ==
NULL))
```

NULL Pointer Dereference\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=83
Status	New

The variable declared in newitem at ossec-hids/cJSON.c in line 2550 is not initialized when it is used by valueint at ossec-hids/cJSON.c in line 2550.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2552	2570
Object	newitem	valueint

Code Snippet

File Name	ossec-hids/cJSON.c
Method	CJSON_PUBLIC(cJSON *) cJSON_Duplicate(const cJSON *item, cJSON_bool recurse)

```
....
2552.       cJSON *newitem = NULL;
....
2570.       newitem->valueint = item->valueint;
```

NULL Pointer Dereference\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=84
Status	New

The variable declared in newitem at ossec-hids/cJSON.c in line 2550 is not initialized when it is used by type at ossec-hids/cJSON.c in line 2550.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2552	2569

Object	newitem	type

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

```
....
2552.       cJSON *newitem = NULL;
....
2569.       newitem->type = item->type & (~cJSON_IsReference);
```

NULL Pointer Dereference\Path 9:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=85
Status New

The variable declared in newitem at ossec-hids/cJSON.c in line 2550 is not initialized when it is used by valuedouble at ossec-hids/cJSON.c in line 2550.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2552	2571
Object	newitem	valuedouble

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

```
....
2552.       cJSON *newitem = NULL;
....
2571.       newitem->valuedouble = item->valuedouble;
```

NULL Pointer Dereference\Path 10:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=86
Status New

The variable declared in newitem at ossec-hids/cJSON.c in line 2550 is not initialized when it is used by valuestring at ossec-hids/cJSON.c in line 2550.

	Source	Destination

File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2552	2574
Object	newitem	valuestring

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

```
....
2552.       cJSON *newitem = NULL;
....
2574.           newitem->valuestring = (char*)cJSON_strdup((unsigned
char*)item->valuestring, &global_hooks);
```

NULL Pointer Dereference\Path 11:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=87
Status New

The variable declared in newitem at ossec-hids/cJSON.c in line 2550 is not initialized when it is used by string at ossec-hids/cJSON.c in line 2550.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2552	2582
Object	newitem	string

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

```
....
2552.       cJSON *newitem = NULL;
....
2582.           newitem->string = (item->type&cJSON_StringIsConst) ?
item->string : (char*)cJSON_strdup((unsigned char*)item->string,
&global_hooks);
```

NULL Pointer Dereference\Path 12:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=88
Status New

The variable declared in newitem at ossec-hids/cJSON.c in line 2550 is not initialized when it is used by string at ossec-hids/cJSON.c in line 2550.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2552	2583
Object	newitem	string

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

```
....
2552.       cJSON *newitem = NULL;
....
2583.           if (!newitem->string)
```

NULL Pointer Dereference\Path 13:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=89
Status New

The variable declared in newitem at ossec-hids/cJSON.c in line 2550 is not initialized when it is used by valuestring at ossec-hids/cJSON.c in line 2550.

	Source	Destination
File	ossec-hids/cJSON.c	ossec-hids/cJSON.c
Line	2552	2575
Object	newitem	valuestring

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

```
....
2552.       cJSON *newitem = NULL;
....
2575.           if (!newitem->valuestring)
```

Incorrect Permission Assignment For Critical Resources

Query Path:
CPP\Cx\CPP Low Visibility\Incorrect Permission Assignment For Critical Resources Version:1

Categories

FISMA 2014: Access Control

NIST SP 800-53: AC-3 Access Enforcement (P1)
OWASP Top 10 2017: A2-Broken Authentication

Description

Incorrect Permission Assignment For Critical Resources\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=194
Status	New

	Source	Destination
File	ossec-hids/active-response.c	ossec-hids/active-response.c
Line	74	74
Object	chmod	chmod

Code Snippet
File Name ossec-hids/active-response.c
Method int ReadActiveResponses(XML_NODE node, void *d1, void *d2)

```
....
74.      if ((chmod(DEFAULTARPATH, 0440)) == -1) {
```

Incorrect Permission Assignment For Critical Resources\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=195
Status	New

	Source	Destination
File	ossec-hids/active-response.c	ossec-hids/active-response.c
Line	53	53
Object	fp	fp

Code Snippet
File Name ossec-hids/active-response.c
Method int ReadActiveResponses(XML_NODE node, void *d1, void *d2)

```
....
53.      fp = fopen(DEFAULTARPATH, "a");
```

Incorrect Permission Assignment For Critical Resources\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=500

	49&pathid=196
Status	New

	Source	Destination
File	ossec-hids/run_rk_check.c	ossec-hids/run_rk_check.c
Line	118	118
Object	fp	fp

Code Snippet
File Name ossec-hids/run_rk_check.c
Method void run_rk_check()

```
....
118.              fp = fopen(rootcheck.rootkit_files, "r");
```

Incorrect Permission Assignment For Critical Resources\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=197
Status	New

	Source	Destination
File	ossec-hids/run_rk_check.c	ossec-hids/run_rk_check.c
Line	139	139
Object	fp	fp

Code Snippet
File Name ossec-hids/run_rk_check.c
Method void run_rk_check()

```
....
139.              fp = fopen(rootcheck.rootkit_trojans, "r");
```

Incorrect Permission Assignment For Critical Resources\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=198
Status	New

	Source	Destination
File	ossec-hids/run_rk_check.c	ossec-hids/run_rk_check.c
Line	161	161

Object	fp	fp

Code Snippet
File Name ossec-hids/run_rk_check.c
Method void run_rk_check()

```
....
161.                fp = fopen(rootcheck.winaudit, "r");
```

Incorrect Permission Assignment For Critical Resources\Path 6:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=199
Status New

	Source	Destination
File	ossec-hids/run_rk_check.c	ossec-hids/run_rk_check.c
Line	177	177
Object	fp	fp

Code Snippet
File Name ossec-hids/run_rk_check.c
Method void run_rk_check()

```
....
177.                fp = fopen(rootcheck.winmalware, "r");
```

Incorrect Permission Assignment For Critical Resources\Path 7:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=200
Status New

	Source	Destination
File	ossec-hids/run_rk_check.c	ossec-hids/run_rk_check.c
Line	193	193
Object	fp	fp

Code Snippet
File Name ossec-hids/run_rk_check.c
Method void run_rk_check()

```
....
193.              fp = fopen(rootcheck.winapps, "r");
```

Incorrect Permission Assignment For Critical Resources\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=201
Status	New

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	39	39
Object	fp	fp

Code Snippet
File Name ossec-hids/validate_op.c
Method static char *_read_file(const char *high_name, const char *low_name, const char *defines_file)

```
....
39.       fp = fopen(def_file, "r");
```

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=1050059&projectid=50049&pathid=187
Status	New

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	56	56
Object	fgets	fgets

Code Snippet

File Name	ossec-hids/validate_op.c
Method	static char *_read_file(const char *high_name, const char *low_name, const char *defines_file)

```
....
56.      while (fgets(buf, OS_SIZE_1024 , fp) != NULL) {
```

Improper Resource Access Authorization\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=188
Status	New

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	56	56
Object	buf	buf

Code Snippet

File Name	ossec-hids/validate_op.c
Method	static char *_read_file(const char *high_name, const char *low_name, const char *defines_file)

```
....
56.      while (fgets(buf, OS_SIZE_1024 , fp) != NULL) {
```

Improper Resource Access Authorization\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=189
Status	New

	Source	Destination
File	ossec-hids/blast.c	ossec-hids/blast.c
Line	437	437
Object	hold	hold

Code Snippet

File Name	ossec-hids/blast.c
Method	local unsigned inf(void *how, unsigned char **buf)

```
....
437.      return fread(hold, 1, CHUNK, (FILE *)how);
```

Improper Resource Access Authorization\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=190
Status	New

	Source	Destination
File	ossec-hids/active-response.c	ossec-hids/active-response.c
Line	281	281
Object	fprintf	fprintf

Code Snippet
File Name ossec-hids/active-response.c
Method int ReadActiveResponses(XML_NODE node, void *d1, void *d2)

```
....
281.      fprintf(fp, "%s - %s - %d\n",
```

Improper Resource Access Authorization\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=191
Status	New

	Source	Destination
File	ossec-hids/blast.c	ossec-hids/blast.c
Line	455	455
Object	fprintf	fprintf

Code Snippet
File Name ossec-hids/blast.c
Method int main(void)

```
....
455.          fprintf(stderr, "blast error: %d\n", ret);
```

Improper Resource Access Authorization\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=192
Status	New

	Source	Destination
File	ossec-hids/blast.c	ossec-hids/blast.c
Line	461	461
Object	fprintf	fprintf

Code Snippet
File Name ossec-hids/blast.c
Method int main(void)

```
....
461.          fprintf(stderr, "blast warning: %u unused bytes of
input\n", left);
```

Improper Resource Access Authorization\Path 7:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=193
Status New

	Source	Destination
File	ossec-hids/blast.c	ossec-hids/blast.c
Line	442	442
Object	fwrite	fwrite

Code Snippet
File Name ossec-hids/blast.c
Method local int outf(void *how, unsigned char *buf, unsigned len)

```
....
442.      return fwrite(buf, 1, len, (FILE *)how) != len;
```

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=1050059&projectid=50049&pathid=203
Status New

The ReadActiveResponses method in ossec-hids/active-response.c file utilizes fopen that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	ossec-hids/active-response.c	ossec-hids/active-response.c
Line	53	53
Object	fopen	fopen

Code Snippet
File Name ossec-hids/active-response.c
Method int ReadActiveResponses(XML_NODE node, void *d1, void *d2)

```
....
53.      fp = fopen(DEFAULTARPATH, "a");
```

TOCTOU\Path 2:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=204
Status New

The run_rk_check method in ossec-hids/run_rk_check.c file utilizes fopen that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	ossec-hids/run_rk_check.c	ossec-hids/run_rk_check.c
Line	118	118
Object	fopen	fopen

Code Snippet
File Name ossec-hids/run_rk_check.c
Method void run_rk_check()

```
....
118.              fp = fopen(rootcheck.rootkit_files, "r");
```

TOCTOU\Path 3:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=205
Status New

The run_rk_check method in ossec-hids/run_rk_check.c file utilizes fopen that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	ossec-hids/run_rk_check.c	ossec-hids/run_rk_check.c

Line	139	139
Object	fopen	fopen

Code Snippet
File Name ossec-hids/run_rk_check.c
Method void run_rk_check()

```
....
139.              fp = fopen(rootcheck.rootkit_trojans, "r");
```

TOCTOU\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=206
Status	New

The run_rk_check method in ossec-hids/run_rk_check.c file utilizes fopen that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	ossec-hids/run_rk_check.c	ossec-hids/run_rk_check.c
Line	161	161
Object	fopen	fopen

Code Snippet
File Name ossec-hids/run_rk_check.c
Method void run_rk_check()

```
....
161.              fp = fopen(rootcheck.winaudit, "r");
```

TOCTOU\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=207
Status	New

The run_rk_check method in ossec-hids/run_rk_check.c file utilizes fopen that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	ossec-hids/run_rk_check.c	ossec-hids/run_rk_check.c
Line	177	177
Object	fopen	fopen

Code Snippet	
File Name	ossec-hids/run_rk_check.c
Method	void run_rk_check()

```
....
177.              fp = fopen(rootcheck.winmalware, "r");
```

TOCTOU\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=208
Status	New

The run_rk_check method in ossec-hids/run_rk_check.c file utilizes fopen that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	ossec-hids/run_rk_check.c	ossec-hids/run_rk_check.c
Line	193	193
Object	fopen	fopen

Code Snippet	
File Name	ossec-hids/run_rk_check.c
Method	void run_rk_check()

```
....
193.              fp = fopen(rootcheck.winapps, "r");
```

TOCTOU\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=209
Status	New

The *_read_file method in ossec-hids/validate_op.c file utilizes fopen that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	39	39
Object	fopen	fopen

Code Snippet	
File Name	ossec-hids/validate_op.c

Method	static char *_read_file(const char *high_name, const char *low_name, const char *defines_file)

```
....
39.      fp = fopen(def_file, "r");
```

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=1050059&projectid=50049&pathid=35
Status	New

The buffer allocated by <= in ossec-hids/blast.c at line 191 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	ossec-hids/blast.c	ossec-hids/blast.c
Line	212	212
Object	<=	<=

Code Snippet	
File Name	ossec-hids/blast.c
Method	local int construct(struct huffman *h, const unsigned char *rep, int n)

```
....
212.      for (len = 0; len <= MAXBITS; len++)
```

Potential Off by One Error in Loops\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=36
Status	New

The buffer allocated by <= in ossec-hids/puff.c at line 340 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination

File	ossec-hids/puff.c	ossec-hids/puff.c
Line	348	348
Object	<=	<=

Code Snippet
File Name ossec-hids/puff.c
Method local int construct(struct huffman *h, const short *length, int n)

```
....
348.      for (len = 0; len <= MAXBITS; len++)
```

Potential Off by One Error in Loops\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=37
Status	New

The buffer allocated by <= in ossec-hids/run_rk_check.c at line 57 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	ossec-hids/run_rk_check.c	ossec-hids/run_rk_check.c
Line	283	283
Object	<=	<=

Code Snippet
File Name ossec-hids/run_rk_check.c
Method void run_rk_check()

```
....
283.          for (li = 0; li <= rk_sys_count; li++) {
```

Potential Off by One Error in Loops\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=38
Status	New

The buffer allocated by <= in ossec-hids/validate_op.c at line 674 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	749	749

Object	<=	<=

Code Snippet
File Name ossec-hids/validate_op.c
Method char *OS_IsValidDay(const char *day_str)

```
....
749.      for (i = 0; i <= 6; i++) {
```

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=1050059&projectid=50049&pathid=32
Status New

	Source	Destination
File	ossec-hids/lgc.c	ossec-hids/lgc.c
Line	471	471
Object	sizeof	sizeof

Code Snippet
File Name ossec-hids/lgc.c
Method static lu_mem traversetable (global_State *g, Table *h) {

```
....
471.                            sizeof(Proto *) * f->sizep +
```

Use of Sizeof On a Pointer Type\Path 2:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=33
Status New

	Source	Destination
File	ossec-hids/lgc.c	ossec-hids/lgc.c
Line	1045	1045
Object	sizeof	sizeof

Code Snippet
File Name ossec-hids/lgc.c

Method	static lu_mem singlestep (lua_State *L) {

```
....
1045.        g->GCmemtrav = g->strt.size * sizeof(GCObject*);
```

Use of Sizeof On a Pointer Type\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=34
Status	New

	Source	Destination
File	ossec-hids/lobject.c	ossec-hids/lobject.c
Line	208	208
Object	sizeof	sizeof

Code Snippet
File Name ossec-hids/lobject.c
Method const char *luaO_pushvfstring (lua_State *L, const char *fmt, va_list argp) {

```
....
208.          char buff[4*sizeof(void *) + 8]; /* should be enough space
for a `%p' */
```

Reliance on DNS Lookups in a Decision

Query Path:
CPP\Cx\CPP Low Visibility\Reliance on DNS Lookups in a Decision Version:0

Categories

FISMA 2014: Identification And Authentication
NIST SP 800-53: SC-23 Session Authenticity (P1)

Description

Reliance on DNS Lookups in a Decision\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050059&projectid=50049&pathid=74
Status	New

The OS_IsValidIP method performs a reverse DNS lookup with getaddrinfo, at line 241 of ossec-hids/validate_op.c. The application then makes a security decision, !=, in ossec-hids/validate_op.c line 241, even though this hostname is not reliable and can be easily spoofed.

	Source	Destination
File	ossec-hids/validate_op.c	ossec-hids/validate_op.c
Line	299	299

Object	getaddrinfo	!=

Code Snippet
File Name ossec-hids/validate_op.c
Method int OS_IsValidIP(const char *in_address, os_ip *final_ip)

```
....
299.      if (getaddrinfo(ip_address, NULL, &hints, &result) != 0) {
```

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=1050059&projectid=50049&pathid=202
Status	New

The system data read by merror in the file ossec-hids/active-response.c at line 341 is potentially exposed by ReadActiveResponses found in ossec-hids/active-response.c at line 26.

	Source	Destination
File	ossec-hids/active-response.c	ossec-hids/active-response.c
Line	341	281
Object	errno	fprintf

Code Snippet
File Name ossec-hids/active-response.c
Method merror(MEM_ERROR, __local_name, errno, strerror(errno));

```
....
341.          merror(MEM_ERROR, __local_name, errno, strerror(errno));
```

File Name ossec-hids/active-response.c
Method int ReadActiveResponses(XML_NODE node, void *d1, void *d2)

```
....
281.      fprintf(fp, "%s - %s - %d\n",
```

Buffer Overflow LongString

Risk

What might happen

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

Cause

How does it happen

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

General Recommendations

How to avoid it

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

Source Code Examples

CPP

Overflowing Buffers

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

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

Checked Buffers

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

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

Buffer Overflow OutOfBound

Risk

What might happen

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

Cause

How does it happen

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

General Recommendations

How to avoid it

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

Source Code Examples

Buffer Overflow StrcpyStrcat

Risk

What might happen

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

Cause

How does it happen

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

General Recommendations

How to avoid it

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

Source Code Examples

Buffer Overflow IndexFromInput

Risk

What might happen

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

Cause

How does it happen

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

General Recommendations

How to avoid it

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

Source Code Examples

Buffer Overflow boundcpy WrongSizeParam

Risk

What might happen

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

Cause

How does it happen

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

General Recommendations

How to avoid it

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

Source Code Examples

MemoryFree on StackVariable

Risk

What might happen

Undefined Behavior may result with a crash. Crashes may give an attacker valuable information about the system and the program internals. Furthermore, it may leave unprotected files (e.g memory) that may be exploited.

Cause

How does it happen

Calling free() on a variable that was not dynamically allocated (e.g. malloc) will result with an Undefined Behavior.

General Recommendations

How to avoid it

Use free() only on dynamically allocated variables in order to prevent unexpected behavior from the compiler.

Source Code Examples

CPP

Bad - Calling free() on a static variable

```
void clean_up(){
  char temp[256];
  do_something();
  free(tmp);
  return;
}
```

Good - Calling free() only on variables that were dynamically allocated

```
void clean_up(){
  char *buff;
  buff = (char*) malloc(1024);
  free(buff);
  return;
}
```

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);
```

Short Overflow

Risk

What might happen

Assigning large data types into smaller data types, without proper checks and explicit casting, will lead to undefined behavior and unintentional effects, such as data corruption (e.g. value wraparound, wherein maximum values become minimum values); system crashes; infinite loops; logic errors, such as bypassing of security mechanisms; or even buffer overflows leading to arbitrary code execution.

Cause

How does it happen

This flaw can occur when implicitly casting numerical data types of a larger size, into a variable with a data type of a smaller size. This forces the program to discard some bits of information from the number. Depending on how the numerical data types are stored in memory, this is often the bits with the highest value, causing substantial corruption of the stored number. Alternatively, the sign bit of a signed integer could be lost, completely reversing the intention of the number.

General Recommendations

How to avoid it

- Avoid casting larger data types to smaller types.
- Prefer promoting the target variable to a large enough data type.
- If downcasting is necessary, always check that values are valid and in range of the target type, before casting

---

Source Code Examples

CPP

Unsafe Downsize Casting

```
int unsafe_addition(short op1, int op2) {

    // op2 gets forced from int into a short
    short total = op1 + op2;

    return total;
}
```

Safer Use of Proper Data Types

```
int safe_addition(short op1, int op2) {

    // total variable is of type int, the largest type that is needed
    int total = 0;

    // check if total will overflow available integer size
    if (INT_MAX - abs(op2) > op1)
```

```
    {
        total = op1 + op2;
    }
    else
    {
        // instead of overflow, saturate (but this is not always a good thing)
        total = INT_MAX
    }

    return total;
}
```

Dangerous Functions

Risk

What might happen

Use of dangerous functions may expose varying risks associated with each particular function, with potential impact of improper usage of these functions varying significantly. The presence of such functions indicates a flaw in code maintenance policies and adherence to secure coding practices, in a way that has allowed introducing known dangerous code into the application.

Cause

How does it happen

A dangerous function has been identified within the code. Functions are often deemed dangerous to use for numerous reasons, as there are different sets of vulnerabilities associated with usage of such functions. For example, some string copy and concatenation functions are vulnerable to Buffer Overflow, Memory Disclosure, Denial of Service and more. Use of these functions is not recommended.

General Recommendations

How to avoid it

- Deploy a secure and recommended alternative to any functions that were identified as dangerous.
    - If no secure alternative is found, conduct further researching and testing to identify whether current usage successfully sanitizes and verifies values, and thus successfully avoids the use-cases for whom the function is indeed dangerous
- Conduct a periodical review of methods that are in use, to ensure that all external libraries and built-in functions are up-to-date and whose use has not been excluded from best secure coding practices.

Source Code Examples

CPP

Buffer Overflow in gets()

```
int main()
{
	char buf[10];

	printf("Please enter your name: ");
	gets(buf); // veryveryverylongname
	if (buf == ACCEPTED_NAME)
	{
		// Do something
	}
	return 0;
}
```

Safe reading from user

```
int main()
{
	char buf[10];

	printf("Please enter your name: ");
	fgets(buf, sizeof(buf), stdin); //setting the amount of bytes to read
	if (buf == ACCEPTED_NAME)
	{
		//Do something
	}
	return 0;
}
```

Unsafe function for string copy

```
int main(int argc, char* argv[])
{
	char buf[10];
	strcpy(buf, argv[1]); // overflow occurs when len(argv[1]) > 10 bytes

	return 0;
}
```

Safe string copy

```
int main(int argc, char* argv[])
{
	char buf[10];
	strncpy(buf, argv[1], sizeof(buf));
	buf[9]= '\0'; //strncpy doesn't NULL terminates

	return 0;
}
```

Unsafe format string

```
int main(int argc, char* argv[])
{
	printf(argv[1]); // If argv[1] contains a format token, such as %s,%x or %d, will cause
an access violation
	return 0;
}
```

Safe format string

```
int main(int argc, char* argv[])
{
      printf("%s", argv[1]); // Second parameter is not a formattable string
      return 0;
}
```

Failure to Release Memory Before Removing Last Reference ('Memory Leak')

Weakness ID: 401 *(Weakness Base)* **Status:** Draft

Description

Description Summary

The software does not sufficiently track and release allocated memory after it has been used, which slowly consumes remaining memory.

Extended Description

This is often triggered by improper handling of malformed data or unexpectedly interrupted sessions.

Terminology Notes

"memory leak" has sometimes been used to describe other kinds of issues, e.g. for information leaks in which the contents of memory are inadvertently leaked (CVE-2003-0400 is one such example of this terminology conflict).

Time of Introduction

- Architecture and Design
- Implementation

Applicable Platforms

Languages

C

C++

Modes of Introduction

Memory leaks have two common and sometimes overlapping causes:

- Error conditions and other exceptional circumstances
- Confusion over which part of the program is responsible for freeing the memory

Common Consequences

Scope	Effect
Availability	Most memory leaks result in general software reliability problems, but if an attacker can intentionally trigger a memory leak, the attacker might be able to launch a denial of service attack (by crashing or hanging the program) or take advantage of other unexpected program behavior resulting from a low memory condition.

Likelihood of Exploit

Medium

Demonstrative Examples

Example 1

The following C function leaks a block of allocated memory if the call to read() fails to return the expected number of bytes:

(Bad Code)

Example Language: **C**

```
char* getBlock(int fd) {
char* buf = (char*) malloc(BLOCK_SIZE);
if (!buf) {
return NULL;
}
if (read(fd, buf, BLOCK_SIZE) != BLOCK_SIZE) {

return NULL;
}
```

```
return buf;
}
```

Example 2

Here the problem is that every time a connection is made, more memory is allocated. So if one just opened up more and more connections, eventually the machine would run out of memory.

(Bad Code)

Example Language: **C**

```
bar connection(){
foo = malloc(1024);
return foo;
}
endConnection(bar foo) {

free(foo);
}
int main() {

while(1) //thread 1
//On a connection
foo=connection(); //thread 2
//When the connection ends
endConnection(foo)
}
```

Observed Examples

Reference	Description
CVE-2005-3119	Memory leak because function does not free() an element of a data structure.
CVE-2004-0427	Memory leak when counter variable is not decremented.
CVE-2002-0574	Memory leak when counter variable is not decremented.
CVE-2005-3181	Kernel uses wrong function to release a data structure, preventing data from being properly tracked by other code.
CVE-2004-0222	Memory leak via unknown manipulations as part of protocol test suite.
CVE-2001-0136	Memory leak via a series of the same command.

Potential Mitigations

Pre-design: Use a language or compiler that performs automatic bounds checking.

Phase: Architecture and Design

Use an abstraction library to abstract away risky APIs. Not a complete solution.

Pre-design through Build: The Boehm-Demers-Weiser Garbage Collector or valgrind can be used to detect leaks in code. This is not a complete solution as it is not 100% effective.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	398	Indicator of Poor Code Quality	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Category	399	Resource Management Errors	**Development Concepts (primary)699**
ChildOf	Category	633	Weaknesses that Affect Memory	**Resource-specific Weaknesses (primary)631**
ChildOf	Category	730	OWASP Top Ten 2004 Category A9 - Denial of Service	**Weaknesses in OWASP Top Ten (2004) (primary)711**
ChildOf	Weakness Base	772	Missing Release of Resource after Effective	**Research Concepts (primary)1000**

			Lifetime	
MemberOf	View	630	Weaknesses Examined by SAMATE	**Weaknesses Examined by SAMATE (primary)630**
CanFollow	Weakness Class	390	Detection of Error Condition Without Action	Research Concepts1000

Relationship Notes

This is often a resultant weakness due to improper handling of malformed data or early termination of sessions.

Affected Resources

- Memory

Functional Areas

- Memory management

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
PLOVER			Memory leak
7 Pernicious Kingdoms			Memory Leak
CLASP			Failure to deallocate data
OWASP Top Ten 2004	A9	CWE More Specific	Denial of Service

White Box Definitions

A weakness where the code path has:

1. start statement that allocates dynamically allocated memory resource

2. end statement that loses identity of the dynamically allocated memory resource creating situation where dynamically allocated memory resource is never relinquished

Where "loses" is defined through the following scenarios:

1. identity of the dynamic allocated memory resource never obtained

2. the statement assigns another value to the data element that stored the identity of the dynamically allocated memory resource and there are no aliases of that data element

3. identity of the dynamic allocated memory resource obtained but never passed on to function for memory resource release

4. the data element that stored the identity of the dynamically allocated resource has reached the end of its scope at the statement and there are no aliases of that data element

References

J. Whittaker and H. Thompson. "How to Break Software Security". Addison Wesley. 2003.

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	PLOVER		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-08-15		Veracode	External
	Suggested OWASP Top Ten 2004 mapping		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Relationships, Other Notes, References, Relationship Notes, Taxonomy Mappings, Terminology Notes		
2008-10-14	CWE Content Team	MITRE	Internal
	updated Description		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Other Notes		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Name		
2009-07-17	KDM Analytics		External
	Improved the White Box Definition		

2009-07-27	CWE Content Team	MITRE	Internal
	updated White Box Definitions		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Modes of Introduction, Other Notes		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		

Previous Entry Names

Change Date	Previous Entry Name
2008-04-11	Memory Leak
2009-05-27	Failure to Release Memory Before Removing Last Reference (aka 'Memory Leak')

BACK TO TOP

Use After Free

Risk

What might happen

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

Cause

How does it happen

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

General Recommendations

How to avoid it

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

---

Source Code Examples

CPP

Use of Variable after It was Freed

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

Use of Pointer to Local Variable That Was Freed On Return

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

void func2()

{

	int j;
	j = 5;
```

```
}

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

Use of Zero Initialized Pointer

Risk

What might happen

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

Cause

How does it happen

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

General Recommendations

How to avoid it

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

Source Code Examples

CPP

Explicit NULL Dereference

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

Implicit NULL Dereference

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

Java

Explicit Null Dereference

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

Unchecked Return Value

Risk

What might happen

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

Cause

How does it happen

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

General Recommendations

How to avoid it

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

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

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

Source Code Examples

CPP

Unchecked Memory Allocation

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

Safer Memory Allocation

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

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

Use of sizeof() on a Pointer Type

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

Description

Description Summary

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

Time of Introduction

- Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

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

Likelihood of Exploit

High

Demonstrative Examples

Example 1

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

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

(Bad Code)

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

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

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

(Good Code)

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

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

Example 2

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

(Bad Code)

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

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

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

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

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

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

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

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

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

(Attack)

```
pass5
passABCDEFGH
passWORD
```

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

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

Potential Mitigations

Phase: Implementation

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

Other Notes

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

Weakness Ordinalities

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

Relationships

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

Taxonomy Mappings

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

White Box Definitions

A weakness where code path has:

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

References

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

Content History

Submissions

Submission Date	Submitter	Organization	Source
	CLASP		Externally Mined

Modifications

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

BACK TO TOP

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
```

Reliance on DNS Lookups in a Decision

Risk

What might happen

Relying on reverse DNS records, without verifying domain ownership via cryptographic certificates or protocols, is not a sufficient authentication mechanism. Basing any security decisions on the registered hostname could allow an external attacker to control the application flow. The attacker could possibly perform restricted operations, bypass access controls, and even spoof the user's identity, inject a bogus hostname into the security log, and possibly other logic attacks.

Cause

How does it happen

The application performs a reverse DNS resolution, based on the remote IP address, and performs a security check based on the returned hostname. However, it is relatively easy to spoof DNS names, or cause them to be misreported, depending on the context of the specific environment. If the remote server is controlled by the attacker, it can be configured to report a bogus hostname. Additionally, the attacker could also spoof the hostname if she controls the associated DNS server, or by attacking the legitimate DNS server, or by poisoning the server's DNS cache, or by modifying unprotected DNS traffic to the server. Regardless of the vector, a remote attacker can alter the detected network address, faking the authentication details.

General Recommendations

How to avoid it

- Do not rely on DNS records, network addresses, or system hostnames as a form of authentication, or any other security-related decision.
- Do not perform reverse DNS resolution over an unprotected protocol without record validation.
- Implement a proper authentication mechanism, such as passwords, cryptographic certificates, or public key digital signatures.
- Consider using proposed protocol extensions to cryptographically protect DNS, e.g. DNSSEC (though note the limited support and other drawbacks).

Source Code Examples

Java

Using Reverse DNS as Authentication

```
private boolean isInternalEmployee(ServletRequest req) {
      boolean isCompany = false;

      String ip = req.getRemoteAddr();
      InetAddress address = InetAddress.getByName(ip);

      if (address.getHostName().endsWith(COMPANYNAME)) {
            isCompany = true;
    }
      return isCompany;
```

```
}
```

Verify Authenticated User's Identity

```
private boolean isInternalEmployee(ServletRequest req) {
      boolean isCompany = false;

      Principal user = req.getUserPrincipal();
      if (user != null) {
      if (user.getName().startsWith(COMPANYDOMAIN + "\\")) {
            isCompany = true;
         }
     }
      return isCompany;
}
```

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 Access Control (Authorization)

Weakness ID: 285 *(Weakness Class)* **Status:** Draft

Description

Description Summary

The software does not perform or incorrectly performs access control checks across all potential execution paths.

Extended Description

When access control checks are not applied consistently - or not at all - users are able to access data or perform actions that they should not be allowed to perform. This can lead to a wide range of problems, including information leaks, denial of service, and arbitrary code execution.

Alternate Terms

AuthZ:	"AuthZ" is typically used as an abbreviation of "authorization" within the web application security community. It is also distinct from "AuthC," which is an abbreviation of "authentication." The use of "Auth" as an abbreviation is discouraged, since it could be used for either authentication or authorization.

Time of Introduction

- Architecture and Design
- Implementation
- Operation

Applicable Platforms

Languages

Language-independent

Technology Classes

Web-Server: *(Often)*

Database-Server: *(Often)*

Modes of Introduction

A developer may introduce authorization weaknesses because of a lack of understanding about the underlying technologies. For example, a developer may assume that attackers cannot modify certain inputs such as headers or cookies.

Authorization weaknesses may arise when a single-user application is ported to a multi-user environment.

Common Consequences

Scope	Effect
Confidentiality	An attacker could read sensitive data, either by reading the data directly from a data store that is not properly restricted, or by accessing insufficiently-protected, privileged functionality to read the data.
Integrity	An attacker could modify sensitive data, either by writing the data directly to a data store that is not properly restricted, or by accessing insufficiently-protected, privileged functionality to write the data.
Integrity	An attacker could gain privileges by modifying or reading critical data directly, or by accessing insufficiently-protected, privileged functionality.

Likelihood of Exploit

High

Detection Methods

Automated Static Analysis

Automated static analysis is useful for detecting commonly-used idioms for authorization. A tool may be able to analyze related configuration files, such as .htaccess in Apache web servers, or detect the usage of commonly-used authorization libraries.

Generally, automated static analysis tools have difficulty detecting custom authorization schemes. In addition, the software's design may include some functionality that is accessible to any user and does not require an authorization check; an automated technique that detects the absence of authorization may report false positives.

Effectiveness: Limited

Automated Dynamic Analysis

Automated dynamic analysis may find many or all possible interfaces that do not require authorization, but manual analysis is required to determine if the lack of authorization violates business logic

Manual Analysis

This weakness can be detected using tools and techniques that require manual (human) analysis, such as penetration testing, threat modeling, and interactive tools that allow the tester to record and modify an active session.

Specifically, manual static analysis is useful for evaluating the correctness of custom authorization mechanisms.

Effectiveness: Moderate

These may be more effective than strictly automated techniques. This is especially the case with weaknesses that are related to design and business rules. However, manual efforts might not achieve desired code coverage within limited time constraints.

Demonstrative Examples

Example 1

The following program could be part of a bulletin board system that allows users to send private messages to each other. This program intends to authenticate the user before deciding whether a private message should be displayed. Assume that LookupMessageObject() ensures that the $id argument is numeric, constructs a filename based on that id, and reads the message details from that file. Also assume that the program stores all private messages for all users in the same directory.

(Bad Code)

Example Language: **Perl**

```
sub DisplayPrivateMessage {
my($id) = @_;
my $Message = LookupMessageObject($id);
print "From: " . encodeHTML($Message->{from}) . "<br>\n";
print "Subject: " . encodeHTML($Message->{subject}) . "\n";
print "<hr>\n";
print "Body: " . encodeHTML($Message->{body}) . "\n";
}

my $q = new CGI;
# For purposes of this example, assume that CWE-309 and
# CWE-523 do not apply.
if (! AuthenticateUser($q->param('username'), $q->param('password'))) {
ExitError("invalid username or password");
}

my $id = $q->param('id');
DisplayPrivateMessage($id);
```

While the program properly exits if authentication fails, it does not ensure that the message is addressed to the user. As a result, an authenticated attacker could provide any arbitrary identifier and read private messages that were intended for other users.

One way to avoid this problem would be to ensure that the "to" field in the message object matches the username of the authenticated user.

Observed Examples

Reference	Description
CVE-2009-3168	Web application does not restrict access to admin scripts, allowing authenticated users to reset administrative passwords.

CVE-2009-2960	Web application does not restrict access to admin scripts, allowing authenticated users to modify passwords of other users.
CVE-2009-3597	Web application stores database file under the web root with insufficient access control (CWE-219), allowing direct request.
CVE-2009-2282	Terminal server does not check authorization for guest access.
CVE-2009-3230	Database server does not use appropriate privileges for certain sensitive operations.
CVE-2009-2213	Gateway uses default "Allow" configuration for its authorization settings.
CVE-2009-0034	Chain: product does not properly interpret a configuration option for a system group, allowing users to gain privileges.
CVE-2008-6123	Chain: SNMP product does not properly parse a configuration option for which hosts are allowed to connect, allowing unauthorized IP addresses to connect.
CVE-2008-5027	System monitoring software allows users to bypass authorization by creating custom forms.
CVE-2008-7109	Chain: reliance on client-side security (CWE-602) allows attackers to bypass authorization using a custom client.
CVE-2008-3424	Chain: product does not properly handle wildcards in an authorization policy list, allowing unintended access.
CVE-2009-3781	Content management system does not check access permissions for private files, allowing others to view those files.
CVE-2008-4577	ACL-based protection mechanism treats negative access rights as if they are positive, allowing bypass of intended restrictions.
CVE-2008-6548	Product does not check the ACL of a page accessed using an "include" directive, allowing attackers to read unauthorized files.
CVE-2007-2925	Default ACL list for a DNS server does not set certain ACLs, allowing unauthorized DNS queries.
CVE-2006-6679	Product relies on the X-Forwarded-For HTTP header for authorization, allowing unintended access by spoofing the header.
CVE-2005-3623	OS kernel does not check for a certain privilege before setting ACLs for files.
CVE-2005-2801	Chain: file-system code performs an incorrect comparison (CWE-697), preventing defauls ACLs from being properly applied.
CVE-2001-1155	Chain: product does not properly check the result of a reverse DNS lookup because of operator precedence (CWE-783), allowing bypass of DNS-based access restrictions.

Potential Mitigations

Phase: Architecture and Design

Divide your application into anonymous, normal, privileged, and administrative areas. Reduce the attack surface by carefully mapping roles with data and functionality. Use role-based access control (RBAC) to enforce the roles at the appropriate boundaries.

Note that this approach may not protect against horizontal authorization, i.e., it will not protect a user from attacking others with the same role.

Phase: Architecture and Design

Ensure that you perform access control checks related to your business logic. These checks may be different than the access control checks that you apply to more generic resources such as files, connections, processes, memory, and database records. For example, a database may restrict access for medical records to a specific database user, but each record might only be intended to be accessible to the patient and the patient's doctor.

Phase: Architecture and Design

Strategy: Libraries or Frameworks

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness

easier to avoid.

For example, consider using authorization frameworks such as the JAAS Authorization Framework and the OWASP ESAPI Access Control feature.

Phase: Architecture and Design

For web applications, make sure that the access control mechanism is enforced correctly at the server side on every page. Users should not be able to access any unauthorized functionality or information by simply requesting direct access to that page.

One way to do this is to ensure that all pages containing sensitive information are not cached, and that all such pages restrict access to requests that are accompanied by an active and authenticated session token associated with a user who has the required permissions to access that page.

Phases: System Configuration; Installation

Use the access control capabilities of your operating system and server environment and define your access control lists accordingly. Use a "default deny" policy when defining these ACLs.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	254	Security Features	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Weakness Class	284	Access Control (Authorization) Issues	**Development Concepts (primary)699 Research Concepts (primary)1000**
ChildOf	Category	721	OWASP Top Ten 2007 Category A10 - Failure to Restrict URL Access	**Weaknesses in OWASP Top Ten (2007) (primary)629**
ChildOf	Category	723	OWASP Top Ten 2004 Category A2 - Broken Access Control	**Weaknesses in OWASP Top Ten (2004) (primary)711**
ChildOf	Category	753	2009 Top 25 - Porous Defenses	**Weaknesses in the 2009 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)750**
ChildOf	Category	803	2010 Top 25 - Porous Defenses	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
ParentOf	Weakness Variant	219	Sensitive Data Under Web Root	**Research Concepts (primary)1000**
ParentOf	Weakness Base	551	Incorrect Behavior Order: Authorization Before Parsing and Canonicalization	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Class	638	Failure to Use Complete Mediation	Research Concepts1000
ParentOf	Weakness Base	804	Guessable CAPTCHA	**Development Concepts (primary)699 Research Concepts (primary)1000**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
7 Pernicious Kingdoms			Missing Access Control
OWASP Top Ten 2007	A10	CWE More Specific	Failure to Restrict URL Access
OWASP Top Ten 2004	A2	CWE More Specific	Broken Access Control

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
1	Accessing Functionality Not Properly Constrained by ACLs	
13	Subverting Environment Variable Values	

17	Accessing, Modifying or Executing Executable Files
87	Forceful Browsing
39	Manipulating Opaque Client-based Data Tokens
45	Buffer Overflow via Symbolic Links
51	Poison Web Service Registry
59	Session Credential Falsification through Prediction
60	Reusing Session IDs (aka Session Replay)
77	Manipulating User-Controlled Variables
76	Manipulating Input to File System Calls
104	Cross Zone Scripting

References

NIST. "Role Based Access Control and Role Based Security". <http://csrc.nist.gov/groups/SNS/rbac/>.

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 4, "Authorization" Page 114; Chapter 6, "Determining Appropriate Access Control" Page 171. 2nd Edition. Microsoft. 2002.

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	7 Pernicious Kingdoms		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-15		Veracode	External
	Suggested OWASP Top Ten 2004 mapping		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Relationships, Other Notes, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences, Description, Likelihood of Exploit, Name, Other Notes, Potential Mitigations, References, Relationships		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Potential Mitigations		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Description, Related Attack Patterns		
2009-07-27	CWE Content Team	MITRE	Internal
	updated Relationships		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Type		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Demonstrative Examples, Detection Factors, Modes of Introduction, Observed Examples, Relationships		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Alternate Terms, Detection Factors, Potential Mitigations, References, Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Potential Mitigations		
Previous Entry Names			
Change Date	**Previous Entry Name**		
2009-01-12	Missing or Inconsistent Access Control		

BACK TO TOP

Incorrect Permission Assignment for Critical Resource

Weakness ID: 732 *(Weakness Class)* **Status:** Draft

Description

Description Summary

The software specifies permissions for a security-critical resource in a way that allows that resource to be read or modified by unintended actors.

Extended Description

When a resource is given a permissions setting that provides access to a wider range of actors than required, it could lead to the disclosure of sensitive information, or the modification of that resource by unintended parties. This is especially dangerous when the resource is related to program configuration, execution or sensitive user data.

Time of Introduction

- Architecture and Design
- Implementation
- Installation
- Operation

Applicable Platforms

Languages

Language-independent

Modes of Introduction

The developer may set loose permissions in order to minimize problems when the user first runs the program, then create documentation stating that permissions should be tightened. Since system administrators and users do not always read the documentation, this can result in insecure permissions being left unchanged.

The developer might make certain assumptions about the environment in which the software runs - e.g., that the software is running on a single-user system, or the software is only accessible to trusted administrators. When the software is running in a different environment, the permissions become a problem.

Common Consequences

Scope	Effect
Confidentiality	An attacker may be able to read sensitive information from the associated resource, such as credentials or configuration information stored in a file.
Integrity	An attacker may be able to modify critical properties of the associated resource to gain privileges, such as replacing a world-writable executable with a Trojan horse.
Availability	An attacker may be able to destroy or corrupt critical data in the associated resource, such as deletion of records from a database.

Likelihood of Exploit

Medium to High

Detection Methods

Automated Static Analysis

Automated static analysis may be effective in detecting permission problems for system resources such as files, directories, shared memory, device interfaces, etc. Automated techniques may be able to detect the use of library functions that modify permissions, then analyze function calls for arguments that contain potentially insecure values.

However, since the software's intended security policy might allow loose permissions for certain operations (such as publishing a file on a web server), automated static analysis may produce some false positives - i.e., warnings that do not have any security consequences or require any code changes.

When custom permissions models are used - such as defining who can read messages in a particular forum in a bulletin board system - these can be difficult to detect using automated static analysis. It may be possible to define custom signatures that

identify any custom functions that implement the permission checks and assignments.

Automated Dynamic Analysis

Automated dynamic analysis may be effective in detecting permission problems for system resources such as files, directories, shared memory, device interfaces, etc.

However, since the software's intended security policy might allow loose permissions for certain operations (such as publishing a file on a web server), automated dynamic analysis may produce some false positives - i.e., warnings that do not have any security consequences or require any code changes.

When custom permissions models are used - such as defining who can read messages in a particular forum in a bulletin board system - these can be difficult to detect using automated dynamic analysis. It may be possible to define custom signatures that identify any custom functions that implement the permission checks and assignments.

Manual Static Analysis

Manual static analysis may be effective in detecting the use of custom permissions models and functions. The code could then be examined to identifying usage of the related functions. Then the human analyst could evaluate permission assignments in the context of the intended security model of the software.

Manual Dynamic Analysis

Manual dynamic analysis may be effective in detecting the use of custom permissions models and functions. The program could then be executed with a focus on exercising code paths that are related to the custom permissions. Then the human analyst could evaluate permission assignments in the context of the intended security model of the software.

Fuzzing

Fuzzing is not effective in detecting this weakness.

Demonstrative Examples

Example 1

The following code sets the umask of the process to 0 before creating a file and writing "Hello world" into the file.

(Bad Code)

Example Language: **C**

```
#define OUTFILE "hello.out"

umask(0);
FILE *out;
/* Ignore CWE-59 (link following) for brevity */
out = fopen(OUTFILE, "w");
if (out) {
fprintf(out, "hello world!\n");
fclose(out);
}
```

After running this program on a UNIX system, running the "ls -l" command might return the following output:

(Result)

```
-rw-rw-rw- 1 username 13 Nov 24 17:58 hello.out
```

The "rw-rw-rw-" string indicates that the owner, group, and world (all users) can read the file and write to it.

Example 2

The following code snippet might be used as a monitor to periodically record whether a web site is alive. To ensure that the file can always be modified, the code uses chmod() to make the file world-writable.

(Bad Code)

Example Language: **Perl**

```
$fileName = "secretFile.out";

if (-e $fileName) {
chmod 0777, $fileName;
}
```

```
my $outFH;
if (! open($outFH, ">>$fileName")) {
ExitError("Couldn't append to $fileName: $!");
}
my $dateString = FormatCurrentTime();
my $status = IsHostAlive("cwe.mitre.org");
print $outFH "$dateString cwe status: $status!\n";
close($outFH);
```

The first time the program runs, it might create a new file that inherits the permissions from its environment. A file listing might look like:

(Result)

```
-rw-r--r-- 1 username 13 Nov 24 17:58 secretFile.out
```

This listing might occur when the user has a default umask of 022, which is a common setting. Depending on the nature of the file, the user might not have intended to make it readable by everyone on the system.

The next time the program runs, however - and all subsequent executions - the chmod will set the file's permissions so that the owner, group, and world (all users) can read the file and write to it:

(Result)

```
-rw-rw-rw- 1 username 13 Nov 24 17:58 secretFile.out
```

Perhaps the programmer tried to do this because a different process uses different permissions that might prevent the file from being updated.

Example 3

The following command recursively sets world-readable permissions for a directory and all of its children:

(Bad Code)

Example Language: **Shell**

```
chmod -R ugo+r DIRNAME
```

If this command is run from a program, the person calling the program might not expect that all the files under the directory will be world-readable. If the directory is expected to contain private data, this could become a security problem.

Observed Examples

Reference	Description
CVE-2009-3482	Anti-virus product sets insecure "Everyone: Full Control" permissions for files under the "Program Files" folder, allowing attackers to replace executables with Trojan horses.
CVE-2009-3897	Product creates directories with 0777 permissions at installation, allowing users to gain privileges and access a socket used for authentication.
CVE-2009-3489	Photo editor installs a service with an insecure security descriptor, allowing users to stop or start the service, or execute commands as SYSTEM.
CVE-2009-3289	Library function copies a file to a new target and uses the source file's permissions for the target, which is incorrect when the source file is a symbolic link, which typically has 0777 permissions.
CVE-2009-0115	Device driver uses world-writable permissions for a socket file, allowing attackers to inject arbitrary commands.
CVE-2009-1073	LDAP server stores a cleartext password in a world-readable file.
CVE-2009-0141	Terminal emulator creates TTY devices with world-writable permissions, allowing an attacker to write to the terminals of other users.

CVE-2008-0662	VPN product stores user credentials in a registry key with "Everyone: Full Control" permissions, allowing attackers to steal the credentials.
CVE-2008-0322	Driver installs its device interface with "Everyone: Write" permissions.
CVE-2009-3939	Driver installs a file with world-writable permissions.
CVE-2009-3611	Product changes permissions to 0777 before deleting a backup; the permissions stay insecure for subsequent backups.
CVE-2007-6033	Product creates a share with "Everyone: Full Control" permissions, allowing arbitrary program execution.
CVE-2007-5544	Product uses "Everyone: Full Control" permissions for memory-mapped files (shared memory) in inter-process communication, allowing attackers to tamper with a session.
CVE-2005-4868	Database product uses read/write permissions for everyone for its shared memory, allowing theft of credentials.
CVE-2004-1714	Security product uses "Everyone: Full Control" permissions for its configuration files.
CVE-2001-0006	"Everyone: Full Control" permissions assigned to a mutex allows users to disable network connectivity.
CVE-2002-0969	Chain: database product contains buffer overflow that is only reachable through a .ini configuration file - which has "Everyone: Full Control" permissions.

Potential Mitigations

Phase: Implementation

When using a critical resource such as a configuration file, check to see if the resource has insecure permissions (such as being modifiable by any regular user), and generate an error or even exit the software if there is a possibility that the resource could have been modified by an unauthorized party.

Phase: Architecture and Design

Divide your application into anonymous, normal, privileged, and administrative areas. Reduce the attack surface by carefully defining distinct user groups, privileges, and/or roles. Map these against data, functionality, and the related resources. Then set the permissions accordingly. This will allow you to maintain more fine-grained control over your resources.

Phases: Implementation; Installation

During program startup, explicitly set the default permissions or umask to the most restrictive setting possible. Also set the appropriate permissions during program installation. This will prevent you from inheriting insecure permissions from any user who installs or runs the program.

Phase: System Configuration

For all configuration files, executables, and libraries, make sure that they are only readable and writable by the software's administrator.

Phase: Documentation

Do not suggest insecure configuration changes in your documentation, especially if those configurations can extend to resources and other software that are outside the scope of your own software.

Phase: Installation

Do not assume that the system administrator will manually change the configuration to the settings that you recommend in the manual.

Phase: Testing

Use tools and techniques that require manual (human) analysis, such as penetration testing, threat modeling, and interactive tools that allow the tester to record and modify an active session. These may be more effective than strictly automated techniques. This is especially the case with weaknesses that are related to design and business rules.

Phase: Testing

Use monitoring tools that examine the software's process as it interacts with the operating system and the network. This technique is useful in cases when source code is unavailable, if the software was not developed by you, or if you want to verify that the build phase did not introduce any new weaknesses. Examples include debuggers that directly attach to the running process; system-call tracing utilities such as truss (Solaris) and strace (Linux); system activity monitors such as FileMon, RegMon, Process Monitor, and other Sysinternals utilities (Windows); and sniffers and protocol analyzers that monitor network traffic.

Attach the monitor to the process and watch for library functions or system calls on OS resources such as files, directories, and shared memory. Examine the arguments to these calls to infer which permissions are being used.

Note that this technique is only useful for permissions issues related to system resources. It is not likely to detect application-level business rules that are related to permissions, such as if a user of a blog system marks a post as "private," but the blog system inadvertently marks it as "public."

Phases: Testing; System Configuration

Ensure that your software runs properly under the Federal Desktop Core Configuration (FDCC) or an equivalent hardening configuration guide, which many organizations use to limit the attack surface and potential risk of deployed software.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	275	Permission Issues	**Development Concepts (primary)699**
ChildOf	Weakness Class	668	Exposure of Resource to Wrong Sphere	**Research Concepts (primary)1000**
ChildOf	Category	753	2009 Top 25 - Porous Defenses	**Weaknesses in the 2009 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)750**
ChildOf	Category	803	2010 Top 25 - Porous Defenses	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
RequiredBy	Compound Element: Composite	689	Permission Race Condition During Resource Copy	Research Concepts1000
ParentOf	Weakness Variant	276	Incorrect Default Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	277	Insecure Inherited Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	278	Insecure Preserved Inherited Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	279	Incorrect Execution-Assigned Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Base	281	Improper Preservation of Permissions	**Research Concepts (primary)1000**

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
232	Exploitation of Privilege/Trust	
1	Accessing Functionality Not Properly Constrained by ACLs	
17	Accessing, Modifying or Executing Executable Files	
60	Reusing Session IDs (aka Session Replay)	
61	Session Fixation	
62	Cross Site Request Forgery (aka Session Riding)	
122	Exploitation of Authorization	
180	Exploiting Incorrectly Configured Access Control Security Levels	
234	Hijacking a privileged process	

References

Mark Dowd, John McDonald and Justin Schuh. "The Art of Software Security Assessment". Chapter 9, "File Permissions." Page 495.. 1st Edition. Addison Wesley. 2006.

John Viega and Gary McGraw. "Building Secure Software". Chapter 8, "Access Control." Page 194.. 1st Edition. Addison-Wesley. 2002.

Maintenance Notes

The relationships between privileges, permissions, and actors (e.g. users and groups) need further refinement within the Research view. One complication is that these concepts apply to two different pillars, related to control of resources (CWE-664) and protection mechanism failures (CWE-396).

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
2008-09-08			Internal CWE Team
	new weakness-focused entry for Research view.		
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2009-01-12	CWE Content Team	MITRE	Internal
	updated Description, Likelihood of Exploit, Name, Potential Mitigations, Relationships		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Potential Mitigations, Related Attack Patterns		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Name		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Demonstrative Examples, Detection Factors, Modes of Introduction, Observed Examples, Potential Mitigations, References		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Potential Mitigations, Related Attack Patterns		
Previous Entry Names			
Change Date	**Previous Entry Name**		
2009-01-12	Insecure Permission Assignment for Resource		
2009-05-27	Insecure Permission Assignment for Critical Resource		

BACK TO TOP

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--;
       }
    }
}
```

Scanned Languages

Language	Hash Number	Change Date
CPP	4541647240435660	6/19/2024
Common	0105849645654507	6/19/2024