wcc Scan Report

Project Name	wcc
Scan Start	Friday, June 21, 2024 10:28:03 PM
Preset	Checkmarx Default
Scan Time	00h:02m:26s
Lines Of Code Scanned	15123
Files Scanned	8
Report Creation Time	Friday, June 21, 2024 10:31:54 PM
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038
Team	CxServer
Checkmarx Version	8.7.0
Scan Type	Full
Source Origin	LocalPath
Density	4/100 (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

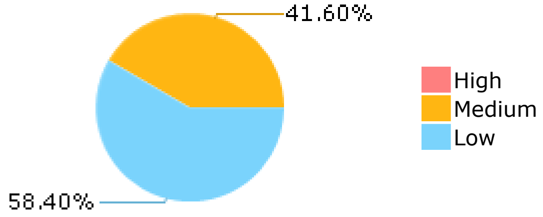
Results Limit

Results limit per query was set to 50

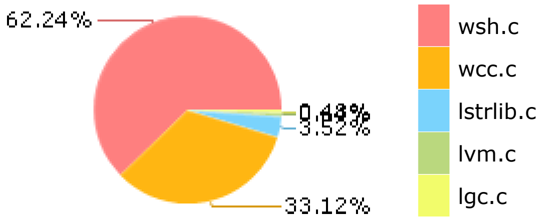
Selected Queries

Selected queries are listed in Result Summary

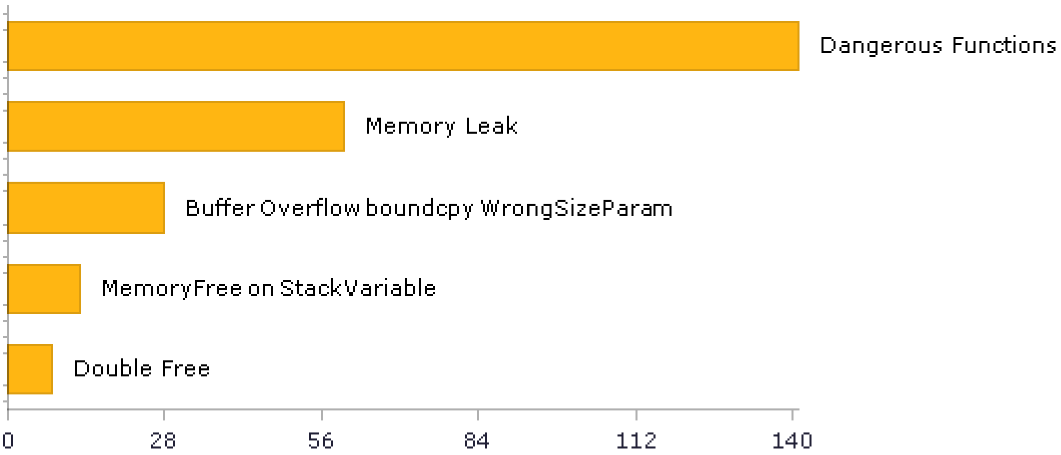
Result Summary



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	126	58
A2-Broken Authentication	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	89	89
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	141	141
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	141	141
A10-Unvalidated Redirects and Forwards	USERS BROWSERS	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	AFFECTED DATA AND FUNCTIONS	0	0

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

Scan Summary - PCI DSS v3.2

Category	Issues Found	Best Fix Locations
PCI DSS (3.2) - 6.5.1 - Injection flaws - particularly SQL injection	1	1
PCI DSS (3.2) - 6.5.2 - Buffer overflows	39	34
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.	1	1
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.	2	2
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.	70	56
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.	88	88
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.	2	2

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

Scan Summary - NIST SP 800-53

Category	Issues Found	Best Fix Locations
AC-12 Session Termination (P2)	0	0
AC-3 Access Enforcement (P1)	159	145
AC-4 Information Flow Enforcement (P1)	0	0
AC-6 Least Privilege (P1)	0	0
AU-9 Protection of Audit Information (P1)	0	0
CM-6 Configuration Settings (P2)	0	0
IA-5 Authenticator Management (P1)	0	0
IA-6 Authenticator Feedback (P2)	0	0
IA-8 Identification and Authentication (Non-Organizational Users) (P1)	0	0
SC-12 Cryptographic Key Establishment and Management (P1)	0	0
SC-13 Cryptographic Protection (P1)	0	0
SC-17 Public Key Infrastructure Certificates (P1)	0	0
SC-18 Mobile Code (P2)	0	0
SC-23 Session Authenticity (P1)*	0	0
SC-28 Protection of Information at Rest (P1)	0	0
SC-4 Information in Shared Resources (P1)	0	0
SC-5 Denial of Service Protection (P1)*	150	88
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*	14	8
SI-11 Error Handling (P2)*	73	73
SI-15 Information Output Filtering (P0)	0	0
SI-16 Memory Protection (P1)	9	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

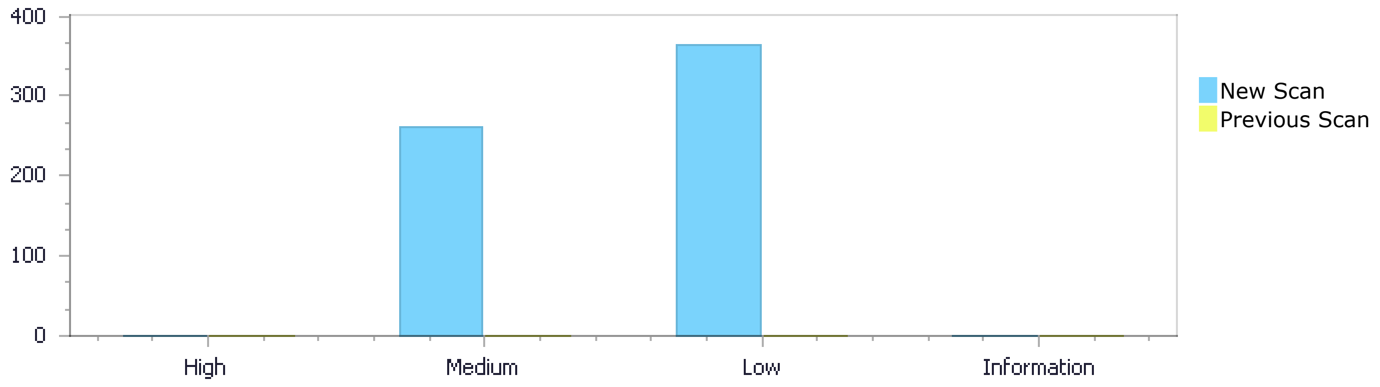
Category	Issues Found	Best Fix Locations
Must audit	0	0
Check	0	0
Optional	0	0

Results Distribution By Status

First scan of the project

	High	Medium	Low	Information	Total
New Issues	0	260	365	0	625
Recurrent Issues	0	0	0	0	0
Total	0	260	365	0	625

Fixed Issues	0	0	0	0	0



Results Distribution By State

	High	Medium	Low	Information	Total
Confirmed	0	0	0	0	0
Not Exploitable	0	0	0	0	0
To Verify	0	260	365	0	625
Urgent	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0
Total	0	260	365	0	625

Result Summary

Vulnerability Type	Occurrences	Severity
Dangerous Functions	141	Medium
Memory Leak	60	Medium
Buffer Overflow boundcpy WrongSizeParam	28	Medium
MemoryFree on StackVariable	13	Medium
Double Free	8	Medium

Buffer Overflow AddressOfLocalVarReturned	3	Medium
Integer Overflow	2	Medium
Stored Buffer Overflow boundcpy	2	Medium
Wrong Size t Allocation	2	Medium
Use After Free	1	Medium
Improper Resource Access Authorization	88	Low
NULL Pointer Dereference	84	Low
Unchecked Return Value	73	Low
Exposure of System Data to Unauthorized Control Sphere	70	Low
TOCTOU	17	Low
Use of Sizeof On a Pointer Type	17	Low
Heuristic 2nd Order Buffer Overflow read	6	Low
Unchecked Array Index	3	Low
Arithmenic Operation On Boolean	2	Low
Inconsistent Implementations	2	Low
Incorrect Permission Assignment For Critical Resources	1	Low
Potential Off by One Error in Loops	1	Low
Potential Precision Problem	1	Low

10 Most Vulnerable Files

High and Medium Vulnerabilities

File Name	Issues Found
wcc/wsh.c	145
wcc/wcc.c	93
wcc/lstrlib.c	18
wcc/lvm.c	4

Scan Results Details

Dangerous Functions

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

Categories

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

Description

Dangerous Functions\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=238
Status	New

The dangerous function, memcpy, was found in use at line 120 in wcc/lstrlib.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c
Line	133	133
Object	memcpy	memcpy

Code Snippet	
File Name	wcc/lstrlib.c
Method	static int str_rep (lua_State *L) {

```
....
133.        memcpy(p, s, l * sizeof(char)); p += l;
```

Dangerous Functions\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=239
Status	New

The dangerous function, memcpy, was found in use at line 120 in wcc/lstrlib.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c
Line	135	135

Object	memcpy	memcpy

Code Snippet
File Name wcc/lstrlib.c
Method static int str_rep (lua_State *L) {

```
....
135.           memcpy(p, sep, lsep * sizeof(char));
```

Dangerous Functions\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=240
Status	New

The dangerous function, memcpy, was found in use at line 120 in wcc/lstrlib.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c
Line	139	139
Object	memcpy	memcpy

Code Snippet
File Name wcc/lstrlib.c
Method static int str_rep (lua_State *L) {

```
....
139.       memcpy(p, s, l * sizeof(char));  /* last copy (not followed by
separator) */
```

Dangerous Functions\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=241
Status	New

The dangerous function, memcpy, was found in use at line 949 in wcc/lstrlib.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c
Line	964	964
Object	memcpy	memcpy

Code Snippet	
File Name	wcc/lstrlib.c
Method	static const char *scanformat (lua_State *L, const char *strfrmt, char *form) {

```
....
964.    memcpy(form, strfrmt, ((p - strfrmt) + 1) * sizeof(char));
```

Dangerous Functions\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=242
Status	New

The dangerous function, memcpy, was found in use at line 446 in wcc/lvm.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/lvm.c	wcc/lvm.c
Line	450	450
Object	memcpy	memcpy

Code Snippet	
File Name	wcc/lvm.c
Method	static void copy2buff (StkId top, int n, char *buff) {

```
....
450.      memcpy(buff + tl, svalue(top - n), l * sizeof(char));
```

Dangerous Functions\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=243
Status	New

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

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	464	464
Object	memcpy	memcpy

Code Snippet	
File Name	wcc/wcc.c

Method	void add_symaddr(ctx_t * ctx, const char *name, int addr, char symclass)

```
....
464.    memcpy(globalstrtab + globalstrtablen, sa->name, strlen(sa-
>name) + 1);
```

Dangerous Functions\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=244
Status	New

The dangerous function, memcpy, was found in use at line 1129 in wcc/wcc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1172	1172
Object	memcpy	memcpy

Code Snippet

File Name	wcc/wcc.c
Method	static unsigned int rd_phdrs(ctx_t * ctx)

```
....
1172.       memcpy(ms, phdr, sizeof(Elf_Phdr));
```

Dangerous Functions\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=245
Status	New

The dangerous function, memcpy, was found in use at line 1725 in wcc/wcc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1734	1734
Object	memcpy	memcpy

Code Snippet

File Name	wcc/wcc.c
Method	unsigned int append_sym(Elf_Sym * s)

```
....
1734.    memcpy(globalsymtab + globalsymtablen, s, sizeof(Elf_Sym));
```

Dangerous Functions\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=246
Status	New

The dangerous function, memcpy, was found in use at line 1745 in wcc/wcc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1755	1755
Object	memcpy	memcpy

Code Snippet

File Name	wcc/wcc.c
Method	unsigned int append_strtab(char *str)

```
....
1755.    memcpy(globalstrtab + globalstrtablen, str, strlen(str) + 1);
```

Dangerous Functions\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=247
Status	New

The dangerous function, memcpy, was found in use at line 1868 in wcc/wcc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1902	1902
Object	memcpy	memcpy

Code Snippet

File Name	wcc/wcc.c
Method	static unsigned int write_shdrs(ctx_t * ctx)

```
....
1902.        memcpy(ctx->strndx + ctx->strndx_len, s->name, strlen(s->name) + 1);       // do copy the final "\x00"
```

Dangerous Functions\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=248
Status	New

The dangerous function, memcpy, was found in use at line 1868 in wcc/wcc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1918	1918
Object	memcpy	memcpy

Code Snippet
File Name wcc/wcc.c
Method static unsigned int write_shdrs(ctx_t * ctx)

```
....
1918.     memcpy(ctx->strndx + ctx->strndx_len, ".rela.all", 10);
```

Dangerous Functions\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=249
Status	New

The dangerous function, memcpy, was found in use at line 1868 in wcc/wcc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1947	1947
Object	memcpy	memcpy

Code Snippet
File Name wcc/wcc.c
Method static unsigned int write_shdrs(ctx_t * ctx)

```
....
1947.    memcpy(ctx->strndx + ctx->strndx_len, ".strtab", 8);
```

Dangerous Functions\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=250
Status	New

The dangerous function, memcpy, was found in use at line 1868 in wcc/wcc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1977	1977
Object	memcpy	memcpy

Code Snippet	
File Name	wcc/wcc.c
Method	static unsigned int write_shdrs(ctx_t * ctx)

```
....
1977.    memcpy(ctx->strndx + ctx->strndx_len, ".symtab", 8);
```

Dangerous Functions\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=251
Status	New

The dangerous function, memcpy, was found in use at line 1868 in wcc/wcc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2007	2007
Object	memcpy	memcpy

Code Snippet	
File Name	wcc/wcc.c
Method	static unsigned int write_shdrs(ctx_t * ctx)

```
....
2007.    memcpy(ctx->strndx + ctx->strndx_len, ".shstrtab", 10);
```

Dangerous Functions\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=252
Status	New

The dangerous function, memcpy, was found in use at line 2042 in wcc/wcc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2057	2057
Object	memcpy	memcpy

Code Snippet

File Name	wcc/wcc.c
Method	static int mk_ehdr(ctx_t * ctx)

```
....
2057.    memcpy(e->e_ident, "\x7f\x45\x4c\x46\x02\x01\x01", 7);
```

Dangerous Functions\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=253
Status	New

The dangerous function, memcpy, was found in use at line 2443 in wcc/wcc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2472	2472
Object	memcpy	memcpy

Code Snippet

File Name	wcc/wcc.c
Method	int craft_section(ctx_t * ctx, msec_t * m)

```
....
2472.    memcpy(ctx->strndx + ctx->strndx_len, s->name, strlen(s-
>name));
```

Dangerous Functions\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=254
Status	New

The dangerous function, memcpy, was found in use at line 2605 in wcc/wcc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2613	2613
Object	memcpy	memcpy

Code Snippet

File Name	wcc/wcc.c
Method	int save_dynstr(ctx_t * ctx, GElf_Shdr shdr, char *binary)

```
....
2613.    memcpy(globalstrtab + globalstrtablen, binary + shdr.sh_offset,
```

Dangerous Functions\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=255
Status	New

The dangerous function, memcpy, was found in use at line 2620 in wcc/wcc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2628	2628
Object	memcpy	memcpy

Code Snippet

File Name	wcc/wcc.c
Method	int save_dynsym(ctx_t * ctx, GElf_Shdr shdr, char *binary)

```
....
2628.     memcpy(globalsymtab + globalsymtablen,
```

Dangerous Functions\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=256
Status	New

The dangerous function, memcpy, was found in use at line 2672 in wcc/wcc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2694	2694
Object	memcpy	memcpy

Code Snippet
File Name wcc/wcc.c
Method int append_reloc(Elf_Rela * r)

```
....
2694.     memcpy(globalreloc + globalreloclen, r, sizeof(Elf_Rela));
```

Dangerous Functions\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=257
Status	New

The dangerous function, memcpy, was found in use at line 2701 in wcc/wcc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2722	2722
Object	memcpy	memcpy

Code Snippet
File Name wcc/wcc.c
Method int save_global_import(ctx_t * ctx, char *sname, msec_t * sec, Elf_Rela * r, unsigned int sindex)

```
....
2722.     memcpy(rnew, r, sizeof(Elf_Rela));
```

Dangerous Functions\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=258
Status	New

The dangerous function, memcpy, was found in use at line 2755 in wcc/wcc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2768	2768
Object	memcpy	memcpy

Code Snippet

File Name	wcc/wcc.c
Method	int save_reloc(ctx_t * ctx, Elf_Rela * r, unsigned int sindex, int has_addend)

```
....
2768.     memcpy(rout, r, sizeof(Elf_Rela));
```

Dangerous Functions\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=259
Status	New

The dangerous function, memcpy, was found in use at line 2990 in wcc/wcc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3171	3171
Object	memcpy	memcpy

Code Snippet

File Name	wcc/wcc.c
Method	static int create_text_data_reloc(ctx_t * ctx, cs_insn * ins, msec_t * m,

```
....
3171.           memcpy(t->data + r->r_offset, &soff, 4);
```

Dangerous Functions\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=260
Status	New

The dangerous function, memcpy, was found in use at line 3556 in wcc/wsh.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3617	3617
Object	memcpy	memcpy

Code Snippet
File Name wcc/wsh.c
Method void sighandler(int signal, siginfo_t * s, void *ptr)

```
....
3617.                memcpy(wsh->errcontext, u, sizeof(ucontext_t));
```

Dangerous Functions\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=261
Status	New

The dangerous function, memcpy, was found in use at line 4118 in wcc/wsh.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	4147	4147
Object	memcpy	memcpy

Code Snippet
File Name wcc/wsh.c
Method int grepptr(lua_State * L)

```
....
4147.        memcpy(pattern, &p, patternsz);
```

Dangerous Functions\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=262
Status	New

The dangerous function, memcpy, was found in use at line 4299 in wcc/wsh.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	4310	4310
Object	memcpy	memcpy

Code Snippet

File Name	wcc/wsh.c
Method	int priv_memcpy(lua_State * L)

```
....
4310.        ret = memcpy(arg1, arg2, arg3);
```

Dangerous Functions\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=263
Status	New

The dangerous function, sprintf, was found in use at line 1603 in wcc/wcc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1643	1643
Object	sprintf	sprintf

Code Snippet

File Name	wcc/wcc.c
Method	int fixup_strtab_and_symtab(ctx_t * ctx)

```
....
1643.          sprintf(globalstrtab + globalstrtablen, "old_%s", sname);
```

Dangerous Functions\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=264
Status	New

The dangerous function, sprintf, was found in use at line 2348 in wcc/wcc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2364	2364
Object	sprintf	sprintf

Code Snippet

File Name	wcc/wcc.c
Method	int open_target(ctx_t * ctx)

```
....
2364.      sprintf(newname, "a.out");
```

Dangerous Functions\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=265
Status	New

The dangerous function, sprintf, was found in use at line 3671 in wcc/wcc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3676	3676
Object	sprintf	sprintf

Code Snippet

File Name	wcc/wcc.c
Method	int print_maps(void)

```
....
3676.    sprintf(cmd, "cat /proc/%u/maps", getpid());
```

Dangerous Functions\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=266
Status	New

The dangerous function, sscanf, was found in use at line 1853 in wcc/wsh.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1877	1877
Object	sscanf	sscanf

Code Snippet
File Name   wcc/wsh.c
Method   int prototypes(lua_State * L)

```
....
1877.             sscanf(line, "%10s %200s %200s %20s %200s %20s", l-
>key.ttype, l->key.tlib, l->key.tfunction, l->key.targ, l->key.tvalue,
l->toffset);
```

Dangerous Functions\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=267
Status	New

The dangerous function, strcat, was found in use at line 560 in wcc/wsh.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	569	569
Object	strcat	strcat

Code Snippet
File Name   wcc/wsh.c
Method   char *decode_flags(unsigned int flags)

```
....
569.                strcat(message, "r");
```

Dangerous Functions\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=268
Status	New

The dangerous function, strcat, was found in use at line 560 in wcc/wsh.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	571	571
Object	strcat	strcat

Code Snippet
File Name wcc/wsh.c
Method char *decode_flags(unsigned int flags)

```
....
571.                strcat(message, "-");
```

Dangerous Functions\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=269
Status	New

The dangerous function, strcat, was found in use at line 560 in wcc/wsh.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	574	574
Object	strcat	strcat

Code Snippet
File Name wcc/wsh.c
Method char *decode_flags(unsigned int flags)

```
....
574.                    strcat(message, "w");
```

Dangerous Functions\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=270
Status	New

The dangerous function, strcat, was found in use at line 560 in wcc/wsh.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	576	576
Object	strcat	strcat

Code Snippet
File Name wcc/wsh.c
Method char *decode_flags(unsigned int flags)

```
....
576.                    strcat(message, "-");
```

Dangerous Functions\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=271
Status	New

The dangerous function, strcat, was found in use at line 560 in wcc/wsh.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	579	579
Object	strcat	strcat

Code Snippet
File Name wcc/wsh.c
Method char *decode_flags(unsigned int flags)

```
....
579.                strcat(message, "x");
```

Dangerous Functions\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=272
Status	New

The dangerous function, strcat, was found in use at line 560 in wcc/wsh.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	581	581
Object	strcat	strcat

Code Snippet
File Name wcc/wsh.c
Method char *decode_flags(unsigned int flags)

```
....
581.                strcat(message, "-");
```

Dangerous Functions\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=273
Status	New

The dangerous function, strcat, was found in use at line 2485 in wcc/wsh.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	2506	2506
Object	strcat	strcat

Code Snippet
File Name wcc/wsh.c
Method int luabuff_append(char *cmd){

```
....
2506.        strcat(wsh->luabuff, cmd);
```

Dangerous Functions\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=274
Status	New

The dangerous function, strcat, was found in use at line 4342 in wcc/wsh.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	4352	4352
Object	strcat	strcat

Code Snippet

File Name	wcc/wsh.c
Method	int priv_strcat(lua_State * L)

```
....
4352.        ret = strcat(arg1, arg2);
```

Dangerous Functions\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=275
Status	New

The dangerous function, strcpy, was found in use at line 974 in wcc/lstrlib.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c
Line	978	978
Object	strcpy	strcpy

Code Snippet

File Name	wcc/lstrlib.c
Method	static void addlenmod (char *form, const char *lenmod) {

```
....
978.     strcpy(form + l - 1, lenmod);
```

Dangerous Functions\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=276
Status	New

The dangerous function, strcpy, was found in use at line 4321 in wcc/wsh.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	4331	4331
Object	strcpy	strcpy

Code Snippet
File Name wcc/wsh.c
Method int priv_strcpy(lua_State * L)

```
....
4331.         ret = strcpy(arg1, arg2);
```

Dangerous Functions\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=277
Status	New

The dangerous function, strlen, was found in use at line 592 in wcc/lstrlib.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c
Line	597	597
Object	strlen	strlen

Code Snippet
File Name wcc/lstrlib.c
Method static int nospecials (const char *p, size_t l) {

```
....
597.        upto += strlen(p + upto) + 1;   /* may have more after \0 */
```

Dangerous Functions\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=278
Status	New

The dangerous function, strlen, was found in use at line 974 in wcc/lstrlib.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c
Line	975	975
Object	strlen	strlen

Code Snippet

File Name	wcc/lstrlib.c
Method	static void addlenmod (char *form, const char *lenmod) {

```
....
975.     size_t l = strlen(form);
```

Dangerous Functions\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=279
Status	New

The dangerous function, strlen, was found in use at line 974 in wcc/lstrlib.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c
Line	976	976
Object	strlen	strlen

Code Snippet

File Name	wcc/lstrlib.c
Method	static void addlenmod (char *form, const char *lenmod) {

```
....
976.     size_t lm = strlen(lenmod);
```

Dangerous Functions\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=280
Status	New

The dangerous function, strlen, was found in use at line 984 in wcc/lstrlib.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c
Line	1037	1037
Object	strlen	strlen

Code Snippet
File Name wcc/lstrlib.c
Method static int str_format (lua_State *L) {

```
....
1037.             luaL_argcheck(L, l == strlen(s), arg, "string
contains zeros");
```

Dangerous Functions\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=281
Status	New

The dangerous function, strlen, was found in use at line 1297 in wcc/lstrlib.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c
Line	1369	1369
Object	strlen	strlen

Code Snippet
File Name wcc/lstrlib.c
Method static int str_pack (lua_State *L) {

```
....
1369.          luaL_argcheck(L, strlen(s) == len, arg, "string contains 
zeros");
```

Dangerous Functions\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=282
Status	New

The dangerous function, strlen, was found in use at line 1445 in wcc/lstrlib.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c
Line	1493	1493
Object	strlen	strlen

Code Snippet

File Name wcc/lstrlib.c

Method static int str_unpack (lua_State *L) {

```
....
1493.          size_t len = (int)strlen(data + pos);
```

Dangerous Functions\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=283
Status	New

The dangerous function, strlen, was found in use at line 234 in wcc/lvm.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	wcc/lvm.c	wcc/lvm.c
Line	244	244
Object	strlen	strlen

Code Snippet

File Name wcc/lvm.c

Method static int l_strcmp (const TString *ls, const TString *rs) {

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

Dangerous Functions\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=284
Status	New

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

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	458	458
Object	strlen	strlen

Code Snippet
File Name wcc/wcc.c
Method void add_symaddr(ctx_t * ctx, const char *name, int addr, char symclass)

```
....
458.      globalstrtab = calloc(1, strlen(sa->name) + 3);
```

Dangerous Functions\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=285
Status	New

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

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	462	462
Object	strlen	strlen

Code Snippet
File Name wcc/wcc.c
Method void add_symaddr(ctx_t * ctx, const char *name, int addr, char symclass)

```
....
462.          realloc(globalstrtab, globalstrtablen + strlen(sa->name) +
2);
```

Dangerous Functions\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=286
Status	New

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

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	464	464
Object	strlen	strlen

Code Snippet

File Name	wcc/wcc.c
Method	void add_symaddr(ctx_t * ctx, const char *name, int addr, char symclass)

```
....
464.     memcpy(globalstrtab + globalstrtablen, sa->name, strlen(sa-
>name) + 1);
```

Dangerous Functions\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=287
Status	New

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

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	466	466
Object	strlen	strlen

Code Snippet

File Name	wcc/wcc.c
Method	void add_symaddr(ctx_t * ctx, const char *name, int addr, char symclass)

```
....
466.    globalstrtablen += strlen(sa->name) + 1;
```

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=1050048&projectid=50038&pathid=387
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	656	656
Object	pflags	pflags

Code Snippet
File Name wcc/wsh.c
Method int phdr_callback(struct dl_phdr_info *info, size_t size, void *data)

```
....
656.            pflags = p ? decode_flags(p->p_flags) : 0;
```

Memory Leak\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=388
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	656	656
Object	decode_flags	decode_flags

Code Snippet
File Name wcc/wsh.c
Method int phdr_callback(struct dl_phdr_info *info, size_t size, void *data)

```
....
656.                pflags = p ? decode_flags(p->p_flags) : 0;
```

Memory Leak\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=389
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1166	1166
Object	ms	ms

Code Snippet
File Name wcc/wcc.c
Method static unsigned int rd_phdrs(ctx_t * ctx)

```
....
1166.       mseg_t *ms = calloc(1, sizeof(mseg_t));
```

Memory Leak\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=390
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1241	1241
Object	phdr	phdr

Code Snippet
File Name wcc/wcc.c
Method static unsigned int write_phdrs(ctx_t * ctx)

```
....
1241.     Elf_Phdr *phdr = calloc(1, sizeof(Elf_Phdr));
```

Memory Leak\Path 5:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=391
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2519	2519
Object	ms	ms

Code Snippet

File Name	wcc/wcc.c
Method	static int read_section(ctx_t * ctx, asection * s)

```
....
2519.     msec_t *ms = calloc(1, sizeof(msec_t));
```

Memory Leak\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=392
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	443	443
Object	sa	sa

Code Snippet

File Name	wcc/wcc.c
Method	void add_symaddr(ctx_t * ctx, const char *name, int addr, char symclass)

```
....
443.     sa = (struct symaddr *) malloc(sizeof(struct symaddr));
```

Memory Leak\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=393
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	445	445

Object	name	name

Code Snippet
File Name wcc/wcc.c
Method void add_symaddr(ctx_t * ctx, const char *name, int addr, char symclass)

```
....
445.     sa->name = strdup(name);
```

Memory Leak\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=394
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	458	458
Object	globalstrtab	globalstrtab

Code Snippet
File Name wcc/wcc.c
Method void add_symaddr(ctx_t * ctx, const char *name, int addr, char symclass)

```
....
458.       globalstrtab = calloc(1, strlen(sa->name) + 3);
```

Memory Leak\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=395
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	472	472
Object	globalsymtab	globalsymtab

Code Snippet
File Name wcc/wcc.c
Method void add_symaddr(ctx_t * ctx, const char *name, int addr, char symclass)

```
....
472.        globalsymtab = calloc(1, sizeof(Elf_Sym) * 2);
```

Memory Leak\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=396
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1144	1144
Object	p	p

Code Snippet
File Name wcc/wcc.c
Method static unsigned int rd_phdrs(ctx_t * ctx)

```
....
1144.     p = calloc(1, sb.st_size);
```

Memory Leak\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=397
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1308	1308
Object	ms	ms

Code Snippet
File Name wcc/wcc.c
Method msec_t *mk_section(void)

```
....
1308.     ms = calloc(1, sizeof(msec_t));
```

Memory Leak\Path 12:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=398
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1314	1314
Object	s_elf	s_elf

Code Snippet

File Name	wcc/wcc.c
Method	msec_t *mk_section(void)

```
....
1314.     ms->s_elf = calloc(1, sizeof(Elf_Shdr));
```

Memory Leak\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=399
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1728	1728
Object	globalsymtab	globalsymtab

Code Snippet

File Name	wcc/wcc.c
Method	unsigned int append_sym(Elf_Sym * s)

```
....
1728.      globalsymtab = calloc(1, sizeof(Elf_Sym) * 2);
```

Memory Leak\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=400
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1750	1750

Object	globalstrtab	globalstrtab

Code Snippet
File Name wcc/wcc.c
Method unsigned int append_strtab(char *str)

```
....
1750.       globalstrtab = calloc(1, strlen(str) + 3);
```

Memory Leak\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=401
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2046	2046
Object	e	e

Code Snippet
File Name wcc/wcc.c
Method static int mk_ehdr(ctx_t * ctx)

```
....
2046.     e = calloc(1, sizeof(Elf_Ehdr));
```

Memory Leak\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=402
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2363	2363
Object	newname	newname

Code Snippet
File Name wcc/wcc.c
Method int open_target(ctx_t * ctx)

```
....
2363.       newname = calloc(1, strlen(ctx->binname) + 20);
```

Memory Leak\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=403
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2516	2516
Object	buf	buf

Code Snippet

File Name	wcc/wcc.c
Method	static int read_section(ctx_t * ctx, asection * s)

```
....
2516.     buf = calloc(1, wantedsz);
```

Memory Leak\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=404
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2525	2525
Object	s_elf	s_elf

Code Snippet

File Name	wcc/wcc.c
Method	static int read_section(ctx_t * ctx, asection * s)

```
....
2525.     ms->s_elf = calloc(1, sizeof(Elf_Shdr));
```

Memory Leak\Path 19:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=405
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2555	2555
Object	name	name

Code Snippet
File Name wcc/wcc.c
Method static int read_section(ctx_t * ctx, asection * s)

```
....
2555.     ms->name = strdup(s->name);
```

Memory Leak\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=406
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2608	2608
Object	globalstrtab	globalstrtab

Code Snippet
File Name wcc/wcc.c
Method int save_dynstr(ctx_t * ctx, GElf_Shdr shdr, char *binary)

```
....
2608.      globalstrtab = calloc(1, shdr.sh_size + 3);
```

Memory Leak\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=407
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2623	2623

Object	globalsymtab	globalsymtab

Code Snippet
File Name wcc/wcc.c
Method int save_dynsym(ctx_t * ctx, GElf_Shdr shdr, char *binary)

```
....
2623.       globalsymtab = calloc(1, sizeof(Elf_Sym) + shdr.sh_size);
```

Memory Leak\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=408
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2689	2689
Object	globalreloc	globalreloc

Code Snippet
File Name wcc/wcc.c
Method int append_reloc(Elf_Rela * r)

```
....
2689.       globalreloc = calloc(1, sizeof(Elf_Rela));
```

Memory Leak\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=409
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2716	2716
Object	g	g

Code Snippet
File Name wcc/wcc.c
Method int save_global_import(ctx_t * ctx, char *sname, msec_t * sec, Elf_Rela * r, unsigned int sindex)

```
....
2716.     g = calloc(1, sizeof(gimport_t));
```

Memory Leak\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=410
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2717	2717
Object	sname	sname

Code Snippet

File Name	wcc/wcc.c
Method	int save_global_import(ctx_t * ctx, char *sname, msec_t * sec, Elf_Rela * r, unsigned int sindex)

```
....
2717.     g->sname = strdup(sname);
```

Memory Leak\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=411
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2721	2721
Object	rnew	rnew

Code Snippet

File Name	wcc/wcc.c
Method	int save_global_import(ctx_t * ctx, char *sname, msec_t * sec, Elf_Rela * r, unsigned int sindex)

```
....
2721.     rnew = calloc(1, sizeof(Elf_Rela));
```

Memory Leak\Path 26:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=412
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2727	2727
Object	gimports	gimports

Code Snippet

File Name wcc/wcc.c

Method int save_global_import(ctx_t * ctx, char *sname, msec_t * sec, Elf_Rela * r, unsigned int sindex)

```
....
2727.       gimports = calloc(1, sizeof(gimport_t *));
```

Memory Leak\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=413
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2767	2767
Object	rout	rout

Code Snippet

File Name wcc/wcc.c

Method int save_reloc(ctx_t * ctx, Elf_Rela * r, unsigned int sindex, int has_addend)

```
....
2767.     rout = calloc(1, sizeof(Elf_Rela));     // Work on a copy of the
relocation instead of the original one
```

Memory Leak\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=414
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3837	3837
Object	opt_interp	opt_interp

Code Snippet
File Name wcc/wcc.c
Method int ctx_getopt(ctx_t * ctx, int argc, char **argv)

```
....
3837.         ctx->opt_interp = strdup(optarg);
```

Memory Leak\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=415
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3847	3847
Object	opt_binname	opt_binname

Code Snippet
File Name wcc/wcc.c
Method int ctx_getopt(ctx_t * ctx, int argc, char **argv)

```
....
3847.         ctx->opt_binname = strdup(optarg);
```

Memory Leak\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=416
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3916	3916
Object	binname	binname

Code Snippet

File Name	wcc/wcc.c
Method	int ctx_getopt(ctx_t * ctx, int argc, char **argv)

```
....
3916.     ctx->binname = strdup(argv[count + 1]);
```

Memory Leak\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=417
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	412	412
Object	opt	opt

Code Snippet

File Name	wcc/wsh.c
Method	void completion(const char *buf, linenoiseCompletions * lc)

```
....
412.               opt = strdup(buf);
```

Memory Leak\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=418
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	636	636
Object	ret	ret

Code Snippet

File Name	wcc/wsh.c
Method	char *decode_type(unsigned int type)

```
....
636.               ret = calloc(1, 200);
```

Memory Leak\Path 33:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=419
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	686	686
Object	s	s

Code Snippet

File Name wcc/wsh.c

Method int add_symbol(char *symbol, char *libname, char *htype, char *hbind, unsigned long value, unsigned int size, unsigned long int addr)

```
....
686.         s = calloc(1, sizeof(symbols_t));
```

Memory Leak\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=420
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	689	689
Object	symbol	symbol

Code Snippet

File Name wcc/wsh.c

Method int add_symbol(char *symbol, char *libname, char *htype, char *hbind, unsigned long value, unsigned int size, unsigned long int addr)

```
....
689.         s->symbol = strdup(symbol);
```

Memory Leak\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=421
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	692	692
Object	libname	libname

Code Snippet
File Name wcc/wsh.c
Method int add_symbol(char *symbol, char *libname, char *htype, char *hbind, unsigned long value, unsigned int size, unsigned long int addr)

```
....
692.         s->libname = strdup(libname);
```

Memory Leak\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=422
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	693	693
Object	htype	htype

Code Snippet
File Name wcc/wsh.c
Method int add_symbol(char *symbol, char *libname, char *htype, char *hbind, unsigned long value, unsigned int size, unsigned long int addr)

```
....
693.         s->htype = strdup(htype);
```

Memory Leak\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=423
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	694	694
Object	hbind	hbind

Code Snippet

File Name	wcc/wsh.c
Method	int add_symbol(char *symbol, char *libname, char *htype, char *hbind, unsigned long value, unsigned int size, unsigned long int addr)

```
....
694.         s->hbind = strdup(hbind);
```

Memory Leak\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=424
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	717	717
Object	s	s

Code Snippet

File Name	wcc/wsh.c
Method	void section_add(unsigned long int addr, unsigned long int size, char *libname, char *name, char *perms, int flags)

```
....
717.         s = calloc(1, sizeof(sections_t));
```

Memory Leak\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=425
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	722	722
Object	libname	libname

Code Snippet

File Name	wcc/wsh.c
Method	void section_add(unsigned long int addr, unsigned long int size, char *libname, char *name, char *perms, int flags)

```
....
722.         s->libname = strdup(libname);
```

Memory Leak\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=426
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	723	723
Object	name	name

Code Snippet	
File Name	wcc/wsh.c
Method	void section_add(unsigned long int addr, unsigned long int size, char *libname, char *name, char *perms, int flags)

```
....
723.         s->name = strdup(name);
```

Memory Leak\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=427
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	724	724
Object	perms	perms

Code Snippet	
File Name	wcc/wsh.c
Method	void section_add(unsigned long int addr, unsigned long int size, char *libname, char *name, char *perms, int flags)

```
....
724.         s->perms = strdup(perms);
```

Memory Leak\Path 42:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=428
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	736	736
Object	s	s

Code Snippet

File Name wcc/wsh.c

Method void segment_add(unsigned long int addr, unsigned long int size, char *perms, char *fname, char *ptype, int flags)

```
....
736.         s = calloc(1, sizeof(segments_t));
```

Memory Leak\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=429
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	741	741
Object	libname	libname

Code Snippet

File Name wcc/wsh.c

Method void segment_add(unsigned long int addr, unsigned long int size, char *perms, char *fname, char *ptype, int flags)

```
....
741.         s->libname = strdup(fname);
```

Memory Leak\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=430
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	742	742
Object	perms	perms

Code Snippet
File Name wcc/wsh.c
Method void segment_add(unsigned long int addr, unsigned long int size, char *perms, char *fname, char *ptype, int flags)

```
....
742.         s->perms = strdup(perms);
```

Memory Leak\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=431
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	743	743
Object	type	type

Code Snippet
File Name wcc/wsh.c
Method void segment_add(unsigned long int addr, unsigned long int size, char *perms, char *fname, char *ptype, int flags)

```
....
743.         s->type = strdup(ptype);
```

Memory Leak\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=432
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	755	755
Object	s	s

Code Snippet	
File Name	wcc/wsh.c
Method	void entry_point_add(unsigned long long int addr, char *fname)

```
....
755.        s = calloc(1, sizeof(eps_t));
```

Memory Leak\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=433
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	756	756
Object	name	name

Code Snippet	
File Name	wcc/wsh.c
Method	void entry_point_add(unsigned long long int addr, char *fname)

```
....
756.        s->name = strdup(fname);
```

Memory Leak\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=434
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1585	1585
Object	ptr	ptr

Code Snippet	
File Name	wcc/wsh.c
Method	int alloccharbuf(lua_State * L)

```
....
1585.       ptr = calloc(n * sizeof(char *), 1);
```

Memory Leak\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=435
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	2099	2099
Object	errcontext	errcontext

Code Snippet

File Name	wcc/wsh.c
Method	static int libcall(lua_State * L)

```
....
2099.           wsh->errcontext = calloc(1, sizeof(ucontext_t));
```

Memory Leak\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=436
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	2491	2491
Object	luabuff	luabuff

Code Snippet

File Name	wcc/wsh.c
Method	int luabuff_append(char *cmd){

```
....
2491.           wsh->luabuff = calloc(1, 4096);
```

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=1050048&projectid=50038&pathid=97
Status	New

The size of the buffer used by append_sym in Elf64_Sym, at line 1725 of wcc/wcc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that append_sym passes to Elf64_Sym, at line 1725 of wcc/wcc.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1734	1734
Object	Elf64_Sym	Elf64_Sym

Code Snippet

File Name	wcc/wcc.c
Method	unsigned int append_sym(Elf_Sym * s)

```
....
1734.     memcpy(globalsymtab + globalsymtablen, s, sizeof(Elf_Sym));
```

Buffer Overflow boundcpy WrongSizeParam\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=98
Status	New

The size of the buffer used by append_reloc in Elf64_Rela, at line 2672 of wcc/wcc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that append_reloc passes to Elf64_Rela, at line 2672 of wcc/wcc.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2694	2694
Object	Elf64_Rela	Elf64_Rela

Code Snippet

File Name	wcc/wcc.c
Method	int append_reloc(Elf_Rela * r)

```
....
2694.     memcpy(globalreloc + globalreloclen, r, sizeof(Elf_Rela));
```

Buffer Overflow boundcpy WrongSizeParam\Path 3:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=99
Status	New

The size of the buffer used by save_global_import in Elf64_Rela, at line 2701 of wcc/wcc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that save_global_import passes to Elf64_Rela, at line 2701 of wcc/wcc.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2722	2722
Object	Elf64_Rela	Elf64_Rela

Code Snippet
File Name wcc/wcc.c
Method int save_global_import(ctx_t * ctx, char *sname, msec_t * sec, Elf_Rela * r, unsigned int sindex)

```
....
2722.     memcpy(rnew, r, sizeof(Elf_Rela));
```

Buffer Overflow boundcpy WrongSizeParam\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=100
Status	New

The size of the buffer used by save_reloc in Elf64_Rela, at line 2755 of wcc/wcc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that save_reloc passes to Elf64_Rela, at line 2755 of wcc/wcc.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2768	2768
Object	Elf64_Rela	Elf64_Rela

Code Snippet
File Name wcc/wcc.c
Method int save_reloc(ctx_t * ctx, Elf_Rela * r, unsigned int sindex, int has_addend)

```
....
2768.     memcpy(rout, r, sizeof(Elf_Rela));
```

Buffer Overflow boundcpy WrongSizeParam\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=500

	38&pathid=101
Status	New

The size of the buffer used by sighandler in ucontext_t, at line 3556 of wcc/wsh.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sighandler passes to ucontext_t, at line 3556 of wcc/wsh.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3617	3617
Object	ucontext_t	ucontext_t

Code Snippet
File Name wcc/wsh.c
Method void sighandler(int signal, siginfo_t * s, void *ptr)

```
....
3617.             memcpy(wsh->errcontext, u, sizeof(ucontext_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=102
Status	New

The size of the buffer used by add_symaddr in symaddr, at line 418 of wcc/wcc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that add_symaddr passes to symaddr, at line 418 of wcc/wcc.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	444	444
Object	symaddr	symaddr

Code Snippet
File Name wcc/wcc.c
Method void add_symaddr(ctx_t * ctx, const char *name, int addr, char symclass)

```
....
444.     memset(sa, 0, sizeof(struct symaddr));
```

Buffer Overflow boundcpy WrongSizeParam\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=103
Status	New

The size of the buffer used by libcall in ucontext_t, at line 2059 of wcc/wsh.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that libcall passes to ucontext_t, at line 2059 of wcc/wsh.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	2105	2105
Object	ucontext_t	ucontext_t

Code Snippet
File Name wcc/wsh.c
Method static int libcall(lua_State * L)

```
....
2105.        memset(wsh->errcontext, 0x00, sizeof(ucontext_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=104
Status	New

The size of the buffer used by sort_learnt in learn_key_t, at line 1845 of wcc/wsh.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sort_learnt passes to learn_key_t, at line 1845 of wcc/wsh.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1847	1847
Object	learn_key_t	learn_key_t

Code Snippet
File Name wcc/wsh.c
Method int sort_learnt(learn_t *a, learn_t *b)

```
....
1847.        return memcmp(&a->key, &b->key, sizeof(learn_key_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=105
Status	New

The size of the buffer used by str_rep in l, at line 120 of wcc/lstrlib.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that str_rep passes to l, at line 120 of wcc/lstrlib.c, to overwrite the target buffer.

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c
Line	133	133
Object	l	l

Code Snippet
File Name wcc/lstrlib.c
Method static int str_rep (lua_State *L) {

```
....
133.        memcpy(p, s, l * sizeof(char)); p += l;
```

Buffer Overflow boundcpy WrongSizeParam\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=106
Status	New

The size of the buffer used by str_rep in char, at line 120 of wcc/lstrlib.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that str_rep passes to char, at line 120 of wcc/lstrlib.c, to overwrite the target buffer.

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c
Line	133	133
Object	char	char

Code Snippet
File Name wcc/lstrlib.c
Method static int str_rep (lua_State *L) {

```
....
133.        memcpy(p, s, l * sizeof(char)); p += l;
```

Buffer Overflow boundcpy WrongSizeParam\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=107
Status	New

The size of the buffer used by str_rep in lsep, at line 120 of wcc/lstrlib.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that str_rep passes to lsep, at line 120 of wcc/lstrlib.c, to overwrite the target buffer.

	Source	Destination

File	wcc/lstrlib.c	wcc/lstrlib.c
Line	135	135
Object	lsep	lsep

Code Snippet
File Name wcc/lstrlib.c
Method static int str_rep (lua_State *L) {

```
....
135.          memcpy(p, sep, lsep * sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=108
Status	New

The size of the buffer used by str_rep in char, at line 120 of wcc/lstrlib.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that str_rep passes to char, at line 120 of wcc/lstrlib.c, to overwrite the target buffer.

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c
Line	135	135
Object	char	char

Code Snippet
File Name wcc/lstrlib.c
Method static int str_rep (lua_State *L) {

```
....
135.          memcpy(p, sep, lsep * sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=109
Status	New

The size of the buffer used by str_rep in l, at line 120 of wcc/lstrlib.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that str_rep passes to l, at line 120 of wcc/lstrlib.c, to overwrite the target buffer.

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c

Line	139	139
Object	l	l

Code Snippet
File Name wcc/lstrlib.c
Method static int str_rep (lua_State *L) {

```
....
139.       memcpy(p, s, l * sizeof(char));  /* last copy (not followed by
separator) */
```

Buffer Overflow boundcpy WrongSizeParam\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=110
Status	New

The size of the buffer used by str_rep in char, at line 120 of wcc/lstrlib.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that str_rep passes to char, at line 120 of wcc/lstrlib.c, to overwrite the target buffer.

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c
Line	139	139
Object	char	char

Code Snippet
File Name wcc/lstrlib.c
Method static int str_rep (lua_State *L) {

```
....
139.       memcpy(p, s, l * sizeof(char));  /* last copy (not followed by
separator) */
```

Buffer Overflow boundcpy WrongSizeParam\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=111
Status	New

The size of the buffer used by *scanformat in char, at line 949 of wcc/lstrlib.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *scanformat passes to char, at line 949 of wcc/lstrlib.c, to overwrite the target buffer.

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c

Line	964	964
Object	char	char

Code Snippet
File Name wcc/lstrlib.c
Method static const char *scanformat (lua_State *L, const char *strfrmt, char *form) {

```
....
964.    memcpy(form, strfrmt, ((p - strfrmt) + 1) * sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=112
Status	New

The size of the buffer used by copy2buff in l, at line 446 of wcc/lvm.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that copy2buff passes to l, at line 446 of wcc/lvm.c, to overwrite the target buffer.

	Source	Destination
File	wcc/lvm.c	wcc/lvm.c
Line	450	450
Object	l	l

Code Snippet
File Name wcc/lvm.c
Method static void copy2buff (StkId top, int n, char *buff) {

```
....
450.      memcpy(buff + tl, svalue(top - n), l * sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=113
Status	New

The size of the buffer used by copy2buff in char, at line 446 of wcc/lvm.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that copy2buff passes to char, at line 446 of wcc/lvm.c, to overwrite the target buffer.

	Source	Destination
File	wcc/lvm.c	wcc/lvm.c
Line	450	450

Object	char	char

Code Snippet
File Name wcc/lvm.c
Method static void copy2buff (StkId top, int n, char *buff) {

```
....
450.        memcpy(buff + tl, svalue(top - n), l * sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=114
Status	New

The size of the buffer used by add_symaddr in sa, at line 418 of wcc/wcc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that add_symaddr passes to sa, at line 418 of wcc/wcc.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	464	464
Object	sa	sa

Code Snippet
File Name wcc/wcc.c
Method void add_symaddr(ctx_t * ctx, const char *name, int addr, char symclass)

```
....
464.     memcpy(globalstrtab + globalstrtablen, sa->name, strlen(sa->name) + 1);
```

Buffer Overflow boundcpy WrongSizeParam\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=115
Status	New

The size of the buffer used by append_strtab in str, at line 1745 of wcc/wcc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that append_strtab passes to str, at line 1745 of wcc/wcc.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1755	1755
Object	str	str

Code Snippet	
File Name	wcc/wcc.c
Method	unsigned int append_strtab(char *str)

```
....
1755.    memcpy(globalstrtab + globalstrtablen, str, strlen(str) + 1);
```

Buffer Overflow boundcpy WrongSizeParam\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=116
Status	New

The size of the buffer used by write_shdrs in s, at line 1868 of wcc/wcc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that write_shdrs passes to s, at line 1868 of wcc/wcc.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1902	1902
Object	s	s

Code Snippet	
File Name	wcc/wcc.c
Method	static unsigned int write_shdrs(ctx_t * ctx)

```
....
1902.      memcpy(ctx->strndx + ctx->strndx_len, s->name, strlen(s->name) + 1);       // do copy the final "\x00"
```

Buffer Overflow boundcpy WrongSizeParam\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=117
Status	New

The size of the buffer used by craft_section in s, at line 2443 of wcc/wcc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that craft_section passes to s, at line 2443 of wcc/wcc.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2472	2472
Object	s	s

Code Snippet

File Name	wcc/wcc.c
Method	int craft_section(ctx_t * ctx, msec_t * m)

```
....
2472.    memcpy(ctx->strndx + ctx->strndx_len, s->name, strlen(s-
>name));
```

Buffer Overflow boundcpy WrongSizeParam\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=118
Status	New

The size of the buffer used by priv_memcpy in arg3, at line 4299 of wcc/wsh.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that priv_memcpy passes to arg3, at line 4299 of wcc/wsh.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	4310	4310
Object	arg3	arg3

Code Snippet

File Name	wcc/wsh.c
Method	int priv_memcpy(lua_State * L)

```
....
4310.        ret = memcpy(arg1, arg2, arg3);
```

Buffer Overflow boundcpy WrongSizeParam\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=119
Status	New

The size of the buffer used by rawmemwrite in len, at line 5252 of wcc/wsh.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that rawmemwrite passes to len, at line 5252 of wcc/wsh.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	5263	5263
Object	len	len

Code Snippet

File Name	wcc/wsh.c
Method	int rawmemwrite (lua_State *L) {

```
....
5263.       memmove(addr, data, len);
```

Buffer Overflow boundcpy WrongSizeParam\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=120
Status	New

The size of the buffer used by open_target in sb, at line 2348 of wcc/wcc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that open_target passes to sb, at line 2348 of wcc/wcc.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2383	2383
Object	sb	sb

Code Snippet
File Name wcc/wcc.c
Method int open_target(ctx_t * ctx)

```
....
2383.      memset(p, ctx->opt_poison, sb.st_size);
```

Buffer Overflow boundcpy WrongSizeParam\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=121
Status	New

The size of the buffer used by bsspolute in s, at line 3834 of wcc/wsh.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that bsspolute passes to s, at line 3834 of wcc/wsh.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3845	3845
Object	s	s

Code Snippet
File Name wcc/wsh.c
Method int bsspolute(lua_State * L)

```
....
3845.                          memset(s->addr, poison--, s->size);
```

Buffer Overflow boundcpy WrongSizeParam\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=122
Status	New

The size of the buffer used by ralloc in sz, at line 3880 of wcc/wsh.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ralloc passes to sz, at line 3880 of wcc/wsh.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3911	3911
Object	sz	sz

Code Snippet
File Name wcc/wsh.c
Method int ralloc(lua_State * L)

```
....
3911.       memset(ptr, poison ? poison : default_poison +
global_xalloc, sz);     // map with default poison bytes
```

Buffer Overflow boundcpy WrongSizeParam\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=123
Status	New

The size of the buffer used by xalloc in sz, at line 3935 of wcc/wsh.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that xalloc passes to sz, at line 3935 of wcc/wsh.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3970	3970
Object	sz	sz

Code Snippet
File Name wcc/wsh.c
Method int xalloc(lua_State * L)

```
....
3970.       memset(ptr, poison ? poison : default_poison +
global_xalloc, sz);     // map with default poison bytes
```

Buffer Overflow boundcpy WrongSizeParam\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=124
Status	New

The size of the buffer used by xfree in sz, at line 3996 of wcc/wsh.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that xfree passes to sz, at line 3996 of wcc/wsh.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	4006	4006
Object	sz	sz

Code Snippet
File Name wcc/wsh.c
Method void xfree(lua_State * L)

```
....
4006.       memset(trueptr, 0x00, sz);
```

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=1050048&projectid=50038&pathid=125
Status	New

Calling free() (line 1048) on a variable that was not dynamically allocated (line 1048) in file wcc/wcc.c may result with a crash.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1071	1071
Object	n	n

Code Snippet

File Name	wcc/wcc.c
Method	int merge_phdrs(ctx_t * ctx)

```
....
1071.         free(n);
```

MemoryFree on StackVariable\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=126
Status	New

Calling free() (line 938) on a variable that was not dynamically allocated (line 938) in file wcc/wsh.c may result with a crash.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	948	948
Object	s	s

Code Snippet

File Name	wcc/wsh.c
Method	int empty_symbols(void)

```
....
948.                    free(s);
```

MemoryFree on StackVariable\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=127
Status	New

Calling free() (line 958) on a variable that was not dynamically allocated (line 958) in file wcc/wsh.c may result with a crash.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	968	968
Object	s	s

Code Snippet

File Name	wcc/wsh.c
Method	int empty_phdrs(void)

```
....
968.                        free(s);
```

MemoryFree on StackVariable\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=128
Status	New

Calling free() (line 978) on a variable that was not dynamically allocated (line 978) in file wcc/wsh.c may result with a crash.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	988	988
Object	s	s

Code Snippet
File Name wcc/wsh.c
Method int empty_shdrs(void)

```
....
988.                        free(s);
```

MemoryFree on StackVariable\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=129
Status	New

Calling free() (line 997) on a variable that was not dynamically allocated (line 997) in file wcc/wsh.c may result with a crash.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1005	1005
Object	s	s

Code Snippet
File Name wcc/wsh.c
Method int empty_eps(void)

```
....
1005.                        free(s);
```

MemoryFree on StackVariable\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=130
Status	New

Calling free() (line 1673) on a variable that was not dynamically allocated (line 1673) in file wcc/wsh.c may result with a crash.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1736	1736
Object	input	input

Code Snippet

File Name	wcc/wsh.c
Method	int run_shell(lua_State * L)

```
....
1736.                               free(input);
```

MemoryFree on StackVariable\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=131
Status	New

Calling free() (line 1673) on a variable that was not dynamically allocated (line 1673) in file wcc/wsh.c may result with a crash.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1741	1741
Object	input	input

Code Snippet

File Name	wcc/wsh.c
Method	int run_shell(lua_State * L)

```
....
1741.                               free(input);
```

MemoryFree on StackVariable\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=132
Status	New

Calling free() (line 1673) on a variable that was not dynamically allocated (line 1673) in file wcc/wsh.c may result with a crash.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1745	1745
Object	input	input

Code Snippet

File Name	wcc/wsh.c
Method	int run_shell(lua_State * L)

```
....
1745.                               free(input);
```

MemoryFree on StackVariable\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=133
Status	New

Calling free() (line 1673) on a variable that was not dynamically allocated (line 1673) in file wcc/wsh.c may result with a crash.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1749	1749
Object	input	input

Code Snippet

File Name	wcc/wsh.c
Method	int run_shell(lua_State * L)

```
....
1749.                              free(input);
```

MemoryFree on StackVariable\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=134
Status	New

Calling free() (line 1673) on a variable that was not dynamically allocated (line 1673) in file wcc/wsh.c may result with a crash.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1775	1775
Object	input	input

Code Snippet

File Name	wcc/wsh.c
Method	int run_shell(lua_State * L)

```
....
1775.                       free(input);
```

MemoryFree on StackVariable\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=135
Status	New

Calling free() (line 2512) on a variable that was not dynamically allocated (line 2512) in file wcc/wsh.c may result with a crash.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	2648	2648
Object	demangled	demangled

Code Snippet

File Name	wcc/wsh.c
Method	void scan_syms(char *dynstr, Elf_Sym * sym, unsigned long int sz, char *libname)

```
....
2648.                 free(demangled);
```

MemoryFree on StackVariable\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=136
Status	New

Calling free() (line 2932) on a variable that was not dynamically allocated (line 2932) in file wcc/wsh.c may result with a crash.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	2953	2953
Object	funcnames	funcnames

Code Snippet

File Name	wcc/wsh.c
Method	void print_backtrace(void)

```
....
2953.         free(funcnames);
```

MemoryFree on StackVariable\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=137
Status	New

Calling free() (line 3203) on a variable that was not dynamically allocated (line 3203) in file wcc/wsh.c may result with a crash.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3218	3218
Object	bt_syms	bt_syms

Code Snippet

File Name	wcc/wsh.c
Method	int mk_backtrace(void)

```
....
3218.       free(bt_syms);
```

Double Free

Query Path:
CPP\Cx\CPP Medium Threat\Double Free Version:1

Categories

NIST SP 800-53: SI-16 Memory Protection (P1)

Description

Double Free\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=379
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1741	1741
Object	input	input

Code Snippet
File Name wcc/wsh.c
Method int run_shell(lua_State * L)

```
....
1741.                           free(input);
```

Double Free\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=380
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1736	1745
Object	input	input

Code Snippet
File Name wcc/wsh.c
Method int run_shell(lua_State * L)

```
....
1736.                                   free(input);
....
1745.                                   free(input);
```

Double Free\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=381
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1745	1749
Object	input	input

Code Snippet
File Name wcc/wsh.c
Method int run_shell(lua_State * L)

```
....
1745.                                   free(input);
....
1749.                                   free(input);
```

Double Free\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=382
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1736	1749
Object	input	input

Code Snippet
File Name wcc/wsh.c
Method int run_shell(lua_State * L)

```
....
1736.                                   free(input);
....
1749.                                   free(input);
```

Double Free\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=383
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1749	1775
Object	input	input

Code Snippet

File Name	wcc/wsh.c
Method	int run_shell(lua_State * L)

```
....
1749.                                    free(input);
....
1775.                            free(input);
```

Double Free\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=384
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1745	1775
Object	input	input

Code Snippet

File Name	wcc/wsh.c
Method	int run_shell(lua_State * L)

```
....
1745.                                    free(input);
....
1775.                            free(input);
```

Double Free\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=500

	38&pathid=385
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1736	1775
Object	input	input

Code Snippet

File Name wcc/wsh.c

Method int run_shell(lua_State * L)

```
....
1736.                           free(input);
....
1775.                     free(input);
```

Double Free\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=386
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1882	1889
Object	l	l

Code Snippet

File Name wcc/wsh.c

Method int prototypes(lua_State * L)

```
....
1882.                     free(l);
....
1889.                     free(l);
```

Buffer Overflow AddressOfLocalVarReturned

Query Path:

CPP\Cx\CPP Buffer Overflow\Buffer Overflow AddressOfLocalVarReturned Version:1

Categories

PCI DSS v3.2: PCI DSS (3.2) - 6.5.2 - Buffer overflows
NIST SP 800-53: SC-5 Denial of Service Protection (P1)
OWASP Top 10 2017: A1-Injection

Description

Buffer Overflow AddressOfLocalVarReturned\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=94
Status	New

The pointer res at wcc/wsh.c in line 830 is being used after it has been freed.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	839	839
Object	res	res

Code Snippet
File Name wcc/wsh.c
Method sections_t *section_from_addr(unsigned long int addr)

```
....
839.        return res;
```

Buffer Overflow AddressOfLocalVarReturned\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=95
Status	New

The pointer res at wcc/wsh.c in line 845 is being used after it has been freed.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	854	854
Object	res	res

Code Snippet
File Name wcc/wsh.c
Method segments_t *segment_from_addr(unsigned long int addr)

```
....
854.        return res;
```

Buffer Overflow AddressOfLocalVarReturned\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=500

	38&pathid=96
Status	New

The pointer res at wcc/wsh.c in line 860 is being used after it has been freed.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	869	869
Object	res	res

Code Snippet

File Name wcc/wsh.c

Method symbols_t *symbol_from_addr(unsigned long int addr)

```
....
869.          return res;
```

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=1050048&projectid=50038&pathid=138
Status	New

The function storage_needed in wcc/wcc.c at line 551 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	wcc/wcc.c	wcc/wcc.c
Line	582	582
Object	storage_needed	storage_needed

Code Snippet

File Name wcc/wcc.c

Method int rd_symbols(ctx_t * ctx)

```
....
582.     symbol_table = (asymbol **) malloc(storage_needed);
```

Wrong Size t Allocation\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=500

	38&pathid=139
Status	New

The function storage_needed in wcc/wcc.c at line 551 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	wcc/wcc.c	wcc/wcc.c
Line	626	626
Object	storage_needed	storage_needed

Code Snippet
File Name wcc/wcc.c
Method int rd_symbols(ctx_t * ctx)

```
....
626.     symbol_table = (asymbol **) malloc(storage_needed);
```

Integer Overflow

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

Categories

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

Description

Integer Overflow\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=231
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 1847 of wcc/wcc.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1853	1853
Object	AssignExpr	AssignExpr

Code Snippet
File Name wcc/wcc.c
Method static unsigned int process_text(ctx_t * ctx)

```
....
1853.     delta = orig_text - textvma;
```

Integer Overflow\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=232
Status	New

A variable of a larger data type, newsz, is being assigned to a smaller data type, in 1656 of wcc/wcc.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1666	1666
Object	newsz	newsz

Code Snippet

File Name	wcc/wcc.c
Method	int fixup_text(ctx_t * ctx)

```
....
1666.        unsigned int newsz = datavma - textvma + maxdata - mindata;
```

Stored Buffer Overflow boundcpy

Query Path:
CPP\Cx\CPP Stored Vulnerabilities\Stored Buffer Overflow boundcpy Version:1

Categories

NIST SP 800-53: SI-10 Information Input Validation (P1)
OWASP Top 10 2017: A1-Injection

Description

Stored Buffer Overflow boundcpy\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=448
Status	New

The size of the buffer used by rd_phdrs in eph, at line 1129 of wcc/wcc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that rd_phdrs passes to p, at line 1129 of wcc/wcc.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1150	1172
Object	p	eph

Code Snippet

File Name	wcc/wcc.c

Method static unsigned int rd_phdrs(ctx_t * ctx)

```
....
1150.     nread = read(fdin, p, sb.st_size);
....
1172.       memcpy(ms, phdr, sizeof(Elf_Phdr));
```

Stored Buffer Overflow boundcpy\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=449
Status	New

The size of the buffer used by rd_phdrs in sizeof, at line 1129 of wcc/wcc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that rd_phdrs passes to p, at line 1129 of wcc/wcc.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1150	1172
Object	p	sizeof

Code Snippet
File Name wcc/wcc.c
Method static unsigned int rd_phdrs(ctx_t * ctx)

```
....
1150.     nread = read(fdin, p, sb.st_size);
....
1172.       memcpy(ms, phdr, sizeof(Elf_Phdr));
```

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=1050048&projectid=50038&pathid=447
Status	New

The pointer input at wcc/wsh.c in line 1673 is being used after it has been freed.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1736	1739
Object	input	input

Code Snippet
File Name wcc/wsh.c
Method int run_shell(lua_State * L)

```
....
1736.                          free(input);
....
1739.                    if (!strncmp(input, "exec ", 5)) {
```

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=1050048&projectid=50038&pathid=450
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1679	1679
Object	fgets	fgets

Code Snippet
File Name wcc/wsh.c
Method int run_shell(lua_State * L)

```
....
1679.                    if (fgets(shell_prompt, sizeof(shell_prompt),
stdin) == 0 || strcmp(shell_prompt, "cont\n") == 0){
```

Improper Resource Access Authorization\Path 2:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=451
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1874	1874
Object	fgets	fgets

Code Snippet

File Name	wcc/wsh.c
Method	int prototypes(lua_State * L)

```
....
1874.        while (fgets(line, sizeof(line), wsh->learnfile)) {
```

Improper Resource Access Authorization\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=452
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1679	1679
Object	shell_prompt	shell_prompt

Code Snippet

File Name	wcc/wsh.c
Method	int run_shell(lua_State * L)

```
....
1679.                    if (fgets(shell_prompt, sizeof(shell_prompt),
stdin) == 0 || strcmp(shell_prompt, "cont\n") == 0){
```

Improper Resource Access Authorization\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=453
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c

Line	1874	1874
Object	line	line

Code Snippet
File Name wcc/wsh.c
Method int prototypes(lua_State * L)

```
....
1874.        while (fgets(line, sizeof(line), wsh->learnfile)) {
```

Improper Resource Access Authorization\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=454
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1150	1150
Object	p	p

Code Snippet
File Name wcc/wcc.c
Method static unsigned int rd_phdrs(ctx_t * ctx)

```
....
1150.     nread = read(fdin, p, sb.st_size);
```

Improper Resource Access Authorization\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=455
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2387	2387
Object	p	p

Code Snippet
File Name wcc/wcc.c
Method int open_target(ctx_t * ctx)

```
....
2387.       read(fdin, p, sb.st_size);
```

Improper Resource Access Authorization\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=456
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2538	2538
Object	buf	buf

Code Snippet
File Name wcc/wcc.c
Method static int read_section(ctx_t * ctx, asection * s)

```
....
2538.       nread = read(fd, buf, s->size);
```

Improper Resource Access Authorization\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=457
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2541	2541
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name wcc/wcc.c
Method static int read_section(ctx_t * ctx, asection * s)

```
....
2541.         nread = read(fd, buf + n, s->size - n);
```

Improper Resource Access Authorization\Path 9:

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

BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=458
Status New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	2861	2861
Object	buff	buff

Code Snippet
File Name wcc/wsh.c
Method int print_procmap(unsigned int pid)

```
....
2861.        while ((n = read(fd, buff, 4096)) > 0){
```

Improper Resource Access Authorization\Path 10:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=459
Status New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	4635	4635
Object	sig	sig

Code Snippet
File Name wcc/wsh.c
Method unsigned int read_elf_sig(char *fname, struct stat *sb)

```
....
4635.        read(fd, sig, 4);
```

Improper Resource Access Authorization\Path 11:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=460
Status New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	579	579

Object	fprintf	fprintf

Code Snippet
File Name wcc/wcc.c
Method int rd_symbols(ctx_t * ctx)

```
....
579.        fprintf(stderr, "warning: no symbols\n");
```

Improper Resource Access Authorization\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=461
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	623	623
Object	fprintf	fprintf

Code Snippet
File Name wcc/wcc.c
Method int rd_symbols(ctx_t * ctx)

```
....
623.        fprintf(stderr, "warning: no symbols\n");
```

Improper Resource Access Authorization\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=462
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3887	3887
Object	fprintf	fprintf

Code Snippet
File Name wcc/wcc.c
Method int ctx_getopt(ctx_t * ctx, int argc, char **argv)

```
....
3887.        fprintf(stderr, "Try `%s --help' for more information.\n",
argv[0]);
```

Improper Resource Access Authorization\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=463
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3891	3891
Object	fprintf	fprintf

Code Snippet
File Name   wcc/wcc.c
Method   int ctx_getopt(ctx_t * ctx, int argc, char **argv)

```
....
3891.        fprintf(stderr, "%s: invalid option -- %c\n", argv[0], c);
```

Improper Resource Access Authorization\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=464
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3892	3892
Object	fprintf	fprintf

Code Snippet
File Name   wcc/wcc.c
Method   int ctx_getopt(ctx_t * ctx, int argc, char **argv)

```
....
3892.        fprintf(stderr, "Try `%s --help' for more information.\n",
argv[0]);
```

Improper Resource Access Authorization\Path 16:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=465
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3899	3899
Object	fprintf	fprintf

Code Snippet
File Name wcc/wcc.c
Method int ctx_getopt(ctx_t * ctx, int argc, char **argv)

```
....
3899.       fprintf(stderr, "error: No source binary found in
arguments.\n");
```

Improper Resource Access Authorization\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=466
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3900	3900
Object	fprintf	fprintf

Code Snippet
File Name wcc/wcc.c
Method int ctx_getopt(ctx_t * ctx, int argc, char **argv)

```
....
3900.       fprintf(stderr, "Try `%s --help' for more information.\n",
argv[0]);
```

Improper Resource Access Authorization\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=467
Status	New

	Source	Destination

File	wcc/wsh.c	wcc/wsh.c
Line	173	173
Object	fprintf	fprintf

Code Snippet

File Name	wcc/wsh.c
Method	void fatal_error(lua_State * L, char *msg)

```
....
173.        fprintf(stderr, "\nFATAL ERROR:\n  %s: %s\n\n", msg,
lua_tostring(L, -1));
```

Improper Resource Access Authorization\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=468
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	249	249
Object	fprintf	fprintf

Code Snippet

File Name	wcc/wsh.c
Method	static unsigned long int resolve_addr(char *symbol, char *libname)

```
....
249.              fprintf(stderr, "ERROR: %s\n", dlerror());
```

Improper Resource Access Authorization\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=469
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	263	263
Object	fprintf	fprintf

Code Snippet

File Name	wcc/wsh.c

Method static unsigned long int resolve_addr(char *symbol, char *libname)

```
....
263.                        fprintf(stderr, "ERROR: %s\n", err);
```

Improper Resource Access Authorization\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=470
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	356	356
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method int scan_symbol(char *symbol, char *libname)

```
....
356.                  fprintf(stderr, "ERROR: %s\n", dlerror());
```

Improper Resource Access Authorization\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=471
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	468	468
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method int disable_aslr(void)

```
....
468.                  fprintf(stderr, "!! ERROR : open(%s, O_RDWR) %s\n",
PROC_ASLR_PATH, strerror(errno));
```

Improper Resource Access Authorization\Path 23:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=472
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	487	487
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method int enable_aslr(void)

```
....
487.              fprintf(stderr, "!! ERROR : open(%s,O_RDWR) %s\n",
PROC_ASLR_PATH, strerror(errno));
```

Improper Resource Access Authorization\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=473
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	687	687
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method int add_symbol(char *symbol, char *libname, char *htype, char *hbind, unsigned long value, unsigned int size, unsigned long int addr)

```
....
687.         if(!s){ fprintf(stderr, " !! Error: calloc() = %s\n",
strerror(errno)); return -1; }
```

Improper Resource Access Authorization\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=474
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	718	718
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method void section_add(unsigned long int addr, unsigned long int size, char *libname, char *name, char *perms, int flags)

```
....
718.         if(!s){ fprintf(stderr, " !! Error: calloc() = %s\n", 
strerror(errno)); return; }
```

Improper Resource Access Authorization\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=475
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	737	737
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method void segment_add(unsigned long int addr, unsigned long int size, char *perms, char *fname, char *ptype, int flags)

```
....
737.         if(!s){ fprintf(stderr, " !! Error: calloc() = %s\n", 
strerror(errno)); return; }
```

Improper Resource Access Authorization\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=476
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1545	1545

Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method int info(lua_State * L)

```
....
1545.                    fprintf(stderr, "ERROR: %s\n", error);
```

Improper Resource Access Authorization\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=477
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1684	1684
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method int run_shell(lua_State * L)

```
....
1684.                          fprintf(stderr, "ERROR: %s\n",
lua_tostring(L, -1));
```

Improper Resource Access Authorization\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=478
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1770	1770
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method int run_shell(lua_State * L)

```
....
1770.                          fprintf(stderr, "ERROR: %s\n",
lua_tostring(L, -1));
```

Improper Resource Access Authorization\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=479
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1839	1839
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method int learn_proto(unsigned long*arg, unsigned long int faultaddr, int reason)

```
....
1839.       fprintf(wsh->learnfile, "TAG %s %s argument%u %s %ld\n", s-
>libname, s->symbol, argn, tag, offset);
```

Improper Resource Access Authorization\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=480
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	2234	2234
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method static int libcall(lua_State * L)

```
....
2234.             fprintf(stderr, "ERROR: %s (%u)\n",
strerror(callerrno), callerrno);
```

Improper Resource Access Authorization\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=481
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	2756	2756
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method int parse_link_map_dyn(struct link_map *map)

```
....
2756.                   fprintf(stderr, "WARNING: No binary loaded in
memory. Try loadbin(). For help type help(\"loadbin\").\n");
```

Improper Resource Access Authorization\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=482
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	2807	2807
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method int exec_luabuff(void)

```
....
2807.             fprintf(stderr, "ERROR: lua_pcall() failed with
%s\n",lua_tostring(wsh->L, -1));
```

Improper Resource Access Authorization\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=483
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	2891	2891
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method int execlib(lua_State * L)

```
....
2891.             fprintf(stderr, "ERROR: fork() : %s\n",
strerror(errno));
```

Improper Resource Access Authorization\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=484
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	2904	2904
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method int execlib(lua_State * L)

```
....
2904.                                 fprintf(stderr, "ERROR: ptrace() :
%s\n", strerror(errno));
```

Improper Resource Access Authorization\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=485
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3043	3043
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method void affinity(int procnum)

```
....
3043.              fprintf(stderr, " !! ERROR: sched_setaffinity(%u): 
%s\n", procnum, strerror(errno));
```

Improper Resource Access Authorization\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=486
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3059	3059
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method void btr_enable(int procnum)

```
....
3059.        if(fd <= 0){ fprintf(stderr, "ERROR: open(%s): %s\n", 
cpupath,strerror(errno)); return; }
```

Improper Resource Access Authorization\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=487
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3061	3061
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method void btr_enable(int procnum)

```
....
3061.        if(ret != 0x00){ fprintf(stderr, "ERROR: lseek(): %s\n",
strerror(errno)); return; }
```

Improper Resource Access Authorization\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=488
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3063	3063
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method void btr_enable(int procnum)

```
....
3063.        if(ret != sizeof(data)){ fprintf(stderr, "ERROR: write():
%s\n", strerror(errno)); return; }
```

Improper Resource Access Authorization\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=489
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3065	3065
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method void btr_enable(int procnum)

```
....
3065.        if(ret != 0){ fprintf(stderr, "ERROR: close(): %s\n",
strerror(errno)); return; }
```

Improper Resource Access Authorization\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=490
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3080	3080
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method void btr_disable(int procnum)

```
....
3080.        if(fd <= 0){ fprintf(stderr, "ERROR: open(%s): %s\n",
cpupath,strerror(errno)); return; }
```

Improper Resource Access Authorization\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=491
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3082	3082
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method void btr_disable(int procnum)

```
....
3082.        if(ret != 0x00){ fprintf(stderr, "ERROR: lseek(): %s\n",
strerror(errno)); return; }
```

Improper Resource Access Authorization\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=492
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3084	3084
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method void btr_disable(int procnum)

```
....
3084.        if(ret != sizeof(data)){ fprintf(stderr, "ERROR: write():
%s\n", strerror(errno)); return; }
```

Improper Resource Access Authorization\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=493
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3086	3086
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method void btr_disable(int procnum)

```
....
3086.        if(ret != 0){ fprintf(stderr, "ERROR: close(): %s\n",
strerror(errno)); return; }
```

Improper Resource Access Authorization\Path 45:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=494
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3159	3159
Object	fprintf	fprintf

Code Snippet

File Name	wcc/wsh.c
Method	void bushandler(int signal, siginfo_t * s, void *ptr)

```
....
3159.                          fprintf(stderr, " -- SIGBUS[%03u]
%llx\t%s()+%llu\t%s\n", wsh->sigbus_count+1, u-
>uc_mcontext.gregs[REG_RIP], s->symbol, u->uc_mcontext.gregs[REG_RIP] -
s->addr, s->libname);
```

Improper Resource Access Authorization\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=495
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3161	3161
Object	fprintf	fprintf

Code Snippet

File Name	wcc/wsh.c
Method	void bushandler(int signal, siginfo_t * s, void *ptr)

```
....
3161.                          fprintf(stderr, " -- SIGBUS[%03u] %llx\n",
wsh->sigbus_count+1, u->uc_mcontext.gregs[REG_RIP]);
```

Improper Resource Access Authorization\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=496
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3234	3234
Object	fprintf	fprintf

Code Snippet

File Name	wcc/wsh.c
Method	void exit(int status)

```
....
3234.        fprintf(stderr, " + Called exit(%d), restoring...\n",
status);
```

Improper Resource Access Authorization\Path 48:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=497
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3241	3241
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method void _exit(int status)

```
....
3241.        fprintf(stderr, " + Called _exit(%d), restoring...\n",
status);
```

Improper Resource Access Authorization\Path 49:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=498
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3248	3248
Object	fprintf	fprintf

Code Snippet
File Name wcc/wsh.c
Method void exit_group(int status)

```
....
3248.        fprintf(stderr, " + Called exit_group(%d), restoring...\n",
status);
```

Improper Resource Access Authorization\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=499
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3265	3265
Object	fprintf	fprintf

Code Snippet

File Name	wcc/wsh.c
Method	int printarg(unsigned long int val)

```
....
3265.                   fprintf(stderr,"\"%s\"", ptrx);
```

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=1050048&projectid=50038&pathid=140
Status	New

The variable declared in 0 at wcc/lgc.c in line 844 is not initialized when it is used by gcfinnum at wcc/lgc.c in line 844.

	Source	Destination
File	wcc/lgc.c	wcc/lgc.c
Line	850	850
Object	0	gcfinnum

Code Snippet

File Name	wcc/lgc.c
Method	static int runafewfinalizers (lua_State *L) {

```
....
850.    g->gcfinnum = (!g->tobefnz) ? 0  /* nothing more to finalize? */
```

NULL Pointer Dereference\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=141
Status	New

The variable declared in 0 at wcc/lstrlib.c in line 603 is not initialized when it is used by nrep at wcc/lstrlib.c in line 603.

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c
Line	613	613
Object	0	nrep

Code Snippet
File Name wcc/lstrlib.c
Method static void prepstate (MatchState *ms, lua_State *L,

```
....
613.      ms->nrep = MAX_SIZET;   /* no limit */
```

NULL Pointer Dereference\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=142
Status	New

The variable declared in 0 at wcc/wcc.c in line 698 is not initialized when it is used by s_bfd at wcc/wcc.c in line 698.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	700	703
Object	0	s_bfd

Code Snippet
File Name wcc/wcc.c
Method msec_t *section_from_addr(ctx_t * ctx, unsigned long int addr)

```
....
700.    msec_t *s = 0;
....
703.      if ((s->s_bfd->vma) && (s->s_bfd->vma <= addr)
```

NULL Pointer Dereference\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=143
Status	New

The variable declared in 0 at wcc/wcc.c in line 698 is not initialized when it is used by s_bfd at wcc/wcc.c in line 698.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	700	703
Object	0	s_bfd

Code Snippet
File Name wcc/wcc.c
Method msec_t *section_from_addr(ctx_t * ctx, unsigned long int addr)

```
....
700.    msec_t *s = 0;
....
703.      if ((s->s_bfd->vma) && (s->s_bfd->vma <= addr)
```

NULL Pointer Dereference\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=144
Status	New

The variable declared in 0 at wcc/wcc.c in line 698 is not initialized when it is used by s_bfd at wcc/wcc.c in line 698.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	700	704
Object	0	s_bfd

Code Snippet
File Name wcc/wcc.c
Method msec_t *section_from_addr(ctx_t * ctx, unsigned long int addr)

```
....
700.     msec_t *s = 0;
....
704.             && (s->s_bfd->vma + s->s_bfd->size > addr)) {
```

NULL Pointer Dereference\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=145
Status	New

The variable declared in 0 at wcc/wcc.c in line 698 is not initialized when it is used by s_bfd at wcc/wcc.c in line 698.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	700	704
Object	0	s_bfd

Code Snippet	
File Name	wcc/wcc.c
Method	msec_t *section_from_addr(ctx_t * ctx, unsigned long int addr)

```
....
700.     msec_t *s = 0;
....
704.             && (s->s_bfd->vma + s->s_bfd->size > addr)) {
```

NULL Pointer Dereference\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=146
Status	New

The variable declared in 0 at wcc/wcc.c in line 768 is not initialized when it is used by name at wcc/wcc.c in line 768.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	771	785
Object	0	name

Code Snippet	
File Name	wcc/wcc.c
Method	char *sec_name_from_index_after_strip(ctx_t * ctx, unsigned int index)

```
....
771.    msec_t *s = 0;
....
785.        return s->name;
```

NULL Pointer Dereference\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=147
Status	New

The variable declared in 0 at wcc/wcc.c in line 1656 is not initialized when it is used by s_elf at wcc/wcc.c in line 1656.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1658	1689
Object	0	s_elf

Code Snippet

File Name	wcc/wcc.c
Method	int fixup_text(ctx_t * ctx)

```
....
1658.    msec_t *s = 0;
....
1689.        ftruncate(ctx->fdout, s->outoffset + s->s_elf->sh_size);
```

NULL Pointer Dereference\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=148
Status	New

The variable declared in 0 at wcc/wcc.c in line 1656 is not initialized when it is used by s_elf at wcc/wcc.c in line 1656.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1658	1670
Object	0	s_elf

Code Snippet

File Name	wcc/wcc.c
Method	int fixup_text(ctx_t * ctx)

```
....
1658.    msec_t *s = 0;
....
1670.                s->s_elf->sh_size, newsz, s->len);
```

NULL Pointer Dereference\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=149
Status	New

The variable declared in 0 at wcc/wcc.c in line 1700 is not initialized when it is used by name at wcc/wcc.c in line 1518.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1702	1536
Object	0	name

Code Snippet

File Name wcc/wcc.c

Method static unsigned int parse_relocations(ctx_t * ctx)

```
....
1702.    msec_t *s = 0;
```

File Name wcc/wcc.c

Method static int parse_reloc(ctx_t * ctx, msec_t * s)

```
....
1536.              shdr->sh_entsize, entszfromname(".rela.dyn"), s-
>name);
```

NULL Pointer Dereference\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=150
Status	New

The variable declared in 0 at wcc/wcc.c in line 1700 is not initialized when it is used by name at wcc/wcc.c in line 1518.

	Source	Destination

File	wcc/wcc.c	wcc/wcc.c
Line	1702	1543
Object	0	name

Code Snippet
File Name wcc/wcc.c
Method static unsigned int parse_relocations(ctx_t * ctx)

```
....
1702.     msec_t *s = 0;
```

▼

File Name wcc/wcc.c

Method static int parse_reloc(ctx_t * ctx, msec_t * s)

```
....
1543.              shdr->sh_entsize, entszfromname(".rel.dyn"), s->name);
```

NULL Pointer Dereference\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=151
Status	New

The variable declared in 0 at wcc/wcc.c in line 1700 is not initialized when it is used by name at wcc/wcc.c in line 1518.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1702	1551
Object	0	name

Code Snippet
File Name wcc/wcc.c
Method static unsigned int parse_relocations(ctx_t * ctx)

```
....
1702.     msec_t *s = 0;
```

▼

File Name wcc/wcc.c

Method static int parse_reloc(ctx_t * ctx, msec_t * s)

```
....
1551.            shdr->sh_size, entszfromname(".rela.dyn"), s->name);
```

NULL Pointer Dereference\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=152
Status	New

The variable declared in 0 at wcc/wcc.c in line 1700 is not initialized when it is used by name at wcc/wcc.c in line 1518.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1702	1558
Object	0	name

Code Snippet

File Name wcc/wcc.c

Method static unsigned int parse_relocations(ctx_t * ctx)

```
....
1702.     msec_t *s = 0;
```

File Name wcc/wcc.c

Method static int parse_reloc(ctx_t * ctx, msec_t * s)

```
....
1558.            shdr->sh_size, entszfromname(".rel.dyn"), s->name);
```

NULL Pointer Dereference\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=153
Status	New

The variable declared in 0 at wcc/wcc.c in line 1700 is not initialized when it is used by name at wcc/wcc.c in line 1518.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1702	1563
Object	0	name

Code Snippet

File Name wcc/wcc.c

Method static unsigned int parse_relocations(ctx_t * ctx)

```
....
1702.     msec_t *s = 0;
```

File Name wcc/wcc.c

Method static int parse_reloc(ctx_t * ctx, msec_t * s)

```
....
1563.       printf("\t%s\tsize:%u\t%lu relocations\n", s->name, sz,
```

NULL Pointer Dereference\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=154
Status	New

The variable declared in 0 at wcc/wcc.c in line 1700 is not initialized when it is used by s_elf at wcc/wcc.c in line 1700.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1702	1710
Object	0	s_elf

Code Snippet

File Name wcc/wcc.c

Method static unsigned int parse_relocations(ctx_t * ctx)

```
....
1702.     msec_t *s = 0;
....
1710.       } else if ((s->s_elf) && (s->s_elf->sh_type == SHT_REL)) {
      // relocations without addends
```

NULL Pointer Dereference\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=155
Status	New

The variable declared in 0 at wcc/wcc.c in line 1700 is not initialized when it is used by s_elf at wcc/wcc.c in line 1700.

	Source	Destination

File	wcc/wcc.c	wcc/wcc.c
Line	1702	1708
Object	0	s_elf

Code Snippet
File Name wcc/wcc.c
Method static unsigned int parse_relocations(ctx_t * ctx)

```
....
1702.    msec_t *s = 0;
....
1708.      if ((s->s_elf) && (s->s_elf->sh_type == SHT_RELA)) {   // 
relocations with addends
```

NULL Pointer Dereference\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=156
Status	New

The variable declared in 0 at wcc/wcc.c in line 1868 is not initialized when it is used by s_elf at wcc/wcc.c in line 1868.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1872	1911
Object	0	s_elf

Code Snippet
File Name wcc/wcc.c
Method static unsigned int write_shdrs(ctx_t * ctx)

```
....
1872.    msec_t *s = 0;
....
1911.      write(ctx->fdout, s->s_elf, sizeof(Elf_Shdr));
```

NULL Pointer Dereference\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=157
Status	New

The variable declared in 0 at wcc/wcc.c in line 1868 is not initialized when it is used by name at wcc/wcc.c in line 1868.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1872	1902
Object	0	name

Code Snippet
File Name wcc/wcc.c
Method static unsigned int write_shdrs(ctx_t * ctx)

```
....
1872.    msec_t *s = 0;
....
1902.       memcpy(ctx->strndx + ctx->strndx_len, s->name, strlen(s->name) + 1);       // do copy the final "\x00"
```

NULL Pointer Dereference\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=158
Status	New

The variable declared in 0 at wcc/wcc.c in line 1868 is not initialized when it is used by name at wcc/wcc.c in line 1868.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1872	1902
Object	0	name

Code Snippet
File Name wcc/wcc.c
Method static unsigned int write_shdrs(ctx_t * ctx)

```
....
1872.    msec_t *s = 0;
....
1902.       memcpy(ctx->strndx + ctx->strndx_len, s->name, strlen(s->name) + 1);       // do copy the final "\x00"
```

NULL Pointer Dereference\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=159
Status	New

The variable declared in 0 at wcc/wcc.c in line 2402 is not initialized when it is used by outoffset at wcc/wcc.c in line 2110.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2404	2115
Object	0	outoffset

Code Snippet
File Name wcc/wcc.c
Method int copy_body(ctx_t * ctx)

```
....
2404.     msec_t *s = 0;
```

File Name wcc/wcc.c
Method static int write_section(ctx_t * ctx, msec_t * m)

```
....
2115.     lseek(ctx->fdout, m->outoffset, SEEK_SET);
```

NULL Pointer Dereference\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=160
Status	New

The variable declared in 0 at wcc/wcc.c in line 2576 is not initialized when it is used by len at wcc/wcc.c in line 2576.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2578	2585
Object	0	len

Code Snippet
File Name wcc/wcc.c
Method int print_msec(ctx_t * ctx)

```
....
2578.     msec_t *ms = 0;
....
2585.       printf("%s  %lu\n", ms->name, ms->len);
```

NULL Pointer Dereference\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=161
Status	New

The variable declared in 0 at wcc/wcc.c in line 2576 is not initialized when it is used by name at wcc/wcc.c in line 2576.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2578	2585
Object	0	name

Code Snippet

File Name	wcc/wcc.c
Method	int print_msec(ctx_t * ctx)

```
....
2578.     msec_t *ms = 0;
....
2585.       printf("%s  %lu\n", ms->name, ms->len);
```

NULL Pointer Dereference\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=162
Status	New

The variable declared in 0 at wcc/wcc.c in line 3305 is not initialized when it is used by detail at wcc/wcc.c in line 3229.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3309	3236
Object	0	detail

Code Snippet

File Name	wcc/wcc.c
Method	int analyze_text(ctx_t * ctx, char *data, unsigned int datalen,

```
....
3309.     cs_insn *insn = 0;
```

▼

File Name	wcc/wcc.c

Method	static void parse_text_data_reloc(ctx_t * ctx, csh ud, cs_mode mode,

```
....
3236.     if (ins->detail == NULL)
```

NULL Pointer Dereference\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=163
Status	New

The variable declared in 0 at wcc/wcc.c in line 3305 is not initialized when it is used by address at wcc/wcc.c in line 2990.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3309	3111
Object	0	address

Code Snippet

File Name	wcc/wcc.c
Method	int analyze_text(ctx_t * ctx, char *data, unsigned int datalen,

```
....
3309.     cs_insn *insn = 0;
```

▼

File Name	wcc/wcc.c
Method	static int create_text_data_reloc(ctx_t * ctx, cs_insn * ins, msec_t * m,

```
....
3111.                ins->address - textvma + wheretowrite,
```

NULL Pointer Dereference\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=164
Status	New

The variable declared in 0 at wcc/wcc.c in line 3305 is not initialized when it is used by detail at wcc/wcc.c in line 2883.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c

Line	3309	2890
Object	0	detail

Code Snippet
File Name wcc/wcc.c
Method int analyze_text(ctx_t * ctx, char *data, unsigned int datalen,

```
....
3309.     cs_insn *insn = 0;
```

File Name wcc/wcc.c

Method static void print_insn_detail(ctx_t * ctx, csh handle, cs_mode mode,

```
....
2890.     if (ins->detail == NULL)
```

NULL Pointer Dereference\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=165
Status	New

The variable declared in 0 at wcc/wcc.c in line 3305 is not initialized when it is used by address at wcc/wcc.c in line 2883.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3309	2894
Object	0	address

Code Snippet
File Name wcc/wcc.c
Method int analyze_text(ctx_t * ctx, char *data, unsigned int datalen,

```
....
3309.     cs_insn *insn = 0;
```

File Name wcc/wcc.c

Method static void print_insn_detail(ctx_t * ctx, csh handle, cs_mode mode,

```
....
2894.     printf("\tAddress: %lu\n", ins->address);
```

NULL Pointer Dereference\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=166
Status	New

The variable declared in 0 at wcc/wcc.c in line 3305 is not initialized when it is used by size at wcc/wcc.c in line 2883.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3309	2895
Object	0	size

Code Snippet

File Name wcc/wcc.c

Method int analyze_text(ctx_t * ctx, char *data, unsigned int datalen,

```
....
3309.     cs_insn *insn = 0;
```

File Name wcc/wcc.c

Method static void print_insn_detail(ctx_t * ctx, csh handle, cs_mode mode,

```
....
2895.     printf("\tInstruction Length: %u\n", ins->size);
```

NULL Pointer Dereference\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=167
Status	New

The variable declared in 0 at wcc/wcc.c in line 3470 is not initialized when it is used by name at wcc/wcc.c in line 3470.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3472	3494
Object	0	name

Code Snippet

File Name wcc/wcc.c

Method int strip_binary_reloc(ctx_t * ctx)

```
....
3472.    msec_t *s = 0, *tmp = 0;
....
3494.        rm_section(ctx, s->name);
```

NULL Pointer Dereference\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=168
Status	New

The variable declared in 0 at wcc/wcc.c in line 3470 is not initialized when it is used by name at wcc/wcc.c in line 3470.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3472	3491
Object	0	name

Code Snippet	
File Name	wcc/wcc.c
Method	int strip_binary_reloc(ctx_t * ctx)

```
....
3472.    msec_t *s = 0, *tmp = 0;
....
3491.          printf(" * %s\n", s->name);
```

NULL Pointer Dereference\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=169
Status	New

The variable declared in 0 at wcc/wsh.c in line 938 is not initialized when it is used by htype at wcc/wsh.c in line 938.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	940	947
Object	0	htype

Code Snippet	
File Name	wcc/wsh.c
Method	int empty_symbols(void)

```
....
940.        symbols_t *s = 0, *stmp = 0;
....
947.                    free(s->htype);
```

NULL Pointer Dereference\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=170
Status	New

The variable declared in 0 at wcc/wsh.c in line 938 is not initialized when it is used by libname at wcc/wsh.c in line 938.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	940	946
Object	0	libname

Code Snippet
File Name wcc/wsh.c
Method int empty_symbols(void)

```
....
940.        symbols_t *s = 0, *stmp = 0;
....
946.                    free(s->libname);
```

NULL Pointer Dereference\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=171
Status	New

The variable declared in 0 at wcc/wsh.c in line 938 is not initialized when it is used by hbind at wcc/wsh.c in line 938.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	940	945
Object	0	hbind

Code Snippet
File Name wcc/wsh.c
Method int empty_symbols(void)

```
....
940.         symbols_t *s = 0, *stmp = 0;
....
945.                     free(s->hbind);
```

NULL Pointer Dereference\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=172
Status	New

The variable declared in 0 at wcc/wsh.c in line 938 is not initialized when it is used by symbol at wcc/wsh.c in line 938.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	940	944
Object	0	symbol

Code Snippet
File Name wcc/wsh.c
Method int empty_symbols(void)

```
....
940.         symbols_t *s = 0, *stmp = 0;
....
944.                     free(s->symbol);
```

NULL Pointer Dereference\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=173
Status	New

The variable declared in 0 at wcc/wsh.c in line 958 is not initialized when it is used by perms at wcc/wsh.c in line 958.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	960	967
Object	0	perms

Code Snippet
File Name wcc/wsh.c
Method int empty_phdrs(void)

```
....
960.        segments_t *s = 0, *stmp = 0;
....
967.                    free(s->perms);
```

NULL Pointer Dereference\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=174
Status	New

The variable declared in 0 at wcc/wsh.c in line 958 is not initialized when it is used by libname at wcc/wsh.c in line 958.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	960	966
Object	0	libname

Code Snippet
File Name wcc/wsh.c
Method int empty_phdrs(void)

```
....
960.        segments_t *s = 0, *stmp = 0;
....
966.                    free(s->libname);
```

NULL Pointer Dereference\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=175
Status	New

The variable declared in 0 at wcc/wsh.c in line 958 is not initialized when it is used by type at wcc/wsh.c in line 958.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	960	965
Object	0	type

Code Snippet
File Name wcc/wsh.c
Method int empty_phdrs(void)

```
....
960.        segments_t *s = 0, *stmp = 0;
....
965.                    free(s->type);
```

NULL Pointer Dereference\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=176
Status	New

The variable declared in 0 at wcc/wsh.c in line 978 is not initialized when it is used by perms at wcc/wsh.c in line 978.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	980	987
Object	0	perms

Code Snippet
File Name wcc/wsh.c
Method int empty_shdrs(void)

```
....
980.        sections_t *s = 0, *stmp = 0;
....
987.                    free(s->perms);
```

NULL Pointer Dereference\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=177
Status	New

The variable declared in 0 at wcc/wsh.c in line 978 is not initialized when it is used by libname at wcc/wsh.c in line 978.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	980	986
Object	0	libname

Code Snippet
File Name wcc/wsh.c
Method int empty_shdrs(void)

```
....
980.        sections_t *s = 0, *stmp = 0;
....
986.                    free(s->libname);
```

NULL Pointer Dereference\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=178
Status	New

The variable declared in 0 at wcc/wsh.c in line 978 is not initialized when it is used by name at wcc/wsh.c in line 978.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	980	985
Object	0	name

Code Snippet	
File Name	wcc/wsh.c
Method	int empty_shdrs(void)

```
....
980.        sections_t *s = 0, *stmp = 0;
....
985.                    free(s->name);
```

NULL Pointer Dereference\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=179
Status	New

The variable declared in 0 at wcc/wsh.c in line 997 is not initialized when it is used by name at wcc/wsh.c in line 997.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	999	1004
Object	0	name

Code Snippet	
File Name	wcc/wsh.c
Method	int empty_eps(void)

```
....
999.        eps_t *s = 0, *stmp = 0;
....
1004.                   free(s->name);
```

NULL Pointer Dereference\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=180
Status	New

The variable declared in 0 at wcc/wsh.c in line 1014 is not initialized when it is used by type at wcc/wsh.c in line 1014.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1017	1057
Object	0	type

Code Snippet
File Name wcc/wsh.c
Method int print_phdrs(void)

```
....
1017.       segments_t *s = 0, *stmp = 0;
....
1057.                                   s->perms, s->size, s->libname, s-
>type);
```

NULL Pointer Dereference\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=181
Status	New

The variable declared in 0 at wcc/wsh.c in line 1014 is not initialized when it is used by size at wcc/wsh.c in line 1014.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1017	1057
Object	0	size

Code Snippet
File Name wcc/wsh.c

Method	int print_phdrs(void)

```
....
1017.       segments_t *s = 0, *stmp = 0;
....
1057.                               s->perms, s->size, s->libname, s-
>type);
```

NULL Pointer Dereference\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=182
Status	New

The variable declared in 0 at wcc/wsh.c in line 1014 is not initialized when it is used by perms at wcc/wsh.c in line 1014.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1017	1057
Object	0	perms

Code Snippet	
File Name	wcc/wsh.c
Method	int print_phdrs(void)

```
....
1017.       segments_t *s = 0, *stmp = 0;
....
1057.                               s->perms, s->size, s->libname, s-
>type);
```

NULL Pointer Dereference\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=183
Status	New

The variable declared in 0 at wcc/wsh.c in line 1014 is not initialized when it is used by addr at wcc/wsh.c in line 1014.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1017	1056
Object	0	addr

Code Snippet

File Name	wcc/wsh.c
Method	int print_phdrs(void)

```
....
1017.        segments_t *s = 0, *stmp = 0;
....
1056.                          printf("%012lx-%012lx\t%s\t%lu\t%s\t%s\n",
s->addr, s->addr + s->size,
```

NULL Pointer Dereference\Path 45:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=184
Status	New

The variable declared in 0 at wcc/wsh.c in line 1014 is not initialized when it is used by size at wcc/wsh.c in line 1014.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1017	1056
Object	0	size

Code Snippet

File Name	wcc/wsh.c
Method	int print_phdrs(void)

```
....
1017.        segments_t *s = 0, *stmp = 0;
....
1056.                          printf("%012lx-%012lx\t%s\t%lu\t%s\t%s\n",
s->addr, s->addr + s->size,
```

NULL Pointer Dereference\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=185
Status	New

The variable declared in 0 at wcc/wsh.c in line 1014 is not initialized when it is used by addr at wcc/wsh.c in line 1014.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1017	1056

Object	0	addr

Code Snippet
File Name wcc/wsh.c
Method int print_phdrs(void)

```
....
1017.        segments_t *s = 0, *stmp = 0;
....
1056.                          printf("%012lx-%012lx\t%s\t%lu\t%s\t%s\n",
s->addr, s->addr + s->size,
```

NULL Pointer Dereference\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=186
Status	New

The variable declared in 0 at wcc/wsh.c in line 1014 is not initialized when it is used by size at wcc/wsh.c in line 1014.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1017	1047
Object	0	size

Code Snippet
File Name wcc/wsh.c
Method int print_phdrs(void)

```
....
1017.        segments_t *s = 0, *stmp = 0;
....
1047.                    if(s->size == 0){
```

NULL Pointer Dereference\Path 48:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=187
Status	New

The variable declared in 0 at wcc/wsh.c in line 1014 is not initialized when it is used by libname at wcc/wsh.c in line 1014.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c

Line	1017	1057
Object	0	libname

Code Snippet
File Name wcc/wsh.c
Method int print_phdrs(void)

```
....
1017.        segments_t *s = 0, *stmp = 0;
....
1057.                                   s->perms, s->size, s->libname, s-
>type);
```

NULL Pointer Dereference\Path 49:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=188
Status	New

The variable declared in 0 at wcc/wsh.c in line 1329 is not initialized when it is used by name at wcc/wsh.c in line 1329.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1332	1381
Object	0	name

Code Snippet
File Name wcc/wsh.c
Method int print_shdrs(void)

```
....
1332.        sections_t *s = 0, *stmp = 0;
....
1381.                              printf("%012lx-
%012lx\t%s\t%lu\t%s\t%25s\t%s\t%s\n", s->addr, s->addr + s->size, s-
>perms, s->size, s->libname, s->name, segmenttype, segmentperms);
```

NULL Pointer Dereference\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=189
Status	New

The variable declared in 0 at wcc/wsh.c in line 1329 is not initialized when it is used by size at wcc/wsh.c in line 1329.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1332	1381
Object	0	size

Code Snippet
File Name wcc/wsh.c
Method int print_shdrs(void)

```
....
1332.         sections_t *s = 0, *stmp = 0;
....
1381.                           printf("%012lx-
%012lx\t%s\t%lu\t%s\t%25s\t%s\t%s\n", s->addr, s->addr + s->size, s-
>perms, s->size, s->libname, s->name, segmenttype, segmentperms);
```

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=1050048&projectid=50038&pathid=3
Status	New

The fixup_strtab_and_symtab method calls the sprintf function, at line 1603 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	1643	1643
Object	sprintf	sprintf

Code Snippet
File Name wcc/wcc.c
Method int fixup_strtab_and_symtab(ctx_t * ctx)

```
....
1643.           sprintf(globalstrtab + globalstrtablen, "old_%s", sname);
```

Unchecked Return Value\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=4
Status	New

The open_target method calls the sprintf function, at line 2348 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	2364	2364
Object	sprintf	sprintf

Code Snippet
File Name wcc/wcc.c
Method int open_target(ctx_t * ctx)

```
....
2364.       sprintf(newname, "a.out");
```

Unchecked Return Value\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=5
Status	New

The internal_function_store method calls the snprintf function, at line 3200 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	3207	3207
Object	snprintf	snprintf

Code Snippet
File Name wcc/wcc.c
Method int internal_function_store(ctx_t * ctx, unsigned long long int addr)

```
....
3207.     snprintf(buff, 200, "internal_%08llx", addr);
```

Unchecked Return Value\Path 4:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=6
Status	New

The print_maps method calls the sprintf function, at line 3671 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	3676	3676
Object	sprintf	sprintf

Code Snippet

File Name	wcc/wcc.c
Method	int print_maps(void)

```
....
3676.     sprintf(cmd, "cat /proc/%u/maps", getpid());
```

Unchecked Return Value\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=7
Status	New

The ptoh method calls the snprintf function, at line 152 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	154	154
Object	snprintf	snprintf

Code Snippet

File Name	wcc/wsh.c
Method	int ptoh(int perms, char hperms[])

```
....
154.         snprintf(hperms, 5, "%s%s%s", (perms & 0x04) ? "r" : "-",
(perms & 0x02) ? "w" : "-", (perms & 0x01) ? "x" : "-");
```

Unchecked Return Value\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=500

	38&pathid=8
Status	New

The *decode_flags method calls the strdup function, at line 560 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	583	583
Object	strdup	strdup

Code Snippet

File Name	wcc/wsh.c
Method	char *decode_flags(unsigned int flags)

```
....
583.        return strdup(message);
```

Unchecked Return Value\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=9
Status	New

The *decode_type method calls the snprintf function, at line 589 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	637	637
Object	snprintf	snprintf

Code Snippet

File Name	wcc/wsh.c
Method	char *decode_type(unsigned int type)

```
....
637.            snprintf(ret, 199, "Unknown: 0x%x\n", type);
```

Unchecked Return Value\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=10
Status	New

The scan_section method calls the snprintf function, at line 765 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	772	772
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method void scan_section(Elf_Shdr * shdr, char *strTab, int shnum, char *fname, unsigned long int baseaddr)

```
....
772.                snprintf(hperms, 5, "%s%s%s", (shdr[i].sh_flags & 0x02) ? "r" : "-", (shdr[i].sh_flags & 0x01) ? "w" : "-", (shdr[i].sh_flags & 0x04) ? "x" : "-");
```

Unchecked Return Value\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=11
Status	New

The man method calls the snprintf function, at line 1465 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	1473	1473
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method int man(lua_State * L)

```
....
1473.               snprintf(cmd, 254, "man %s", (char*) arg);      // Obvious injection. We don't care
```

Unchecked Return Value\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=12
Status	New

The run_shell method calls the snprintf function, at line 1673 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	1699	1699
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method int run_shell(lua_State * L)

```
....
1699.           snprintf(SHELL_HISTORY, 1023, "%s/%s", getenv("HOME"),
SHELL_HISTORY_NAME);
```

Unchecked Return Value\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=13
Status	New

The run_shell method calls the snprintf function, at line 1673 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	1709	1709
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method int run_shell(lua_State * L)

```
....
1709.           snprintf(shell_prompt, sizeof(shell_prompt), "> ");
```

Unchecked Return Value\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=14
Status	New

The libcall method calls the snprintf function, at line 2059 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	2397	2397
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method static int libcall(lua_State * L)

```
....
2397.               snprintf(argname, 9, "arg%u", j);
```

Unchecked Return Value\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=15
Status	New

The scan_syms method calls the snprintf function, at line 2512 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	2605	2605
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method void scan_syms(char *dynstr, Elf_Sym * sym, unsigned long int sz, char *libname)

```
....
2605.                               snprintf(newname, 1023, "reflect_%s",
symname);
```

Unchecked Return Value\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=16
Status	New

The scan_syms method calls the snprintf function, at line 2512 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	2614	2614
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method void scan_syms(char *dynstr, Elf_Sym * sym, unsigned long int sz, char *libname)

```
....
2614.                         snprintf(luacmd, 2047, "function %s (a, b, c, d, e, f, g, h) j,k = libcall(%s, a, b, c, d, e, f, g, h); return j, k; end\n", symname, newname);
```

Unchecked Return Value\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=17
Status	New

The print_procmap method calls the snprintf function, at line 2848 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	2856	2856
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method int print_procmap(unsigned int pid)

```
....
2856.        snprintf(path, 99, "/proc/%u/maps", pid);
```

Unchecked Return Value\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=18
Status	New

The btr_enable method calls the snprintf function, at line 3050 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	3057	3057
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method void btr_enable(int procnum)

```
....
3057.        snprintf(cpupath, 199, "/dev/cpu/%d/msr", procnum);
```

Unchecked Return Value\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=19
Status	New

The btr_disable method calls the snprintf function, at line 3071 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	3078	3078
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method void btr_disable(int procnum)

```
....
3078.        snprintf(cpupath, 199, "/dev/cpu/%d/msr", procnum);
```

Unchecked Return Value\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=20
Status	New

The sighandler method calls the snprintf function, at line 3556 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	3599	3599
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method void sighandler(int signal, siginfo_t * s, void *ptr)

```
....
3599.            snprintf(defsicode, 199, "Error code %d", s->si_code);
```

Unchecked Return Value\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=21
Status	New

The declare_internals method calls the snprintf function, at line 4432 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	4450	4450
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method void declare_internals(void)

```
....
4450.       snprintf(luacmd, 1023, "function %s (a, b, c, d, e, f, g, h)
j,k = libcall(%s, a, b, c, d, e, f, g, h); return j, k; end\n",
"hexdump", "lhexdump");
```

Unchecked Return Value\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=22
Status	New

The declare_internals method calls the snprintf function, at line 4432 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	4453	4453
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method void declare_internals(void)

```
....
4453.       snprintf(luacmd, 1023, "function %s (a, b, c, d, e, f, g, h)
j,k = libcall(%s, a, string.len(a), c, d, e, f, g, h); return j, k;
end\n", "hex", "lhexdump");
```

Unchecked Return Value\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=23
Status	New

The declare_internals method calls the snprintf function, at line 4432 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	4456	4456
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method void declare_internals(void)

```
....
4456.       snprintf(luacmd, 1023, "function %s (a, b, c, d, e, f, g, h)
j,k = libcall(%s, a, b, c, d, e, f, g, h); return j, k; end\n",
"execlib", "lexeclib");
```

Unchecked Return Value\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=24
Status	New

The declare_internals method calls the snprintf function, at line 4432 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	4459	4459
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method void declare_internals(void)

```
....
4459.        snprintf(luacmd, 1023, "function %s (a, b, c, d, e, f, g, h)
j,k = libcall(%s, a, b, c, d, e, f, g, h); return j, k; end\n",
"disasm", "ldisasm");
```

Unchecked Return Value\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=25
Status	New

The declare_internals method calls the snprintf function, at line 4432 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	4462	4462
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method void declare_internals(void)

```
....
4462.        snprintf(luacmd, 1023, "function %s (a, b, c, d, e, f, g, h)
j,k = libcall(%s, a, b, c, d, e, f, g, h); return j, k; end\n", "deref",
"lderef");
```

Unchecked Return Value\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=26

Status	New

The declare_internals method calls the snprintf function, at line 4432 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	4465	4465
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method void declare_internals(void)

```
....
4465.        snprintf(luacmd, 1023, "function %s (a, b, c, d, e, f, g, h)
j,k = libcall(%s, a, b, c, d, e, f, g, h); return j, k; end\n",
"strace", "lstrace");
```

Unchecked Return Value\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=27
Status	New

The declare_internals method calls the snprintf function, at line 4432 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	4468	4468
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method void declare_internals(void)

```
....
4468.        snprintf(luacmd, 1023, "function %s (a, b, c, d, e, f, g, h)
j,k = libcall(%s, a, b, c, d, e, f, g, h); return j, k; end\n",
"script", "lscript");
```

Unchecked Return Value\Path 26:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=28
Status	New

The run_script method calls the snprintf function, at line 4585 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	4599	4599
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method int run_script(char *name)

```
....
4599.                snprintf(myerror, 199, "error %d : %s", err, 
lua_strerror(err));
```

Unchecked Return Value\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=29
Status	New

The load_home_user_file method calls the snprintf function, at line 4661 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	4674	4674
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method int load_home_user_file(char *fname)

```
....
4674.        snprintf(pathname, 254, "%s/%s", getenv("HOME"), fname);
```

Unchecked Return Value\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=500

	38&pathid=30
Status	New

The attempt_to_patch method calls the snprintf function, at line 4929 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	4962	4962
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method int attempt_to_patch(char *libname)

```
....
4962.        snprintf(tmp_dirname, 19, "/tmp/.wsh-%u", getpid());
```

Unchecked Return Value\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=31
Status	New

The attempt_to_patch method calls the snprintf function, at line 4929 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	4973	4973
Object	snprintf	snprintf

Code Snippet
File Name wcc/wsh.c
Method int attempt_to_patch(char *libname)

```
....
4973.        snprintf(outlib, 299, "/%s/%s", tmp_dirname,
basename(libname));
```

Unchecked Return Value\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=500

	38&pathid=32
Status	New

The add_symaddr method calls the sa function, at line 418 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	443	443
Object	sa	sa

Code Snippet
File Name wcc/wcc.c
Method void add_symaddr(ctx_t * ctx, const char *name, int addr, char symclass)

```
....
443.     sa = (struct symaddr *) malloc(sizeof(struct symaddr));
```

Unchecked Return Value\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=33
Status	New

The add_symaddr method calls the name function, at line 418 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	445	445
Object	name	name

Code Snippet
File Name wcc/wcc.c
Method void add_symaddr(ctx_t * ctx, const char *name, int addr, char symclass)

```
....
445.     sa->name = strdup(name);
```

Unchecked Return Value\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=34
Status	New

The *alloc_phdr method calls the p function, at line 984 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	990	990
Object	p	p

Code Snippet

File Name	wcc/wcc.c
Method	mseg_t *alloc_phdr(msec_t * ms)

```
....
990.     p = calloc(1, sizeof(mseg_t));
```

Unchecked Return Value\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=35
Status	New

The rd_phdrs method calls the p function, at line 1129 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	1144	1144
Object	p	p

Code Snippet

File Name	wcc/wcc.c
Method	static unsigned int rd_phdrs(ctx_t * ctx)

```
....
1144.     p = calloc(1, sb.st_size);
```

Unchecked Return Value\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=36
Status	New

The create_section_symbols method calls the sym function, at line 1765 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	1773	1773
Object	sym	sym

Code Snippet

File Name	wcc/wcc.c
Method	static int create_section_symbols(ctx_t * ctx)

```
....
1773.     sym = calloc(1, sizeof(Elf_Sym));
```

Unchecked Return Value\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=37
Status	New

The write_shdrs method calls the shdr function, at line 1868 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	1920	1920
Object	shdr	shdr

Code Snippet

File Name	wcc/wcc.c
Method	static unsigned int write_shdrs(ctx_t * ctx)

```
....
1920.     shdr = calloc(1, sizeof(Elf_Shdr));
```

Unchecked Return Value\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=38
Status	New

The write_shdrs method calls the shdr function, at line 1868 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c

Line	1949	1949
Object	shdr	shdr

Code Snippet
File Name wcc/wcc.c
Method static unsigned int write_shdrs(ctx_t * ctx)

```
....
1949.     shdr = calloc(1, sizeof(Elf_Shdr));
```

Unchecked Return Value\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=39
Status	New

The write_shdrs method calls the shdr function, at line 1868 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	1979	1979
Object	shdr	shdr

Code Snippet
File Name wcc/wcc.c
Method static unsigned int write_shdrs(ctx_t * ctx)

```
....
1979.     shdr = calloc(1, sizeof(Elf_Shdr));
```

Unchecked Return Value\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=40
Status	New

The write_shdrs method calls the shdr function, at line 1868 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	2009	2009
Object	shdr	shdr

Code Snippet	
File Name	wcc/wcc.c
Method	static unsigned int write_shdrs(ctx_t * ctx)

```
....
2009.     shdr = calloc(1, sizeof(Elf_Shdr));
```

Unchecked Return Value\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=41
Status	New

The open_target method calls the newname function, at line 2348 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	2363	2363
Object	newname	newname

Code Snippet	
File Name	wcc/wcc.c
Method	int open_target(ctx_t * ctx)

```
....
2363.       newname = calloc(1, strlen(ctx->binname) + 20);
```

Unchecked Return Value\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=42
Status	New

The open_target method calls the p function, at line 2348 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	2380	2380
Object	p	p

Code Snippet	
File Name	wcc/wcc.c

Method	int open_target(ctx_t * ctx)

```
....
2380.     p = calloc(1, sb.st_size);
```

Unchecked Return Value\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=43
Status	New

The read_section method calls the buf function, at line 2495 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	2516	2516
Object	buf	buf

Code Snippet

File Name	wcc/wcc.c
Method	static int read_section(ctx_t * ctx, asection * s)

```
....
2516.     buf = calloc(1, wantedsz);
```

Unchecked Return Value\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=44
Status	New

The read_section method calls the buf function, at line 2495 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	2534	2534
Object	buf	buf

Code Snippet

File Name	wcc/wcc.c
Method	static int read_section(ctx_t * ctx, asection * s)

```
....
2534.        buf = realloc(buf, 0);
```

Unchecked Return Value\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=45
Status	New

The save_global_import method calls the g function, at line 2701 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	2716	2716
Object	g	g

Code Snippet

File Name wcc/wcc.c

Method int save_global_import(ctx_t * ctx, char *sname, msec_t * sec, Elf_Rela * r, unsigned int sindex)

```
....
2716.     g = calloc(1, sizeof(gimport_t));
```

Unchecked Return Value\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=46
Status	New

The save_global_import method calls the sname function, at line 2701 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	2717	2717
Object	sname	sname

Code Snippet

File Name wcc/wcc.c

Method int save_global_import(ctx_t * ctx, char *sname, msec_t * sec, Elf_Rela * r, unsigned int sindex)

```
....
2717.     g->sname = strdup(sname);
```

Unchecked Return Value\Path 45:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=47
Status	New

The save_global_import method calls the rnew function, at line 2701 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	2721	2721
Object	rnew	rnew

Code Snippet
File Name wcc/wcc.c
Method int save_global_import(ctx_t * ctx, char *sname, msec_t * sec, Elf_Rela * r, unsigned int sindex)

```
....
2721.     rnew = calloc(1, sizeof(Elf_Rela));
```

Unchecked Return Value\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=48
Status	New

The save_reloc method calls the rout function, at line 2755 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	2767	2767
Object	rout	rout

Code Snippet
File Name wcc/wcc.c
Method int save_reloc(ctx_t * ctx, Elf_Rela * r, unsigned int sindex, int has_addend)

```
....
2767.     rout = calloc(1, sizeof(Elf_Rela));     // Work on a copy of the
relocation instead of the original one
```

Unchecked Return Value\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=49
Status	New

The *ctx_init method calls the strndx function, at line 3684 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	3699	3699
Object	strndx	strndx

Code Snippet
File Name wcc/wcc.c
Method ctx_t *ctx_init(void)

```
....
3699.     ctx->strndx = calloc(1, DEFAULT_STRNDX_SIZE);
```

Unchecked Return Value\Path 48:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=50
Status	New

The ctx_getopt method calls the opt_interp function, at line 3756 of wcc/wcc.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	wcc/wcc.c	wcc/wcc.c
Line	3837	3837
Object	opt_interp	opt_interp

Code Snippet
File Name wcc/wcc.c
Method int ctx_getopt(ctx_t * ctx, int argc, char **argv)

```
....
3837.          ctx->opt_interp = strdup(optarg);
```

Unchecked Return Value\Path 49:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=51
Status	New

The *decode_type method calls the ret function, at line 589 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	636	636
Object	ret	ret

Code Snippet	
File Name	wcc/wsh.c
Method	char *decode_type(unsigned int type)

```
....
636.                ret = calloc(1, 200);
```

Unchecked Return Value\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=52
Status	New

The add_symbol method calls the libname function, at line 681 of wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	692	692
Object	libname	libname

Code Snippet	
File Name	wcc/wsh.c
Method	int add_symbol(char *symbol, char *libname, char *htype, char *hbind, unsigned long value, unsigned int size, unsigned long int addr)

```
....
692.        s->libname = strdup(libname);
```

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=1050048&projectid=50038&pathid=539
Status	New

The system data read by rd_phdrs in the file wcc/wcc.c at line 1129 is potentially exposed by rd_phdrs found in wcc/wcc.c at line 1129.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1140	1140
Object	perror	perror

Code Snippet
File Name wcc/wcc.c
Method static unsigned int rd_phdrs(ctx_t * ctx)

```
....
1140.      perror("stat");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=540
Status	New

The system data read by rd_phdrs in the file wcc/wcc.c at line 1129 is potentially exposed by rd_phdrs found in wcc/wcc.c at line 1129.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1147	1147

Object	perror	perror

Code Snippet
File Name wcc/wcc.c
Method static unsigned int rd_phdrs(ctx_t * ctx)

```
....
1147.       perror("open");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=541
Status	New

The system data read by rd_phdrs in the file wcc/wcc.c at line 1129 is potentially exposed by rd_phdrs found in wcc/wcc.c at line 1129.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1152	1152
Object	perror	perror

Code Snippet
File Name wcc/wcc.c
Method static unsigned int rd_phdrs(ctx_t * ctx)

```
....
1152.       perror("read");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=542
Status	New

The system data read by rd_phdrs in the file wcc/wcc.c at line 1129 is potentially exposed by rd_phdrs found in wcc/wcc.c at line 1129.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1168	1168
Object	perror	perror

Code Snippet
File Name wcc/wcc.c
Method static unsigned int rd_phdrs(ctx_t * ctx)

```
....
1168.        perror("calloc");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=543
Status	New

The system data read by *mk_section in the file wcc/wcc.c at line 1303 is potentially exposed by *mk_section found in wcc/wcc.c at line 1303.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1310	1310
Object	perror	perror

Code Snippet
File Name wcc/wcc.c
Method msec_t *mk_section(void)

```
....
1310.      perror("calloc");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=544
Status	New

The system data read by *mk_section in the file wcc/wcc.c at line 1303 is potentially exposed by *mk_section found in wcc/wcc.c at line 1303.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1316	1316
Object	perror	perror

Code Snippet
File Name wcc/wcc.c
Method msec_t *mk_section(void)

```
....
1316.       perror("calloc");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=545
Status	New

The system data read by mk_ehdr in the file wcc/wcc.c at line 2042 is potentially exposed by mk_ehdr found in wcc/wcc.c at line 2042.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2048	2048
Object	perror	perror

Code Snippet

File Name	wcc/wcc.c
Method	static int mk_ehdr(ctx_t * ctx)

```
....
2048.       perror("calloc");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=546
Status	New

The system data read by open_target in the file wcc/wcc.c at line 2348 is potentially exposed by open_target found in wcc/wcc.c at line 2348.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2356	2356
Object	perror	perror

Code Snippet

File Name	wcc/wcc.c
Method	int open_target(ctx_t * ctx)

```
....
2356.       perror("stat");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=547
Status	New

The system data read by read_section in the file wcc/wcc.c at line 2495 is potentially exposed by read_section found in wcc/wcc.c at line 2495.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2521	2521
Object	perror	perror

Code Snippet
File Name wcc/wcc.c
Method static int read_section(ctx_t * ctx, asection * s)

```
....
2521.       perror("calloc");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=548
Status	New

The system data read by read_section in the file wcc/wcc.c at line 2495 is potentially exposed by read_section found in wcc/wcc.c at line 2495.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2527	2527
Object	perror	perror

Code Snippet
File Name wcc/wcc.c
Method static int read_section(ctx_t * ctx, asection * s)

```
....
2527.        perror("calloc");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=549
Status	New

The system data read by create_text_data_reloc in the file wcc/wcc.c at line 2990 is potentially exposed by create_text_data_reloc found in wcc/wcc.c at line 2990.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3031	3031
Object	perror	perror

Code Snippet
File Name wcc/wcc.c
Method static int create_text_data_reloc(ctx_t * ctx, cs_insn * ins, msec_t * m,

```
....
3031.          perror("calloc");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=550
Status	New

The system data read by create_text_data_reloc in the file wcc/wcc.c at line 2990 is potentially exposed by create_text_data_reloc found in wcc/wcc.c at line 2990.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3071	3071
Object	perror	perror

Code Snippet
File Name wcc/wcc.c
Method static int create_text_data_reloc(ctx_t * ctx, cs_insn * ins, msec_t * m,

```
....
3071.        perror("calloc");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=551
Status	New

The system data read by create_text_data_reloc in the file wcc/wcc.c at line 2990 is potentially exposed by create_text_data_reloc found in wcc/wcc.c at line 2990.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3104	3104
Object	perror	perror

Code Snippet
File Name wcc/wcc.c
Method static int create_text_data_reloc(ctx_t * ctx, cs_insn * ins, msec_t * m,

```
....
3104.        perror("calloc");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=552
Status	New

The system data read by rd_symtab in the file wcc/wcc.c at line 3353 is potentially exposed by rd_symtab found in wcc/wcc.c at line 3353.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3374	3374
Object	perror	perror

Code Snippet
File Name wcc/wcc.c
Method int rd_symtab(ctx_t * ctx)

```
....
3374.       perror("mmap");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=553
Status	New

The system data read by set_sighandlers in the file wcc/wsh.c at line 3644 is potentially exposed by set_sighandlers found in wcc/wsh.c at line 3644.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3653	3653
Object	perror	perror

Code Snippet
File Name   wcc/wsh.c
Method   int set_sighandlers(void)

```
....
3653.             perror("sigaction");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=554
Status	New

The system data read by set_sighandlers in the file wcc/wsh.c at line 3644 is potentially exposed by set_sighandlers found in wcc/wsh.c at line 3644.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3658	3658
Object	perror	perror

Code Snippet
File Name   wcc/wsh.c
Method   int set_sighandlers(void)

```
....
3658.             perror("sigaction");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=555
Status	New

The system data read by set_sighandlers in the file wcc/wsh.c at line 3644 is potentially exposed by set_sighandlers found in wcc/wsh.c at line 3644.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3663	3663
Object	perror	perror

Code Snippet

File Name	wcc/wsh.c
Method	int set_sighandlers(void)

```
....
3663.             perror("sigaction");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=556
Status	New

The system data read by set_sighandlers in the file wcc/wsh.c at line 3644 is potentially exposed by set_sighandlers found in wcc/wsh.c at line 3644.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3670	3670
Object	perror	perror

Code Snippet

File Name	wcc/wsh.c
Method	int set_sighandlers(void)

```
....
3670.               perror("sigaction");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=557
Status	New

The system data read by set_sighandlers in the file wcc/wsh.c at line 3644 is potentially exposed by set_sighandlers found in wcc/wsh.c at line 3644.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3677	3677
Object	perror	perror

Code Snippet

File Name	wcc/wsh.c
Method	int set_sighandlers(void)

```
....
3677.               perror("sigaction");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=558
Status	New

The system data read by set_sighandlers in the file wcc/wsh.c at line 3644 is potentially exposed by set_sighandlers found in wcc/wsh.c at line 3644.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3684	3684
Object	perror	perror

Code Snippet

File Name	wcc/wsh.c
Method	int set_sighandlers(void)

```
....
3684.                perror("sigaction");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=559
Status	New

The system data read by set_sighandlers in the file wcc/wsh.c at line 3644 is potentially exposed by set_sighandlers found in wcc/wsh.c at line 3644.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3692	3692
Object	perror	perror

Code Snippet	
File Name	wcc/wsh.c
Method	int set_sighandlers(void)

```
....
3692.                perror("sigaction");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=560
Status	New

The system data read by read_elf_sig in the file wcc/wsh.c at line 4621 is potentially exposed by read_elf_sig found in wcc/wsh.c at line 4621.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	4631	4631
Object	perror	perror

Code Snippet	
File Name	wcc/wsh.c
Method	unsigned int read_elf_sig(char *fname, struct stat *sb)

```
....
4631.                perror("open");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=561
Status	New

The system data read by load_home_user_file in the file wcc/wsh.c at line 4661 is potentially exposed by load_home_user_file found in wcc/wsh.c at line 4661.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	4674	4678
Object	getenv	printf

Code Snippet

File Name	wcc/wsh.c
Method	int load_home_user_file(char *fname)

```
....
4674.        snprintf(pathname, 254, "%s/%s", getenv("HOME"), fname);
....
4678.                        printf("WARNING: %s file not found\n",
pathname);
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=562
Status	New

The system data read by load_home_user_file in the file wcc/wsh.c at line 4661 is potentially exposed by load_home_user_file found in wcc/wsh.c at line 4661.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	4674	4691
Object	getenv	printf

Code Snippet

File Name	wcc/wsh.c
Method	int load_home_user_file(char *fname)

```
....
4674.       snprintf(pathname, 254, "%s/%s", getenv("HOME"), fname);
....
4691.             printf("WARNING: %s while running startup script %s
(%s)\n", lua_strerror(err), pathname, lua_tostring(wsh->L, -1));
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=563
Status	New

The system data read by load_home_user_file in the file wcc/wsh.c at line 4661 is potentially exposed by load_home_user_file found in wcc/wsh.c at line 4661.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	4674	4685
Object	getenv	printf

Code Snippet

File Name wcc/wsh.c

Method int load_home_user_file(char *fname)

```
....
4674.       snprintf(pathname, 254, "%s/%s", getenv("HOME"), fname);
....
4685.             printf("  * Running user startup script %s\n",
pathname);
```

Exposure of System Data to Unauthorized Control Sphere\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=564
Status	New

The system data read by fixup_text in the file wcc/wcc.c at line 1656 is potentially exposed by fixup_text found in wcc/wcc.c at line 1656.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1676	1676
Object	errno	printf

Code Snippet

File Name	wcc/wcc.c
Method	int fixup_text(ctx_t * ctx)

```
....
1676.           printf(" ERROR: realloc() %s\n", strerror(errno));
```

Exposure of System Data to Unauthorized Control Sphere\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=565
Status	New

The system data read by write_section in the file wcc/wcc.c at line 2110 is potentially exposed by write_section found in wcc/wcc.c at line 2110.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2120	2120
Object	errno	printf

Code Snippet

File Name	wcc/wcc.c
Method	static int write_section(ctx_t * ctx, msec_t * m)

```
....
2120.       printf("write failed: %u != %lu %s\n", nwrite, m->len,
strerror(errno));
```

Exposure of System Data to Unauthorized Control Sphere\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=566
Status	New

The system data read by open_target in the file wcc/wcc.c at line 2348 is potentially exposed by open_target found in wcc/wcc.c at line 2348.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2373	2373
Object	errno	printf

Code Snippet

File Name	wcc/wcc.c
Method	int open_target(ctx_t * ctx)

```
....
2373.       printf(" ERROR: open(%s) %s\n", newname, strerror(errno));
```

Exposure of System Data to Unauthorized Control Sphere\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=567
Status	New

The system data read by read_section in the file wcc/wcc.c at line 2495 is potentially exposed by read_section found in wcc/wcc.c at line 2495.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2505	2505
Object	errno	printf

Code Snippet
File Name wcc/wcc.c
Method static int read_section(ctx_t * ctx, asection * s)

```
....
2505.       printf("error: open(%s) : %s\n", ctx->binname,
strerror(errno));
```

Exposure of System Data to Unauthorized Control Sphere\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=568
Status	New

The system data read by *ctx_init in the file wcc/wcc.c at line 3684 is potentially exposed by *ctx_init found in wcc/wcc.c at line 3684.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3689	3692
Object	errno	printf

Code Snippet
File Name wcc/wcc.c
Method ctx_t *ctx_init(void)

```
....
3689.     errno = 0;
....
3692.       printf("error: calloc(): %s\n", strerror(errno));
```

Exposure of System Data to Unauthorized Control Sphere\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=569
Status	New

The system data read by *ctx_init in the file wcc/wcc.c at line 3684 is potentially exposed by *ctx_init found in wcc/wcc.c at line 3684.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3692	3692
Object	errno	printf

Code Snippet

File Name	wcc/wcc.c
Method	ctx_t *ctx_init(void)

```
....
3692.       printf("error: calloc(): %s\n", strerror(errno));
```

Exposure of System Data to Unauthorized Control Sphere\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=570
Status	New

The system data read by ctx_getopt in the file wcc/wcc.c at line 3756 is potentially exposed by ctx_getopt found in wcc/wcc.c at line 3756.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3906	3905
Object	errno	printf

Code Snippet

File Name	wcc/wcc.c
Method	int ctx_getopt(ctx_t * ctx, int argc, char **argv)

```
....
3906.               strerror(errno));
....
3905.       printf("error: Could not open file %s : %s\n", argv[count +
1],
```

Exposure of System Data to Unauthorized Control Sphere\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=571
Status	New

The system data read by disable_aslr in the file wcc/wsh.c at line 461 is potentially exposed by disable_aslr found in wcc/wsh.c at line 461.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	468	468
Object	errno	fprintf

Code Snippet
File Name   wcc/wsh.c
Method   int disable_aslr(void)

```
....
468.                fprintf(stderr, "!! ERROR : open(%s, O_RDWR) %s\n",
PROC_ASLR_PATH, strerror(errno));
```

Exposure of System Data to Unauthorized Control Sphere\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=572
Status	New

The system data read by enable_aslr in the file wcc/wsh.c at line 479 is potentially exposed by enable_aslr found in wcc/wsh.c at line 479.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	487	487
Object	errno	fprintf

Code Snippet
File Name   wcc/wsh.c
Method   int enable_aslr(void)

```
....
487.                fprintf(stderr, "!! ERROR : open(%s,O_RDWR) %s\n",
PROC_ASLR_PATH, strerror(errno));
```

Exposure of System Data to Unauthorized Control Sphere\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=573
Status	New

The system data read by add_symbol in the file wcc/wsh.c at line 681 is potentially exposed by add_symbol found in wcc/wsh.c at line 681.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	687	687
Object	errno	fprintf

Code Snippet

File Name	wcc/wsh.c
Method	int add_symbol(char *symbol, char *libname, char *htype, char *hbind, unsigned long value, unsigned int size, unsigned long int addr)

```
....
687.         if(!s){ fprintf(stderr, " !! Error: calloc() = %s\n",
strerror(errno)); return -1; }
```

Exposure of System Data to Unauthorized Control Sphere\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=574
Status	New

The system data read by section_add in the file wcc/wsh.c at line 713 is potentially exposed by section_add found in wcc/wsh.c at line 713.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	718	718
Object	errno	fprintf

Code Snippet

File Name	wcc/wsh.c

Method void section_add(unsigned long int addr, unsigned long int size, char *libname, char *name, char *perms, int flags)

```
....
718.         if(!s){ fprintf(stderr, " !! Error: calloc() = %s\n",
strerror(errno)); return; }
```

Exposure of System Data to Unauthorized Control Sphere\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=575
Status	New

The system data read by segment_add in the file wcc/wsh.c at line 732 is potentially exposed by segment_add found in wcc/wsh.c at line 732.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	737	737
Object	errno	fprintf

Code Snippet

File Name wcc/wsh.c

Method void segment_add(unsigned long int addr, unsigned long int size, char *perms, char *fname, char *ptype, int flags)

```
....
737.         if(!s){ fprintf(stderr, " !! Error: calloc() = %s\n",
strerror(errno)); return; }
```

Exposure of System Data to Unauthorized Control Sphere\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=576
Status	New

The system data read by libcall in the file wcc/wsh.c at line 2059 is potentially exposed by libcall found in wcc/wsh.c at line 2059.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	2092	2234
Object	errno	fprintf

Code Snippet

File Name	wcc/wsh.c
Method	static int libcall(lua_State * L)

```
....
2092.        errno = 0;
....
2234.              fprintf(stderr, "ERROR: %s (%u)\n",
strerror(callerrno), callerrno);
```

Exposure of System Data to Unauthorized Control Sphere\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=577
Status	New

The system data read by libcall in the file wcc/wsh.c at line 2059 is potentially exposed by libcall found in wcc/wsh.c at line 2059.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	2204	2234
Object	errno	fprintf

Code Snippet

File Name	wcc/wsh.c
Method	static int libcall(lua_State * L)

```
....
2204.        callerrno = errno;
....
2234.              fprintf(stderr, "ERROR: %s (%u)\n",
strerror(callerrno), callerrno);
```

Exposure of System Data to Unauthorized Control Sphere\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=578
Status	New

The system data read by print_procmap in the file wcc/wsh.c at line 2848 is potentially exposed by print_procmap found in wcc/wsh.c at line 2848.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	2859	2859
Object	errno	printf

Code Snippet

File Name	wcc/wsh.c
Method	int print_procmap(unsigned int pid)

```
....
2859.        if(fd < 0){ printf(" !! ERROR: open %s : %s\n", path,
strerror(errno)); return -1; }
```

Exposure of System Data to Unauthorized Control Sphere\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=579
Status	New

The system data read by execlib in the file wcc/wsh.c at line 2876 is potentially exposed by execlib found in wcc/wsh.c at line 2876.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	2891	2891
Object	errno	fprintf

Code Snippet

File Name	wcc/wsh.c
Method	int execlib(lua_State * L)

```
....
2891.              fprintf(stderr, "ERROR: fork() : %s\n",
strerror(errno));
```

Exposure of System Data to Unauthorized Control Sphere\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=580
Status	New

The system data read by execlib in the file wcc/wsh.c at line 2876 is potentially exposed by execlib found in wcc/wsh.c at line 2876.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	2904	2904
Object	errno	fprintf

Code Snippet

File Name	wcc/wsh.c
Method	int execlib(lua_State * L)

```
....
2904.                                fprintf(stderr, "ERROR: ptrace() :
%s\n", strerror(errno));
```

Exposure of System Data to Unauthorized Control Sphere\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=581
Status	New

The system data read by affinity in the file wcc/wsh.c at line 3035 is potentially exposed by affinity found in wcc/wsh.c at line 3035.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3043	3043
Object	errno	fprintf

Code Snippet

File Name	wcc/wsh.c
Method	void affinity(int procnum)

```
....
3043.             fprintf(stderr, " !! ERROR: sched_setaffinity(%u):
%s\n", procnum, strerror(errno));
```

Exposure of System Data to Unauthorized Control Sphere\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=582
Status	New

The system data read by btr_enable in the file wcc/wsh.c at line 3050 is potentially exposed by btr_enable found in wcc/wsh.c at line 3050.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3059	3059
Object	errno	fprintf

Code Snippet

File Name	wcc/wsh.c
Method	void btr_enable(int procnum)

```
....
3059.        if(fd <= 0){ fprintf(stderr, "ERROR: open(%s): %s\n",
cpupath,strerror(errno)); return; }
```

Exposure of System Data to Unauthorized Control Sphere\Path 45:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=583
Status	New

The system data read by btr_enable in the file wcc/wsh.c at line 3050 is potentially exposed by btr_enable found in wcc/wsh.c at line 3050.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3061	3061
Object	errno	fprintf

Code Snippet

File Name	wcc/wsh.c
Method	void btr_enable(int procnum)

```
....
3061.        if(ret != 0x00){ fprintf(stderr, "ERROR: lseek(): %s\n",
strerror(errno)); return; }
```

Exposure of System Data to Unauthorized Control Sphere\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=584
Status	New

The system data read by btr_enable in the file wcc/wsh.c at line 3050 is potentially exposed by btr_enable found in wcc/wsh.c at line 3050.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3063	3063
Object	errno	fprintf

Code Snippet

File Name	wcc/wsh.c

Method	void btr_enable(int procnum)

```
....
3063.        if(ret != sizeof(data)){ fprintf(stderr, "ERROR: write():
%s\n", strerror(errno)); return; }
```

Exposure of System Data to Unauthorized Control Sphere\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=585
Status	New

The system data read by btr_enable in the file wcc/wsh.c at line 3050 is potentially exposed by btr_enable found in wcc/wsh.c at line 3050.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3065	3065
Object	errno	fprintf

Code Snippet	
File Name	wcc/wsh.c
Method	void btr_enable(int procnum)

```
....
3065.        if(ret != 0){ fprintf(stderr, "ERROR: close(): %s\n",
strerror(errno)); return; }
```

Exposure of System Data to Unauthorized Control Sphere\Path 48:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=586
Status	New

The system data read by btr_disable in the file wcc/wsh.c at line 3071 is potentially exposed by btr_disable found in wcc/wsh.c at line 3071.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3080	3080
Object	errno	fprintf

Code Snippet	
File Name	wcc/wsh.c
Method	void btr_disable(int procnum)

```
....
3080.        if(fd <= 0){ fprintf(stderr, "ERROR: open(%s): %s\n",
cpupath,strerror(errno)); return; }
```

Exposure of System Data to Unauthorized Control Sphere\Path 49:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=587
Status	New

The system data read by btr_disable in the file wcc/wsh.c at line 3071 is potentially exposed by btr_disable found in wcc/wsh.c at line 3071.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3082	3082
Object	errno	fprintf

Code Snippet
File Name wcc/wsh.c
Method void btr_disable(int procnum)

```
....
3082.        if(ret != 0x00){ fprintf(stderr, "ERROR: lseek(): %s\n",
strerror(errno)); return; }
```

Exposure of System Data to Unauthorized Control Sphere\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=588
Status	New

The system data read by btr_disable in the file wcc/wsh.c at line 3071 is potentially exposed by btr_disable found in wcc/wsh.c at line 3071.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3084	3084
Object	errno	fprintf

Code Snippet
File Name wcc/wsh.c
Method void btr_disable(int procnum)

```
....
3084.        if(ret != sizeof(data)){ fprintf(stderr, "ERROR: write():
%s\n", strerror(errno)); return; }
```

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=1050048&projectid=50038&pathid=76
Status	New

	Source	Destination
File	wcc/lgc.c	wcc/lgc.c
Line	493	493
Object	sizeof	sizeof

Code Snippet
File Name   wcc/lgc.c
Method   static lu_mem traversetable (global_State *g, Table *h) {

```
....
493.                                  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=1050048&projectid=50038&pathid=77
Status	New

	Source	Destination
File	wcc/lgc.c	wcc/lgc.c
Line	1052	1052
Object	sizeof	sizeof

Code Snippet
File Name   wcc/lgc.c
Method   static lu_mem singlestep (lua_State *L) {

```
....
1052.        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=1050048&projectid=50038&pathid=78
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	438	438
Object	sizeof	sizeof

Code Snippet
File Name wcc/wcc.c
Method void add_symaddr(ctx_t * ctx, const char *name, int addr, char symclass)

```
....
438.     for (i = 0; i < sizeof(blnames) / sizeof(char *); i++) {
```

Use of Sizeof On a Pointer Type\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=79
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	758	758
Object	sizeof	sizeof

Code Snippet
File Name wcc/wcc.c
Method unsigned int secindex_from_name_after_strip(ctx_t * ctx, const char *name)

```
....
758.       for (j = 0; j < sizeof(allowed_sections) / sizeof(char *);
j++) {
```

Use of Sizeof On a Pointer Type\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=80
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	777	777
Object	sizeof	sizeof

Code Snippet
File Name wcc/wcc.c
Method char *sec_name_from_index_after_strip(ctx_t * ctx, unsigned int index)

```
....
777.        for (j = 0; j < sizeof(allowed_sections) / sizeof(char *);
j++) {
```

Use of Sizeof On a Pointer Type\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=81
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1632	1632
Object	sizeof	sizeof

Code Snippet
File Name wcc/wcc.c
Method int fixup_strtab_and_symtab(ctx_t * ctx)

```
....
1632.       for (i = 0; i < sizeof(blnames) / sizeof(char *); i++) {
```

Use of Sizeof On a Pointer Type\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=82
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2727	2727
Object	sizeof	sizeof

Code Snippet

File Name	wcc/wcc.c
Method	int save_global_import(ctx_t * ctx, char *sname, msec_t * sec, Elf_Rela * r, unsigned int sindex)

```
....
2727.       gimports = calloc(1, sizeof(gimport_t *));
```

Use of Sizeof On a Pointer Type\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=83
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2729	2729
Object	sizeof	sizeof

Code Snippet

File Name	wcc/wcc.c
Method	int save_global_import(ctx_t * ctx, char *sname, msec_t * sec, Elf_Rela * r, unsigned int sindex)

```
....
2729.       gimports = realloc(gimports, sizeof(gimport_t *) * 
(gimportslen + 1));
```

Use of Sizeof On a Pointer Type\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=84
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2799	2799
Object	sizeof	sizeof

Code Snippet

File Name	wcc/wcc.c
Method	int save_reloc(ctx_t * ctx, Elf_Rela * r, unsigned int sindex, int has_addend)

```
....
2799.     for (i = 0; i < sizeof(blnames) / sizeof(char *); i++) {
```

Use of Sizeof On a Pointer Type\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=85
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3482	3482
Object	sizeof	sizeof

Code Snippet

File Name	wcc/wcc.c
Method	int strip_binary_reloc(ctx_t * ctx)

```
....
3482.       for (i = 0; i < sizeof(allowed_sections) / sizeof(char *);
i++) {
```

Use of Sizeof On a Pointer Type\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=86
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	397	397
Object	sizeof	sizeof

Code Snippet

File Name	wcc/wsh.c
Method	void completion(const char *buf, linenoiseCompletions * lc)

```
....
397.             for (i = 0; i < sizeof(default_options) / sizeof(char
*); i++) {
```

Use of Sizeof On a Pointer Type\Path 12:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=87
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	426	426
Object	sizeof	sizeof

Code Snippet
File Name wcc/wsh.c
Method void completion(const char *buf, linenoiseCompletions * lc)

```
....
426.                for (i = 0; i < sizeof(default_options) / 
sizeof(char *); i++) {
```

Use of Sizeof On a Pointer Type\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=88
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	433	433
Object	sizeof	sizeof

Code Snippet
File Name wcc/wsh.c
Method void completion(const char *buf, linenoiseCompletions * lc)

```
....
433.                for (i = 0; i < sizeof(lua_default_functions) / 
sizeof(char *); i++) {
```

Use of Sizeof On a Pointer Type\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=89
Status	New

	Source	Destination

File	wcc/wsh.c	wcc/wsh.c
Line	1585	1585
Object	sizeof	sizeof

Code Snippet
File Name wcc/wsh.c
Method int alloccharbuf(lua_State * L)

```
....
1585.        ptr = calloc(n * sizeof(char *), 1);
```

Use of Sizeof On a Pointer Type\Path 15:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=90
Status New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	2583	2583
Object	sizeof	sizeof

Code Snippet
File Name wcc/wsh.c
Method void scan_syms(char *dynstr, Elf_Sym * sym, unsigned long int sz, char *libname)

```
....
2583.                   for(j=0; j < 
sizeof(lua_blacklist)/sizeof(char*);j++){
```

Use of Sizeof On a Pointer Type\Path 16:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=91
Status New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	2590	2590
Object	sizeof	sizeof

Code Snippet

File Name	wcc/wsh.c
Method	void scan_syms(char *dynstr, Elf_Sym * sym, unsigned long int sz, char *libname)

```
....
2590.                           for(j=0; j <
sizeof(lua_default_functions)/sizeof(char*);j++){
```

Use of Sizeof On a Pointer Type\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=92
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	4800	4800
Object	sizeof	sizeof

Code Snippet

File Name	wcc/wsh.c
Method	int add_script_arguments(int argc, char **argv, unsigned int i)

```
....
4800.       wsh->script_args = calloc(sizeof(void *), argc);
```

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=1050048&projectid=50038&pathid=609
Status	New

The learn_proto method in wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	1836	1836
Object	fopen	fopen

Code Snippet

File Name	wcc/wsh.c
Method	int learn_proto(unsigned long*arg, unsigned long int faultaddr, int reason)

```
....
1836.                wsh->learnfile = fopen( wsh->learnlog ? wsh->learnlog
: DEFAULT_LEARN_FILE ,"a+");
```

TOCTOU\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=610
Status	New

The prototypes method in wcc/wsh.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	wcc/wsh.c	wcc/wsh.c
Line	1866	1866
Object	fopen	fopen

Code Snippet

File Name	wcc/wsh.c
Method	int prototypes(lua_State * L)

```
....
1866.                wsh->learnfile = fopen( wsh->learnlog ? wsh->learnlog
: DEFAULT_LEARN_FILE ,"a+");
```

TOCTOU\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=611
Status	New

The rd_phdrs method in wcc/wcc.c file utilizes open that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1145	1145
Object	open	open

Code Snippet

File Name	wcc/wcc.c

Method	static unsigned int rd_phdrs(ctx_t * ctx)

```
....
1145.     fdin = open(ctx->binname, O_RDONLY);
```

TOCTOU\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=612
Status	New

The open_target method in wcc/wcc.c file utilizes open that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2371	2371
Object	open	open

Code Snippet

File Name	wcc/wcc.c
Method	int open_target(ctx_t * ctx)

```
....
2371.     fd = open(newname, O_RDWR | O_CREAT | O_TRUNC, 0666);
```

TOCTOU\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=613
Status	New

The open_target method in wcc/wcc.c file utilizes open that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2386	2386
Object	open	open

Code Snippet

File Name	wcc/wcc.c
Method	int open_target(ctx_t * ctx)

```
....
2386.       int fdin = open(ctx->binname, O_RDONLY);
```

TOCTOU\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=614
Status	New

The read_section method in wcc/wcc.c file utilizes open that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2503	2503
Object	open	open

Code Snippet
File Name wcc/wcc.c
Method static int read_section(ctx_t * ctx, asection * s)

```
....
2503.     fd = open(ctx->binname, O_RDONLY);
```

TOCTOU\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=615
Status	New

The rd_symtab method in wcc/wcc.c file utilizes open that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3370	3370
Object	open	open

Code Snippet
File Name wcc/wcc.c
Method int rd_symtab(ctx_t * ctx)

```
....
3370.     fd = open(ctx->binname, O_RDONLY);
```

TOCTOU\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=616
Status	New

The disable_aslr method in wcc/wsh.c file utilizes open that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	466	466
Object	open	open

Code Snippet

File Name	wcc/wsh.c
Method	int disable_aslr(void)

```
....
466.         fd = open(PROC_ASLR_PATH, O_RDWR);
```

TOCTOU\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=617
Status	New

The enable_aslr method in wcc/wsh.c file utilizes open that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	485	485
Object	open	open

Code Snippet

File Name	wcc/wsh.c
Method	int enable_aslr(void)

```
....
485.         fd = open(PROC_ASLR_PATH, O_RDWR);
```

TOCTOU\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=618
Status	New

The scan_sections method in wcc/wsh.c file utilizes open that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	791	791
Object	open	open

Code Snippet

File Name	wcc/wsh.c
Method	int scan_sections(char *fname, unsigned long int baseaddr)

```
....
791.         fd = open(fname, O_RDONLY);
```

TOCTOU\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=619
Status	New

The print_procmap method in wcc/wsh.c file utilizes open that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	2858	2858
Object	open	open

Code Snippet

File Name	wcc/wsh.c
Method	int print_procmap(unsigned int pid)

```
....
2858.        fd = open(path, O_RDONLY);
```

TOCTOU\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=620
Status	New

The btr_enable method in wcc/wsh.c file utilizes open that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3058	3058
Object	open	open

Code Snippet

File Name	wcc/wsh.c
Method	void btr_enable(int procnum)

```
....
3058.        fd = open(cpupath, O_WRONLY);
```

TOCTOU\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=621
Status	New

The btr_disable method in wcc/wsh.c file utilizes open that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	3079	3079
Object	open	open

Code Snippet

File Name	wcc/wsh.c
Method	void btr_disable(int procnum)

```
....
3079.        fd = open(cpupath, O_WRONLY);
```

TOCTOU\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=622
Status	New

The read_elf_sig method in wcc/wsh.c file utilizes open that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	4629	4629
Object	open	open

Code Snippet	
File Name	wcc/wsh.c
Method	unsigned int read_elf_sig(char *fname, struct stat *sb)

```
....
4629.        fd = open(fname, O_RDONLY);
```

TOCTOU\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=623
Status	New

The mk_lib method in wcc/wsh.c file utilizes open that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	4854	4854
Object	open	open

Code Snippet	
File Name	wcc/wsh.c
Method	int mk_lib(char *name)

```
....
4854.     fd = open(name, O_RDWR);
```

TOCTOU\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=624
Status	New

The attempt_to_patch method in wcc/wsh.c file utilizes open that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	4950	4950
Object	open	open

Code Snippet

File Name	wcc/wsh.c
Method	int attempt_to_patch(char *libname)

```
....
4950.         fdin = open(libname, O_RDONLY, 0700);
```

TOCTOU\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=625
Status	New

The attempt_to_patch method in wcc/wsh.c file utilizes open that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	4977	4977
Object	open	open

Code Snippet

File Name	wcc/wsh.c
Method	int attempt_to_patch(char *libname)

```
....
4977.         fdout = open(outlib, O_RDWR|O_CREAT|O_TRUNC, 0700);
```

Heuristic 2nd Order Buffer Overflow read

Query Path:
CPP\Cx\CPP Heuristic\Heuristic 2nd Order Buffer Overflow read Version:0

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

Heuristic 2nd Order Buffer Overflow read\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=224
Status	New

The size of the buffer used by read_section in BinaryExpr, at line 2495 of wcc/wcc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that read_section passes to BinaryExpr, at line 2495 of wcc/wcc.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2541	2541
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name   wcc/wcc.c
Method   static int read_section(ctx_t * ctx, asection * s)

```
....
2541.         nread = read(fd, buf + n, s->size - n);
```

Heuristic 2nd Order Buffer Overflow read\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=225
Status	New

The size of the buffer used by read_section in BinaryExpr, at line 2495 of wcc/wcc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that read_section passes to buf, at line 2495 of wcc/wcc.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c

Line	2538	2541
Object	buf	BinaryExpr

Code Snippet
File Name wcc/wcc.c
Method static int read_section(ctx_t * ctx, asection * s)

```
....
2538.       nread = read(fd, buf, s->size);
....
2541.         nread = read(fd, buf + n, s->size - n);
```

Heuristic 2nd Order Buffer Overflow read\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=226
Status	New

The size of the buffer used by read_section in size, at line 2495 of wcc/wcc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that read_section passes to BinaryExpr, at line 2495 of wcc/wcc.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2541	2541
Object	BinaryExpr	size

Code Snippet
File Name wcc/wcc.c
Method static int read_section(ctx_t * ctx, asection * s)

```
....
2541.         nread = read(fd, buf + n, s->size - n);
```

Heuristic 2nd Order Buffer Overflow read\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=227
Status	New

The size of the buffer used by read_section in size, at line 2495 of wcc/wcc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that read_section passes to buf, at line 2495 of wcc/wcc.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c

Line	2538	2541
Object	buf	size

Code Snippet
File Name wcc/wcc.c
Method static int read_section(ctx_t * ctx, asection * s)

```
....
2538.       nread = read(fd, buf, s->size);
....
2541.         nread = read(fd, buf + n, s->size - n);
```

Heuristic 2nd Order Buffer Overflow read\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=228
Status	New

The size of the buffer used by read_section in n, at line 2495 of wcc/wcc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that read_section passes to BinaryExpr, at line 2495 of wcc/wcc.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	2541	2541
Object	BinaryExpr	n

Code Snippet
File Name wcc/wcc.c
Method static int read_section(ctx_t * ctx, asection * s)

```
....
2541.         nread = read(fd, buf + n, s->size - n);
```

Heuristic 2nd Order Buffer Overflow read\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=229
Status	New

The size of the buffer used by read_section in n, at line 2495 of wcc/wcc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that read_section passes to buf, at line 2495 of wcc/wcc.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c

Line	2538	2541
Object	buf	n

Code Snippet
File Name wcc/wcc.c
Method static int read_section(ctx_t * ctx, asection * s)

```
....
2538.      nread = read(fd, buf, s->size);
....
2541.        nread = read(fd, buf + n, s->size - n);
```

Unchecked Array Index

Query Path:
CPP\Cx\CPP Low Visibility\Unchecked Array Index Version:1

Categories

NIST SP 800-53: SI-10 Information Input Validation (P1)

Description

Unchecked Array Index\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=235
Status	New

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c
Line	854	854
Object	n	n

Code Snippet
File Name wcc/lstrlib.c
Method static lua_Number adddigit (char *buff, int n, lua_Number x) {

```
....
854.    buff[n] = (d < 10 ? d + '0' : d - 10 + 'a');   /* add to buffer
*/
```

Unchecked Array Index\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=236
Status	New

	Source	Destination

File	wcc/wcc.c	wcc/wcc.c
Line	2058	2058
Object	EI_CLASS	EI_CLASS

Code Snippet
File Name wcc/wcc.c
Method static int mk_ehdr(ctx_t * ctx)

```
....
2058.     e->e_ident[EI_CLASS] = ELFCLASS; // 64 or 32 bits
```

Unchecked Array Index\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=237
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	1601	1601
Object	pos	pos

Code Snippet
File Name wcc/wsh.c
Method int setcharbuf(lua_State * L)

```
....
1601.        buff[pos] = val;
```

Inconsistent Implementations

Query Path:
CPP\Cx\CPP Low Visibility\Inconsistent Implementations Version:0

Description

Inconsistent Implementations\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=1
Status	New

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3793	3793
Object	getopt_long	getopt_long

Code Snippet

File Name	wcc/wcc.c
Method	int ctx_getopt(ctx_t * ctx, int argc, char **argv)

```
....
3793.    while ((c = getopt_long(argc, argv, short_opt, long_opt, NULL))
!= -1) {
```

Inconsistent Implementations\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=2
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	5095	5095
Object	getopt_long	getopt_long

Code Snippet

File Name	wcc/wsh.c
Method	int wsh_getopt(int argc, char **argv)

```
....
5095.        while ((c = getopt_long(argc, argv, short_opt, long_opt,
NULL)) != -1) {
```

Arithmenic Operation On Boolean

Query Path:
CPP\Cx\CPP Low Visibility\Arithmenic Operation On Boolean Version:1

Categories

FISMA 2014: Audit And Accountability
NIST SP 800-53: SC-5 Denial of Service Protection (P1)

Description

Arithmenic Operation On Boolean\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=233
Status	New

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c
Line	126	126

Object	BinaryExpr	BinaryExpr

Code Snippet
File Name wcc/lstrlib.c
Method static int str_rep (lua_State *L) {

```
....
126.     else if (l + lsep < l || l + lsep > MAXSIZE / n)   /* may
overflow? */
```

Arithmenic Operation On Boolean\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=234
Status	New

	Source	Destination
File	wcc/lstrlib.c	wcc/lstrlib.c
Line	1395	1395
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name wcc/lstrlib.c
Method static int str_packsize (lua_State *L) {

```
....
1395.       luaL_argcheck(L, totalsize <= MAXSIZE - size, 1,
```

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=1050048&projectid=50038&pathid=93
Status	New

The buffer allocated by <= in wcc/wcc.c at line 2990 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	3131	3131
Object	<=	<=

Code Snippet
File Name wcc/wcc.c
Method static int create_text_data_reloc(ctx_t * ctx, cs_insn * ins, msec_t * m,

```
....
3131.         for (wheretowrite = 0; wheretowrite <= ins->size;
wheretowrite++) {
```

Potential Precision Problem

Query Path:
CPP\Cx\CPP Buffer Overflow\Potential Precision Problem Version:0

Categories

NIST SP 800-53: SI-10 Information Input Validation (P1)
OWASP Top 10 2017: A1-Injection

Description
Potential Precision Problem\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050048&projectid=50038&pathid=230
Status	New

The size of the buffer used by fixup_strtab_and_symtab in "old_%s", at line 1603 of wcc/wcc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that fixup_strtab_and_symtab passes to "old_%s", at line 1603 of wcc/wcc.c, to overwrite the target buffer.

	Source	Destination
File	wcc/wcc.c	wcc/wcc.c
Line	1643	1643
Object	"old_%s"	"old_%s"

Code Snippet
File Name wcc/wcc.c
Method int fixup_strtab_and_symtab(ctx_t * ctx)

```
....
1643.           sprintf(globalstrtab + globalstrtablen, "old_%s", sname);
```

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=1050048&projectid=50038&pathid=538
Status	New

	Source	Destination
File	wcc/wsh.c	wcc/wsh.c
Line	4963	4963
Object	mkdir	mkdir

Code Snippet

File Name	wcc/wsh.c
Method	int attempt_to_patch(char *libname)

```
....
4963.        if(mkdir(tmp_dirname, 0700)){
```

Buffer Overflow AddressOfLocalVarReturned

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
//..
```

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);
```

Integer Overflow

Risk

What might happen

Assigning large data types into smaller data types, without proper checks and explicit casting, will lead to undefined behavior and unintentional effects, such as data corruption (e.g. value wraparound, wherein maximum values become minimum values); system crashes; infinite loops; logic errors, such as bypassing of security mechanisms; or even buffer overflows leading to arbitrary code execution.

Cause

How does it happen

This flaw can occur when implicitly casting numerical data types of a larger size, into a variable with a data type of a smaller size. This forces the program to discard some bits of information from the number. Depending on how the numerical data types are stored in memory, this is often the bits with the highest value, causing substantial corruption of the stored number. Alternatively, the sign bit of a signed integer could be lost, completely reversing the intention of the number.

General Recommendations

How to avoid it

- Avoid casting larger data types to smaller types.
- Prefer promoting the target variable to a large enough data type.
- If downcasting is necessary, always check that values are valid and in range of the target type, before casting

Source Code Examples

CPP

Unsafe Downsize Casting

```
int unsafe_addition(short op1, int op2) {

    // op2 gets forced from int into a short
    short total = op1 + op2;

    return total;
}
```

Safer Use of Proper Data Types

```
int safe_addition(short op1, int op2) {

    // total variable is of type int, the largest type that is needed
    int total = 0;

    // check if total will overflow available integer size
    if (INT_MAX - abs(op2) > op1)
    {
        total = op1 + op2;
    }
    else
    {
        // instead of overflow, saturate (but this is not always a good thing)
        total = INT_MAX
    }

    return total;
}
```

Dangerous Functions

Risk

What might happen

Use of dangerous functions may expose varying risks associated with each particular function, with potential impact of improper usage of these functions varying significantly. The presence of such functions indicates a flaw in code maintenance policies and adherence to secure coding practices, in a way that has allowed introducing known dangerous code into the application.

Cause

How does it happen

A dangerous function has been identified within the code. Functions are often deemed dangerous to use for numerous reasons, as there are different sets of vulnerabilities associated with usage of such functions. For example, some string copy and concatenation functions are vulnerable to Buffer Overflow, Memory Disclosure, Denial of Service and more. Use of these functions is not recommended.

General Recommendations

How to avoid it

- Deploy a secure and recommended alternative to any functions that were identified as dangerous.
    - If no secure alternative is found, conduct further researching and testing to identify whether current usage successfully sanitizes and verifies values, and thus successfully avoids the use-cases for whom the function is indeed dangerous
- Conduct a periodical review of methods that are in use, to ensure that all external libraries and built-in functions are up-to-date and whose use has not been excluded from best secure coding practices.

Source Code Examples

CPP

Buffer Overflow in gets()

```
int main()

{

      char buf[10];

      printf("Please enter your name: ");
      gets(buf); // veryveryverylongname
      if (buf == ACCEPTED_NAME)
      {
             // Do something
      }
      return 0;
}
```

Safe reading from user

```
int main()
{
       char buf[10];

       printf("Please enter your name: ");
       fgets(buf, sizeof(buf), stdin); //setting the amount of bytes to read
       if (buf == ACCEPTED_NAME)
       {
              //Do something
       }
       return 0;
}
```

Unsafe function for string copy

```
int main(int argc, char* argv[])
{
       char buf[10];
       strcpy(buf, argv[1]); // overflow occurs when len(argv[1]) > 10 bytes

       return 0;
}
```

Safe string copy

```
int main(int argc, char* argv[])
{
       char buf[10];
       strncpy(buf, argv[1], sizeof(buf));
       buf[9]= '\0'; //strncpy doesn't NULL terminates

       return 0;
}
```

Unsafe format string

```
int main(int argc, char* argv[])
{
       printf(argv[1]); // If argv[1] contains a format token, such as %s,%x or %d, will cause
an access violation
       return 0;
}
```

Safe format string

```
int main(int argc, char* argv[])
{
      printf("%s", argv[1]); // Second parameter is not a formattable string

      return 0;
}
```

Double Free

Weakness ID: 415 *(Weakness Variant)* **Status:** Draft

Description

Description Summary

The product calls free() twice on the same memory address, potentially leading to modification of unexpected memory locations.

Extended Description

When a program calls free() twice with the same argument, the program's memory management data structures become corrupted. This corruption can cause the program to crash or, in some circumstances, cause two later calls to malloc() to return the same pointer. If malloc() returns the same value twice and the program later gives the attacker control over the data that is written into this doubly-allocated memory, the program becomes vulnerable to a buffer overflow attack.

Alternate Terms

Double-free

Time of Introduction

- Architecture and Design
- Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

Scope	Effect
Access Control	Doubly freeing memory may result in a write-what-where condition, allowing an attacker to execute arbitrary code.

Likelihood of Exploit

Low to Medium

Demonstrative Examples

Example 1

The following code shows a simple example of a double free vulnerability.

(Bad Code)

Example Language: **C**

```
char* ptr = (char*)malloc (SIZE);
...
if (abrt) {
free(ptr);
}
...
free(ptr);
```

Double free vulnerabilities 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

Although some double free vulnerabilities are not much more complicated than the previous example, most are spread out across hundreds of lines of code or even different files. Programmers seem particularly susceptible to freeing global variables more than once.

Example 2

While contrived, this code should be exploitable on Linux distributions which do not ship with heap-chunk check summing turned on.

(Bad Code)

Example Language: **C**

```
#include <stdio.h>
#include <unistd.h>
#define BUFSIZE1 512
#define BUFSIZE2 ((BUFSIZE1/2) - 8)

int main(int argc, char **argv) {
char *buf1R1;
char *buf2R1;
char *buf1R2;
buf1R1 = (char *) malloc(BUFSIZE2);
buf2R1 = (char *) malloc(BUFSIZE2);
free(buf1R1);
free(buf2R1);
buf1R2 = (char *) malloc(BUFSIZE1);
strncpy(buf1R2, argv[1], BUFSIZE1-1);
free(buf2R1);
free(buf1R2);
}
```

Observed Examples

Reference	Description
CVE-2004-0642	Double free resultant from certain error conditions.
CVE-2004-0772	Double free resultant from certain error conditions.
CVE-2005-1689	Double free resultant from certain error conditions.
CVE-2003-0545	Double free from invalid ASN.1 encoding.
CVE-2003-1048	Double free from malformed GIF.
CVE-2005-0891	Double free from malformed GIF.
CVE-2002-0059	Double free from malformed compressed data.

Potential Mitigations

Phase: Architecture and Design

Choose a language that provides automatic memory management.

Phase: Implementation

Ensure that each allocation is freed only once. After freeing a chunk, set the pointer to NULL to ensure the pointer cannot be freed again. In complicated error conditions, be sure that clean-up routines respect the state of allocation properly. If the language is object oriented, ensure that object destructors delete each chunk of memory only once.

Phase: Implementation

Use a static analysis tool to find double free instances.

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	Weakness Base	666	Operation on Resource in Wrong Phase of	**Research Concepts (primary)1000**

			Lifetime	
ChildOf	Weakness Class	675	Duplicate Operations on Resource	Research Concepts1000
ChildOf	Category	742	CERT C Secure Coding Section 08 - Memory Management (MEM)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
PeerOf	Weakness Base	123	Write-what-where Condition	Research Concepts1000
PeerOf	Weakness Base	416	Use After Free	Development Concepts699 Research Concepts1000
MemberOf	View	630	Weaknesses Examined by SAMATE	**Weaknesses Examined by SAMATE (primary)630**
PeerOf	Weakness Base	364	Signal Handler Race Condition	Research Concepts1000

Relationship Notes

This is usually resultant from another weakness, such as an unhandled error or race condition between threads. It could also be primary to weaknesses such as buffer overflows.

Affected Resources

- Memory

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
PLOVER			DFREE - Double-Free Vulnerability
7 Pernicious Kingdoms			Double Free
CLASP			Doubly freeing memory
CERT C Secure Coding	MEM00-C		Allocate and free memory in the same module, at the same level of abstraction
CERT C Secure Coding	MEM01-C		Store a new value in pointers immediately after free()
CERT C Secure Coding	MEM31-C		Free dynamically allocated memory exactly once

White Box Definitions

A weakness where code path has:

1. start statement that relinquishes a dynamically allocated memory resource
2. end statement that relinquishes the dynamically allocated memory resource

Maintenance Notes

It could be argued that Double Free would be most appropriately located as a child of "Use after Free", but "Use" and "Release" are considered to be distinct operations within vulnerability theory, therefore this is more accurately "Release of a Resource after Expiration or Release", which doesn't exist yet.

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 Potential Mitigations, 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, Description, Maintenance Notes, Relationships, Other Notes, Relationship Notes, Taxonomy Mappings		
2008-11-24	CWE Content Team	MITRE	Internal

	updated Relationships, Taxonomy Mappings		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Other Notes		

BACK TO TOP

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

Stored Buffer Overflow boundcpy

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));
    }
}
```

Use of Function with Inconsistent Implementations

Weakness ID: 474 *(Weakness Base)* **Status:** Draft

Description

Description Summary

The code uses a function that has inconsistent implementations across operating systems and versions, which might cause security-relevant portability problems.

Time of Introduction

- Architecture and Design
- Implementation

Applicable Platforms

Languages

C: *(Often)*

PHP: *(Often)*

All

Potential Mitigations

Do not accept inconsistent behavior from the API specifications when the deviant behavior increase the risk level.

Other Notes

The behavior of functions in this category varies by operating system, and at times, even by operating system version. Implementation differences can include:

- Slight differences in the way parameters are interpreted leading to inconsistent results.

- Some implementations of the function carry significant security risks.

- The function might not be defined on all platforms.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	398	Indicator of Poor Code Quality	**Development Concepts (primary)699** **Seven Pernicious Kingdoms (primary)700** **Research Concepts (primary)1000**
ParentOf	Weakness Variant	589	Call to Non-ubiquitous API	**Research Concepts (primary)1000**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
7 Pernicious Kingdoms			Inconsistent Implementations

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	7 Pernicious Kingdoms		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Eric Dalci	Cigital	External
	updated Potential Mitigations, Time of Introduction		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Relationships, Other Notes, Taxonomy Mappings		
Previous Entry Names			
Change Date	**Previous Entry Name**		
2008-04-11	Inconsistent Implementations		

BACK TO TOP

Unchecked Return Value

Risk

What might happen

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

Cause

How does it happen

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

General Recommendations

How to avoid it

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

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

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

Source Code Examples

CPP

Unchecked Memory Allocation

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

Safer Memory Allocation

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

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

Use of sizeof() on a Pointer Type

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

Description

Description Summary

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

Time of Introduction

- Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

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

Likelihood of Exploit

High

Demonstrative Examples

Example 1

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

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

(Bad Code)

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

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

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

(Good Code)

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

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

Example 2

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

(Bad Code)

```

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

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

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

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

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

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

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

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

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

(Attack)

```
pass5
passABCDEFGH
passWORD
```

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

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

Potential Mitigations

Phase: Implementation

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

Other Notes

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

Weakness Ordinalities

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

Relationships

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

Taxonomy Mappings

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

White Box Definitions

A weakness where code path has:

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

References

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

Content History

Submissions

Submission Date	Submitter	Organization	Source
	CLASP		Externally Mined

Modifications

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

BACK TO TOP

Potential Off by One Error in Loops

Risk

What might happen

An off by one error may result in overwriting or over-reading of unintended memory; in most cases, this can result in unexpected behavior and even application crashes. In other cases, where allocation can be controlled by an attacker, a combination of variable assignment and an off by one error can result in execution of malicious code.

Cause

How does it happen

Often when designating variables to memory, a calculation error may occur when determining size or length that is off by one.

For example in loops, when allocating an array of size 2, its cells are counted as 0,1 - therefore, if a For loop iterator on the array is incorrectly set with the start condition i=0 and the continuation condition i<=2, three cells will be accessed instead of 2, and an attempt will be made to write or read cell [2], which was not originally allocated, resulting in potential corruption of memory outside the bounds of the originally assigned array.

Another example occurs when a null-byte terminated string, in the form of a character array, is copied without its terminating null-byte. Without the null-byte, the string representation is unterminated, resulting in certain functions to over-read memory as they expect the missing null terminator.

General Recommendations

How to avoid it

- Always ensure that a given iteration boundary is correct:
    - With array iterations, consider that arrays begin with cell 0 and end with cell n-1, for a size n array.
    - With character arrays and null-byte terminated string representations, consider that the null byte is required and should not be overwritten or ignored; ensure functions in use are not vulnerable to off-by-one, specifically for instances where null-bytes are automatically appended after the buffer, instead of in place of its last character.
- Where possible, use safe functions that manage memory and are not prone to off-by-one errors.

Source Code Examples

CPP

Off-By-One in For Loop

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
for (int i = 0; i <= 5; i++)
{
	ptr[i] = i * 2 + 1; // ptr[5] will be set, but is out of bounds
}
```

Proper Iteration in For Loop

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
for (int i = 0; i < 5; i++)
{
        ptr[i] = i * 2 + 1; // ptr[0-4] are well defined
}
```

Off-By-One in strncat

```
strncat(buf, input, sizeof(buf) - strlen(buf)); // actual value should be sizeof(buf)-
strlen(buf)-1 - this form will overwrite the terminating nullbyte
```

NULL Pointer Dereference

Risk

What might happen

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

Cause

How does it happen

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

General Recommendations

How to avoid it

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

Source Code Examples

CPP

Explicit NULL Dereference

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

Implicit NULL Dereference

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

Java

Explicit Null Dereference

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

Heuristic 2nd Order Buffer Overflow read

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

Potential Precision Problem

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

Indicator of Poor Code Quality

Weakness ID: 398 *(Weakness Class)* **Status:** Draft

Description

Description Summary

The code has features that do not directly introduce a weakness or vulnerability, but indicate that the product has not been carefully developed or maintained.

Extended Description

Programs are more likely to be secure when good development practices are followed. If a program is complex, difficult to maintain, not portable, or shows evidence of neglect, then there is a higher likelihood that weaknesses are buried in the code.

Time of Introduction

- Architecture and Design
- Implementation

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	18	Source Code	**Development Concepts (primary)699**
ChildOf	Weakness Class	710	Coding Standards Violation	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	107	Struts: Unused Validation Form	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	110	Struts: Validator Without Form Field	**Research Concepts (primary)1000**
ParentOf	Category	399	Resource Management Errors	**Development Concepts (primary)699**
ParentOf	Weakness Base	401	Failure to Release Memory Before Removing Last Reference ('Memory Leak')	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	404	Improper Resource Shutdown or Release	Development Concepts699 **Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Variant	415	Double Free	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	416	Use After Free	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Variant	457	Use of Uninitialized Variable	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	474	Use of Function with Inconsistent Implementations	**Development Concepts (primary)699** **Seven Pernicious Kingdoms (primary)700** **Research Concepts (primary)1000**
ParentOf	Weakness Base	475	Undefined Behavior for Input to API	**Development Concepts (primary)699** **Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	476	NULL Pointer	**Development**

			Dereference	**Concepts (primary)699** **Seven Pernicious Kingdoms (primary)700** **Research Concepts (primary)1000**
ParentOf	Weakness Base	477	Use of Obsolete Functions	**Development Concepts (primary)699** **Seven Pernicious Kingdoms (primary)700** **Research Concepts (primary)1000**
ParentOf	Weakness Variant	478	Missing Default Case in Switch Statement	**Development Concepts (primary)699**
ParentOf	Weakness Variant	479	Unsafe Function Call from a Signal Handler	**Development Concepts (primary)699**
ParentOf	Weakness Variant	483	Incorrect Block Delimitation	**Development Concepts (primary)699**
ParentOf	Weakness Base	484	Omitted Break Statement in Switch	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Variant	546	Suspicious Comment	**Development Concepts (primary)699** **Research Concepts (primary)1000**
ParentOf	Weakness Variant	547	Use of Hard-coded, Security-relevant Constants	**Development Concepts (primary)699** **Research Concepts (primary)1000**
ParentOf	Weakness Variant	561	Dead Code	**Development Concepts (primary)699** **Research Concepts (primary)1000**
ParentOf	Weakness Base	562	Return of Stack Variable Address	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Variant	563	Unused Variable	**Development Concepts (primary)699** **Research Concepts (primary)1000**
ParentOf	Category	569	Expression Issues	**Development Concepts (primary)699**
ParentOf	Weakness Variant	585	Empty Synchronized Block	**Development Concepts (primary)699** **Research Concepts (primary)1000**
ParentOf	Weakness Variant	586	Explicit Call to Finalize()	**Development Concepts (primary)699**
ParentOf	Weakness Variant	617	Reachable Assertion	**Development Concepts (primary)699**
ParentOf	Weakness Base	676	Use of Potentially Dangerous Function	**Development Concepts (primary)699** **Research Concepts (primary)1000**
MemberOf	View	700	Seven Pernicious Kingdoms	**Seven Pernicious Kingdoms (primary)700**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name

7 Pernicious Kingdoms			Code Quality

Content History

Submissions

Submission Date	Submitter	Organization	Source
	7 Pernicious Kingdoms		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Description, Relationships, Taxonomy Mappings		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Relationships		

Previous Entry Names

Change Date	Previous Entry Name
2008-04-11	Code Quality

BACK TO TOP

Improper Validation of Array Index

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

Description

Description Summary

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

Alternate Terms

out-of-bounds array index

index-out-of-range

array index underflow

Time of Introduction

- Implementation

Applicable Platforms

Languages

C: *(Often)*

C++: *(Often)*

Language-independent

Common Consequences

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

Likelihood of Exploit

High

Detection Methods

Automated Static Analysis

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

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

Effectiveness: High

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

Automated Dynamic Analysis

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

Black Box

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

Demonstrative Examples

Example 1

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

(Bad Code)

Example Language: **C**

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

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

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

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

(Good Code)

Example Language: **C**

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

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

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

Example 2

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

(Bad Code)

Example Language: **Java**

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

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

Example 3

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

(Bad Code)

Example Language: **Java**

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

String productSummary = new String("");

try {
String productSummary = getProductSummary(index);

} catch (Exception ex) {...}

return productSummary;
}

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

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

(Good Code)

Example Language: **Java**

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

String productSummary = new String("");

try {
String productSummary = getProductSummary(index);

} catch (Exception ex) {...}

return productSummary;
}

public String getProductSummary(int index) {
String productSummary = "";

if ((index >= 0) && (index < MAX_PRODUCTS)) {
productSummary = products[index];
}
else {
System.err.println("index is out of bounds");
throw new IndexOutOfBoundsException();
}

return productSummary;
}
```

An alternative in Java would be to use one of the collection objects such as ArrayList that will automatically generate an exception if an attempt is made to access an array index that is out of bounds.

(Good Code)

Example Language: **Java**

```
ArrayList productArray = new ArrayList(MAX_PRODUCTS);
...
try {
productSummary = (String) productArray.get(index);
} catch (IndexOutOfBoundsException ex) {...}
```

Observed Examples

Reference	Description
CVE-2005-0369	large ID in packet used as array index
CVE-2001-1009	negative array index as argument to POP LIST command
CVE-2003-0721	Integer signedness error leads to negative array index
CVE-2004-1189	product does not properly track a count and a maximum number, which can lead to resultant array index overflow.
CVE-2007-5756	chain: device driver for packet-capturing software allows access to an unintended IOCTL with resultant array index error.

Potential Mitigations

Phase: Architecture and Design

Strategies: Input Validation; Libraries or Frameworks

Use an input validation framework such as Struts or the OWASP ESAPI Validation API. If you use Struts, be mindful of weaknesses covered by the CWE-101 category.

Phase: Architecture and Design

For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side, in order to avoid CWE-602. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the server.

Even though client-side checks provide minimal benefits with respect to server-side security, they are still useful. First, they can support intrusion detection. If the server receives input that should have been rejected by the client, then it may be an indication of an attack. Second, client-side error-checking can provide helpful feedback to the user about the expectations for valid input. Third, there may be a reduction in server-side processing time for accidental input errors, although this is typically a small savings.

Phase: Requirements

Strategy: Language Selection

Use a language with features that can automatically mitigate or eliminate out-of-bounds indexing errors.

For example, Ada allows the programmer to constrain the values of a variable and languages such as Java and Ruby will allow the programmer to handle exceptions when an out-of-bounds index is accessed.

Phase: Implementation

Strategy: Input Validation

Assume all input is malicious. Use an "accept known good" input validation strategy (i.e., use a whitelist). Reject any input that does not strictly conform to specifications, or transform it into something that does. Use a blacklist to reject any unexpected inputs and detect potential attacks.

When accessing a user-controlled array index, use a stringent range of values that are within the target array. Make sure that you do not allow negative values to be used. That is, verify the minimum as well as the maximum of the range of acceptable values.

Phase: Implementation

Be especially careful to validate your input when you invoke code that crosses language boundaries, such as from an interpreted language to native code. This could create an unexpected interaction between the language boundaries. Ensure that you are not violating any of the expectations of the language with which you are interfacing. For example, even though Java may not be susceptible to buffer overflows, providing a large argument in a call to native code might trigger an overflow.

Weakness Ordinalities

Ordinality	Description
Resultant	The most common condition situation leading to unchecked array indexing is the use of loop index variables as buffer indexes. If the end condition for the loop is subject to a flaw, the index can grow or shrink unbounded, therefore causing a buffer overflow or underflow. Another common situation leading to this condition is the use of a function's return value, or the resulting value of a calculation directly as an index in to a buffer.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	20	Improper Input Validation	**Development Concepts (primary)699** **Research Concepts (primary)1000**
ChildOf	Category	189	Numeric Errors	Development Concepts699
ChildOf	Category	633	Weaknesses that Affect Memory	**Resource-specific Weaknesses (primary)631**
ChildOf	Category	738	CERT C Secure Coding Section 04 - Integers (INT)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
ChildOf	Category	740	CERT C Secure Coding Section 06 - Arrays (ARR)	Weaknesses Addressed by the CERT C Secure Coding Standard734
ChildOf	Category	802	2010 Top 25 - Risky Resource Management	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
CanPrecede	Weakness Class	119	Failure to Constrain Operations within the Bounds of a Memory Buffer	Research Concepts1000
CanPrecede	Weakness Variant	789	Uncontrolled Memory Allocation	Research Concepts1000
PeerOf	Weakness Base	124	Buffer Underwrite ('Buffer Underflow')	Research Concepts1000

Theoretical Notes

An improperly validated array index might lead directly to the always-incorrect behavior of "access of array using out-of-bounds index."

Affected Resources

- Memory

f Causal Nature

Explicit

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Unchecked array indexing
PLOVER			INDEX - Array index overflow
CERT C Secure Coding	ARR00-C		Understand how arrays work
CERT C Secure Coding	ARR30-C		Guarantee that array indices are within the valid range
CERT C Secure Coding	ARR38-C		Do not add or subtract an integer to a pointer if the resulting value does not refer to a valid array element
CERT C Secure Coding	INT32-C		Ensure that operations on signed integers do not result in overflow

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
100	Overflow Buffers	

References

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 5, "Array Indexing Errors" Page 144. 2nd Edition. Microsoft. 2002.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	CLASP		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Sean Eidemiller	Cigital	External
	added/updated demonstrative examples		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Alternate Terms, Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Description, Name, Relationships		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Observed Examples, Other Notes, Potential Mitigations, Theoretical Notes, Weakness Ordinalities		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Demonstrative Examples, Detection Factors, Likelihood of Exploit, Potential Mitigations, References, Related Attack Patterns, Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Related Attack Patterns		

Previous Entry Names

Change Date	Previous Entry Name
2009-10-29	Unchecked Array Indexing

BACK TO TOP

Improper Access Control (Authorization)

Weakness ID: 285 *(Weakness Class)* **Status:** Draft

Description

Description Summary

The software does not perform or incorrectly performs access control checks across all potential execution paths.

Extended Description

When access control checks are not applied consistently - or not at all - users are able to access data or perform actions that they should not be allowed to perform. This can lead to a wide range of problems, including information leaks, denial of service, and arbitrary code execution.

Alternate Terms

AuthZ:	"AuthZ" is typically used as an abbreviation of "authorization" within the web application security community. It is also distinct from "AuthC," which is an abbreviation of "authentication." The use of "Auth" as an abbreviation is discouraged, since it could be used for either authentication or authorization.

Time of Introduction

- Architecture and Design
- Implementation
- Operation

Applicable Platforms

Languages

Language-independent

Technology Classes

Web-Server: *(Often)*

Database-Server: *(Often)*

Modes of Introduction

A developer may introduce authorization weaknesses because of a lack of understanding about the underlying technologies. For example, a developer may assume that attackers cannot modify certain inputs such as headers or cookies.

Authorization weaknesses may arise when a single-user application is ported to a multi-user environment.

Common Consequences

Scope	Effect
Confidentiality	An attacker could read sensitive data, either by reading the data directly from a data store that is not properly restricted, or by accessing insufficiently-protected, privileged functionality to read the data.
Integrity	An attacker could modify sensitive data, either by writing the data directly to a data store that is not properly restricted, or by accessing insufficiently-protected, privileged functionality to write the data.
Integrity	An attacker could gain privileges by modifying or reading critical data directly, or by accessing insufficiently-protected, privileged functionality.

Likelihood of Exploit

High

Detection Methods

Automated Static Analysis

Automated static analysis is useful for detecting commonly-used idioms for authorization. A tool may be able to analyze related configuration files, such as .htaccess in Apache web servers, or detect the usage of commonly-used authorization libraries.

Generally, automated static analysis tools have difficulty detecting custom authorization schemes. In addition, the software's design may include some functionality that is accessible to any user and does not require an authorization check; an automated technique that detects the absence of authorization may report false positives.

Effectiveness: Limited

Automated Dynamic Analysis

Automated dynamic analysis may find many or all possible interfaces that do not require authorization, but manual analysis is required to determine if the lack of authorization violates business logic

Manual Analysis

This weakness can be detected using tools and techniques that require manual (human) analysis, such as penetration testing, threat modeling, and interactive tools that allow the tester to record and modify an active session.

Specifically, manual static analysis is useful for evaluating the correctness of custom authorization mechanisms.

Effectiveness: Moderate

These may be more effective than strictly automated techniques. This is especially the case with weaknesses that are related to design and business rules. However, manual efforts might not achieve desired code coverage within limited time constraints.

Demonstrative Examples

Example 1

The following program could be part of a bulletin board system that allows users to send private messages to each other. This program intends to authenticate the user before deciding whether a private message should be displayed. Assume that LookupMessageObject() ensures that the $id argument is numeric, constructs a filename based on that id, and reads the message details from that file. Also assume that the program stores all private messages for all users in the same directory.

(Bad Code)

Example Language: **Perl**

```
sub DisplayPrivateMessage {
my($id) = @_;
my $Message = LookupMessageObject($id);
print "From: " . encodeHTML($Message->{from}) . "<br>\n";
print "Subject: " . encodeHTML($Message->{subject}) . "\n";
print "<hr>\n";
print "Body: " . encodeHTML($Message->{body}) . "\n";
}

my $q = new CGI;
# For purposes of this example, assume that CWE-309 and
# CWE-523 do not apply.
if (! AuthenticateUser($q->param('username'), $q->param('password'))) {
ExitError("invalid username or password");
}

my $id = $q->param('id');
DisplayPrivateMessage($id);
```

While the program properly exits if authentication fails, it does not ensure that the message is addressed to the user. As a result, an authenticated attacker could provide any arbitrary identifier and read private messages that were intended for other users.

One way to avoid this problem would be to ensure that the "to" field in the message object matches the username of the authenticated user.

Observed Examples

Reference	Description
CVE-2009-3168	Web application does not restrict access to admin scripts, allowing authenticated users to reset administrative passwords.

CVE-2009-2960	Web application does not restrict access to admin scripts, allowing authenticated users to modify passwords of other users.
CVE-2009-3597	Web application stores database file under the web root with insufficient access control (CWE-219), allowing direct request.
CVE-2009-2282	Terminal server does not check authorization for guest access.
CVE-2009-3230	Database server does not use appropriate privileges for certain sensitive operations.
CVE-2009-2213	Gateway uses default "Allow" configuration for its authorization settings.
CVE-2009-0034	Chain: product does not properly interpret a configuration option for a system group, allowing users to gain privileges.
CVE-2008-6123	Chain: SNMP product does not properly parse a configuration option for which hosts are allowed to connect, allowing unauthorized IP addresses to connect.
CVE-2008-5027	System monitoring software allows users to bypass authorization by creating custom forms.
CVE-2008-7109	Chain: reliance on client-side security (CWE-602) allows attackers to bypass authorization using a custom client.
CVE-2008-3424	Chain: product does not properly handle wildcards in an authorization policy list, allowing unintended access.
CVE-2009-3781	Content management system does not check access permissions for private files, allowing others to view those files.
CVE-2008-4577	ACL-based protection mechanism treats negative access rights as if they are positive, allowing bypass of intended restrictions.
CVE-2008-6548	Product does not check the ACL of a page accessed using an "include" directive, allowing attackers to read unauthorized files.
CVE-2007-2925	Default ACL list for a DNS server does not set certain ACLs, allowing unauthorized DNS queries.
CVE-2006-6679	Product relies on the X-Forwarded-For HTTP header for authorization, allowing unintended access by spoofing the header.
CVE-2005-3623	OS kernel does not check for a certain privilege before setting ACLs for files.
CVE-2005-2801	Chain: file-system code performs an incorrect comparison (CWE-697), preventing defauls ACLs from being properly applied.
CVE-2001-1155	Chain: product does not properly check the result of a reverse DNS lookup because of operator precedence (CWE-783), allowing bypass of DNS-based access restrictions.

Potential Mitigations

Phase: Architecture and Design

Divide your application into anonymous, normal, privileged, and administrative areas. Reduce the attack surface by carefully mapping roles with data and functionality. Use role-based access control (RBAC) to enforce the roles at the appropriate boundaries.

Note that this approach may not protect against horizontal authorization, i.e., it will not protect a user from attacking others with the same role.

Phase: Architecture and Design

Ensure that you perform access control checks related to your business logic. These checks may be different than the access control checks that you apply to more generic resources such as files, connections, processes, memory, and database records. For example, a database may restrict access for medical records to a specific database user, but each record might only be intended to be accessible to the patient and the patient's doctor.

Phase: Architecture and Design

Strategy: Libraries or Frameworks

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.

For example, consider using authorization frameworks such as the JAAS Authorization Framework and the OWASP ESAPI Access Control feature.

Phase: Architecture and Design

For web applications, make sure that the access control mechanism is enforced correctly at the server side on every page. Users should not be able to access any unauthorized functionality or information by simply requesting direct access to that page.

One way to do this is to ensure that all pages containing sensitive information are not cached, and that all such pages restrict access to requests that are accompanied by an active and authenticated session token associated with a user who has the required permissions to access that page.

Phases: System Configuration; Installation

Use the access control capabilities of your operating system and server environment and define your access control lists accordingly. Use a "default deny" policy when defining these ACLs.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	254	Security Features	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Weakness Class	284	Access Control (Authorization) Issues	**Development Concepts (primary)699** **Research Concepts (primary)1000**
ChildOf	Category	721	OWASP Top Ten 2007 Category A10 - Failure to Restrict URL Access	**Weaknesses in OWASP Top Ten (2007) (primary)629**
ChildOf	Category	723	OWASP Top Ten 2004 Category A2 - Broken Access Control	**Weaknesses in OWASP Top Ten (2004) (primary)711**
ChildOf	Category	753	2009 Top 25 - Porous Defenses	**Weaknesses in the 2009 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)750**
ChildOf	Category	803	2010 Top 25 - Porous Defenses	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
ParentOf	Weakness Variant	219	Sensitive Data Under Web Root	**Research Concepts (primary)1000**
ParentOf	Weakness Base	551	Incorrect Behavior Order: Authorization Before Parsing and Canonicalization	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Class	638	Failure to Use Complete Mediation	Research Concepts1000
ParentOf	Weakness Base	804	Guessable CAPTCHA	**Development Concepts (primary)699** **Research Concepts (primary)1000**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
7 Pernicious Kingdoms			Missing Access Control
OWASP Top Ten 2007	A10	CWE More Specific	Failure to Restrict URL Access
OWASP Top Ten 2004	A2	CWE More Specific	Broken Access Control

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
1	Accessing Functionality Not Properly Constrained by ACLs	
13	Subverting Environment Variable Values	

17	Accessing, Modifying or Executing Executable Files
87	Forceful Browsing
39	Manipulating Opaque Client-based Data Tokens
45	Buffer Overflow via Symbolic Links
51	Poison Web Service Registry
59	Session Credential Falsification through Prediction
60	Reusing Session IDs (aka Session Replay)
77	Manipulating User-Controlled Variables
76	Manipulating Input to File System Calls
104	Cross Zone Scripting

References

NIST. "Role Based Access Control and Role Based Security". <http://csrc.nist.gov/groups/SNS/rbac/>.

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 4, "Authorization" Page 114; Chapter 6, "Determining Appropriate Access Control" Page 171. 2nd Edition. Microsoft. 2002.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	7 Pernicious Kingdoms		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-15		Veracode	External
	Suggested OWASP Top Ten 2004 mapping		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Relationships, Other Notes, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences, Description, Likelihood of Exploit, Name, Other Notes, Potential Mitigations, References, Relationships		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Potential Mitigations		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Description, Related Attack Patterns		
2009-07-27	CWE Content Team	MITRE	Internal
	updated Relationships		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Type		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Demonstrative Examples, Detection Factors, Modes of Introduction, Observed Examples, Relationships		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Alternate Terms, Detection Factors, Potential Mitigations, References, Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Potential Mitigations		

Previous Entry Names

Change Date	Previous Entry Name
2009-01-12	Missing or Inconsistent Access Control

BACK TO TOP

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

```java
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