cosmopolitan Scan Report

Project Name	cosmopolitan
Scan Start	Friday, June 21, 2024 11:42:46 PM
Preset	Checkmarx Default
Scan Time	00h:13m:10s
Lines Of Code Scanned	158739
Files Scanned	62
Report Creation Time	Saturday, June 22, 2024 12:04:21 AM
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070
Team	CxServer
Checkmarx Version	8.7.0
Scan Type	Full
Source Origin	LocalPath
Density	5/1000 (Vulnerabilities/LOC)
Visibility	Public

Filter Settings

Severity

Included: High, Medium, Low, Information

Excluded: None

Result State

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

Excluded: None

Assigned to

Included: All

Categories

Included:

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

Excluded:

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

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

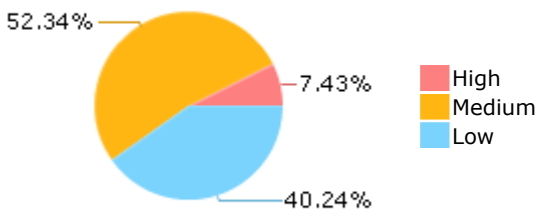
Results Limit

Results limit per query was set to 50

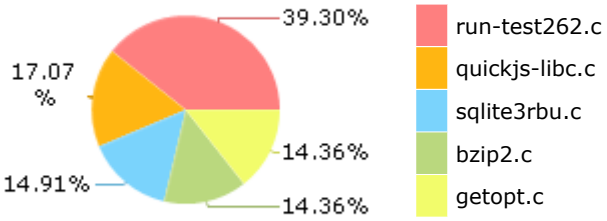
Selected Queries

Selected queries are listed in Result Summary

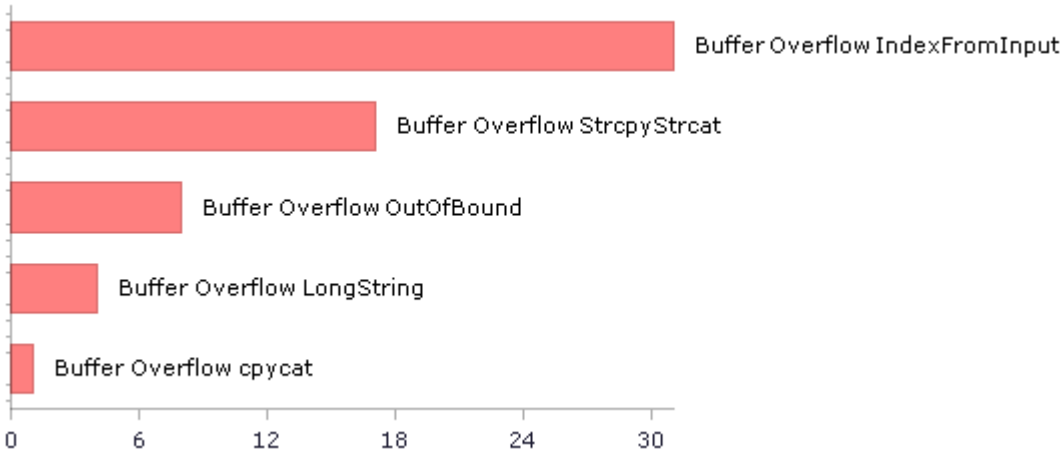
Result Summary



Most Vulnerable Files



Top 5 Vulnerabilities



Scan Summary - OWASP Top 10 2017

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

Category	Threat Agent	Exploitability	Weakness Prevalence	Weakness Detectability	Technical Impact	Business Impact	Issues Found	Best Fix Locations
A1-Injection	App. Specific	EASY	COMMON	EASY	SEVERE	App. Specific	182	117
A2-Broken Authentication	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	166	166
A3-Sensitive Data Exposure	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	App. Specific	1	1
A4-XML External Entities (XXE)	App. Specific	AVERAGE	COMMON	EASY	SEVERE	App. Specific	0	0
A5-Broken Access Control*	App. Specific	AVERAGE	COMMON	AVERAGE	SEVERE	App. Specific	7	3
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	221	221
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	7	3
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	221	221
A10-Unvalidated Redirects and Forwards	USERS BROWSERS	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	AFFECTED DATA AND FUNCTIONS	0	0

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

Scan Summary - PCI DSS v3.2

Category	Issues Found	Best Fix Locations
PCI DSS (3.2) - 6.5.1 - Injection flaws - particularly SQL injection	4	4
PCI DSS (3.2) - 6.5.2 - Buffer overflows	121	100
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.	19	19
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.	3	3
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.	4	4
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.	147	147
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.	5	5

***** 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)	170	170
AC-4 Information Flow Enforcement (P1)	0	0
AC-6 Least Privilege (P1)	0	0
AU-9 Protection of Audit Information (P1)	0	0
CM-6 Configuration Settings (P2)	0	0
IA-5 Authenticator Management (P1)	0	0
IA-6 Authenticator Feedback (P2)	0	0
IA-8 Identification and Authentication (Non-Organizational Users) (P1)	0	0
SC-12 Cryptographic Key Establishment and Management (P1)	0	0
SC-13 Cryptographic Protection (P1)	0	0
SC-17 Public Key Infrastructure Certificates (P1)	0	0
SC-18 Mobile Code (P2)	0	0
SC-23 Session Authenticity (P1)*	0	0
SC-28 Protection of Information at Rest (P1)	0	0
SC-4 Information in Shared Resources (P1)	1	1
SC-5 Denial of Service Protection (P1)*	119	76
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*	65	44
SI-11 Error Handling (P2)*	41	41
SI-15 Information Output Filtering (P0)	0	0
SI-16 Memory Protection (P1)	7	6

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

Scan Summary - 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

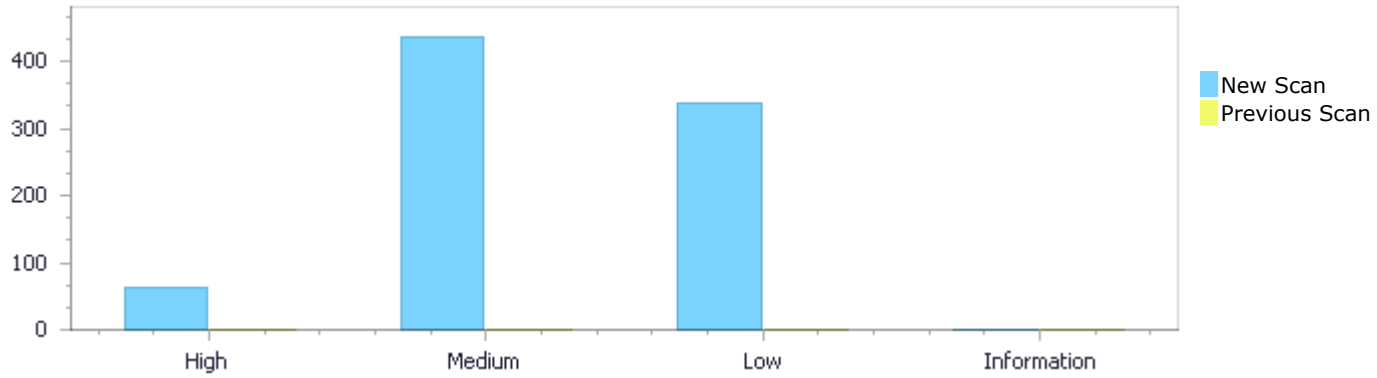
Scan Summary - Custom

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	62	437	336	0	835
Recurrent Issues	0	0	0	0	0
Total	62	437	336	0	835
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	62	437	336	0	835
Urgent	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0
Total	62	437	336	0	835

Result Summary

Vulnerability Type	Occurrences	Severity
Buffer Overflow IndexFromInput	31	High
Buffer Overflow StrcpyStrcat	17	High
Buffer Overflow OutOfBound	8	High
Buffer Overflow LongString	4	High
Buffer Overflow cpycat	1	High

Buffer Overflow Indexes	1	High
Dangerous Functions	221	Medium
Buffer Overflow boundcpy WrongSizeParam	76	Medium
Memory Leak	40	Medium
Use of Zero Initialized Pointer	36	Medium
MemoryFree on StackVariable	24	Medium
Wrong Size t Allocation	21	Medium
Use of Uninitialized Variable	6	Medium
Double Free	3	Medium
Short Overflow	3	Medium
Buffer Overflow AddressOfLocalVarReturned	2	Medium
Integer Overflow	2	Medium
Char Overflow	1	Medium
Stored Buffer Overflow boundcpy	1	Medium
Use After Free	1	Medium
Improper Resource Access Authorization	147	Low
Unchecked Return Value	41	Low
NULL Pointer Dereference	30	Low
TOCTOU	25	Low
Unchecked Array Index	21	Low
Incorrect Permission Assignment For Critical Resources	19	Low
Use of Sizeof On a Pointer Type	15	Low
Sizeof Pointer Argument	8	Low
Potential Path Traversal	7	Low
Exposure of System Data to Unauthorized Control Sphere	4	Low
Inconsistent Implementations	4	Low
Potential Off by One Error in Loops	4	Low
Arithmenic Operation On Boolean	3	Low
Heuristic 2nd Order Buffer Overflow malloc	3	Low
Heuristic Buffer Overflow malloc	3	Low
Insecure Temporary File	1	Low
Unreleased Resource Leak	1	Low

10 Most Vulnerable Files

High and Medium Vulnerabilities

File Name	Issues Found
cosmopolitan/run-test262.c	81
cosmopolitan/sqlite3rbu.c	44
cosmopolitan/getopt.c	39
cosmopolitan/ssl_srv.c	34
cosmopolitan/sqlite3expert.c	34
cosmopolitan/quickjs-libc.c	30
cosmopolitan/bzip2.c	27
cosmopolitan/sds.c	24
cosmopolitan/bzlib.c	22
cosmopolitan/lstrlib.c	21

Scan Results Details

Buffer Overflow IndexFromInput

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

Categories

OWASP Top 10 2017: A1-Injection

Description

Buffer Overflow IndexFromInput\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=7
Status	New

The size of the buffer used by _getopt_internal in PostfixExpr, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	560
Object	argc	PostfixExpr

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

```
....
934.  main (int argc, char **argv)
```

File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
560.          optarg = argv[optind++];
```

Buffer Overflow IndexFromInput\Path 2:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=8
Status	New

The size of the buffer used by _getopt_internal in optind, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	630
Object	argc	optind

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

```
....
934.  main (int argc, char **argv)
```

File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
630.                  argv[0], argv[optind]);
```

Buffer Overflow IndexFromInput\Path 3:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=9
Status	New

The size of the buffer used by _getopt_internal in PostfixExpr, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	670
Object	argc	PostfixExpr

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

```
....
934.  main (int argc, char **argv)
```

File Name cosmopolitan/getopt.c

Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
670.              optarg = argv[optind++];
```

Buffer Overflow IndexFromInput\Path 4:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=10
Status	New

The size of the buffer used by _getopt_internal in optind, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	709
Object	argc	optind

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

```
....
934.  main (int argc, char **argv)
```

File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
709.                    argv[0], argv[optind][0], nextchar);
```

Buffer Overflow IndexFromInput\Path 5:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=11
Status	New

The size of the buffer used by _getopt_internal in optind, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c

Line	934	702
Object	argc	optind

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

```
....
934.  main (int argc, char **argv)
```

File Name cosmopolitan/getopt.c

Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
702.              if (argv[optind][1] == '-')
```

Buffer Overflow IndexFromInput\Path 6:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=12
Status	New

The size of the buffer used by _getopt_internal in optind, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	815
Object	argc	optind

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

```
....
934.  main (int argc, char **argv)
```

File Name cosmopolitan/getopt.c

Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
815.                      argv[0], argv[optind]);
```

Buffer Overflow IndexFromInput\Path 7:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=13
Status	New

The size of the buffer used by _getopt_internal in PostfixExpr, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	843
Object	argc	PostfixExpr

Code Snippet

File Name cosmopolitan/getopt.c

Method main (int argc, char **argv)

```
....
934.  main (int argc, char **argv)
```

File Name cosmopolitan/getopt.c

Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
843.                optarg = argv[optind++];
```

Buffer Overflow IndexFromInput\Path 8:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=14
Status	New

The size of the buffer used by _getopt_internal in PostfixExpr, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	780
Object	argc	PostfixExpr

Code Snippet

File Name cosmopolitan/getopt.c

Method main (int argc, char **argv)

```
....
934.  main (int argc, char **argv)
```

File Name cosmopolitan/getopt.c

Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
780.          optarg = argv[optind++];
```

Buffer Overflow IndexFromInput\Path 9:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=15
Status	New

The size of the buffer used by _getopt_internal in PostfixExpr, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	909
Object	argc	PostfixExpr

Code Snippet

File Name cosmopolitan/getopt.c

Method main (int argc, char **argv)

```
....
934.  main (int argc, char **argv)
```

File Name cosmopolitan/getopt.c

Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
909.              optarg = argv[optind++];
```

Buffer Overflow IndexFromInput\Path 10:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=16
Status	New

The size of the buffer used by _getopt_internal in optind, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	697
Object	argc	optind

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

```
....
934.  main (int argc, char **argv)
```

File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
697.        if (!long_only || argv[optind][1] == '-'
```

Buffer Overflow IndexFromInput\Path 11:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=17
Status	New

The size of the buffer used by _getopt_internal in optind, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	587
Object	argc	optind

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

```
....
934.  main (int argc, char **argv)
```

File Name	cosmopolitan/getopt.c
Method	_getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
587.         && (argv[optind][1] == '-'
```

Buffer Overflow IndexFromInput\Path 12:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=18
Status	New

The size of the buffer used by _getopt_internal in optind, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	588
Object	argc	optind

Code Snippet

File Name	cosmopolitan/getopt.c
Method	main (int argc, char **argv)

```
....
934.  main (int argc, char **argv)
```

File Name	cosmopolitan/getopt.c
Method	_getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
588.           || (long_only && (argv[optind][2] || !my_index (optstring,
argv[optind][1])))))
```

Buffer Overflow IndexFromInput\Path 13:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=19
Status	New

The size of the buffer used by _getopt_internal in optind, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination

File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	588
Object	argc	optind

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

```
....
934.  main (int argc, char **argv)
```

File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
588.          || (long_only && (argv[optind][2] || !my_index (optstring,
argv[optind][1])))))
```

Buffer Overflow IndexFromInput\Path 14:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=20
Status	New

The size of the buffer used by _getopt_internal in optind, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	567
Object	argc	optind

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

```
....
934.  main (int argc, char **argv)
```

File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
567.        nextchar = (argv[optind] + 1
```

Buffer Overflow IndexFromInput\Path 15:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=21
Status	New

The size of the buffer used by _getopt_internal in optind, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	568
Object	argc	optind

Code Snippet

File Name cosmopolitan/getopt.c

Method main (int argc, char **argv)

```
....
934.  main (int argc, char **argv)
```

File Name cosmopolitan/getopt.c

Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
568.                + (longopts != NULL && argv[optind][1] == '-'));
```

Buffer Overflow IndexFromInput\Path 16:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=22
Status	New

The size of the buffer used by _getopt_internal in optind, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	556

Object	argc	optind

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

```
....
934.  main (int argc, char **argv)
```

File Name cosmopolitan/getopt.c

Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
556.        if (NONOPTION_P)
```

Buffer Overflow IndexFromInput\Path 17:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=23
Status	New

The size of the buffer used by _getopt_internal in optind, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	556
Object	argc	optind

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

```
....
934.  main (int argc, char **argv)
```

File Name cosmopolitan/getopt.c

Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
556.        if (NONOPTION_P)
```

Buffer Overflow IndexFromInput\Path 18:

Severity	High

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=24
Status	New

The size of the buffer used by _getopt_internal in optind, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	556
Object	argc	optind

Code Snippet

File Name cosmopolitan/getopt.c

Method main (int argc, char **argv)

```
....
934.  main (int argc, char **argv)
```

File Name cosmopolitan/getopt.c

Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
556.        if (NONOPTION_P)
```

Buffer Overflow IndexFromInput\Path 19:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=25
Status	New

The size of the buffer used by _getopt_internal in optind, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	528
Object	argc	optind

Code Snippet

File Name cosmopolitan/getopt.c

Method main (int argc, char **argv)

```
....
934.  main (int argc, char **argv)
```

File Name cosmopolitan/getopt.c

Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
528.        if (optind != argc && !strcmp (argv[optind], "--"))
```

Buffer Overflow IndexFromInput\Path 20:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=26
Status	New

The size of the buffer used by _getopt_internal in optind, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	518
Object	argc	optind

Code Snippet

File Name cosmopolitan/getopt.c

Method main (int argc, char **argv)

```
....
934.  main (int argc, char **argv)
```

File Name cosmopolitan/getopt.c

Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
518.          while (optind < argc && NONOPTION_P)
```

Buffer Overflow IndexFromInput\Path 21:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=27
Status	New

The size of the buffer used by _getopt_internal in optind, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	518
Object	argc	optind

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

```
....
934.  main (int argc, char **argv)
```

File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
518.          while (optind < argc && NONOPTION_P)
```

Buffer Overflow IndexFromInput\Path 22:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=28
Status	New

The size of the buffer used by _getopt_internal in optind, at line 469 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	518
Object	argc	optind

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

```
....
934.  main (int argc, char **argv)
```

File Name cosmopolitan/getopt.c

Method	_getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
518.          while (optind < argc && NONOPTION_P)
```

Buffer Overflow IndexFromInput\Path 23:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=29
Status	New

The size of the buffer used by main in PostfixExpr, at line 934 of cosmopolitan/getopt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argc, at line 934 of cosmopolitan/getopt.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	934	989
Object	argc	PostfixExpr

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

```
....
934.  main (int argc, char **argv)
....
989.        printf ("%s ", argv[optind++]);
```

Buffer Overflow IndexFromInput\Path 24:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=30
Status	New

The size of the buffer used by *U in PostfixExpr, at line 26 of cosmopolitan/printf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 98 of cosmopolitan/printf.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/printf.c	cosmopolitan/printf.c
Line	98	31
Object	argv	PostfixExpr

Code Snippet	
File Name	cosmopolitan/printf.c
Method	int main(int argc, char *argv[]) {

```
....
98.  int main(int argc, char *argv[]) {
```

File Name cosmopolitan/printf.c

Method char *U(char *p) {

```
....
31.          switch ((c = p[i++] & 255)) {
```

Buffer Overflow IndexFromInput\Path 25:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=31
Status	New

The size of the buffer used by *U in PostfixExpr, at line 26 of cosmopolitan/printf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 98 of cosmopolitan/printf.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/printf.c	cosmopolitan/printf.c
Line	98	29
Object	argv	PostfixExpr

Code Snippet

File Name cosmopolitan/printf.c

Method int main(int argc, char *argv[]) {

```
....
98.  int main(int argc, char *argv[]) {
```

File Name cosmopolitan/printf.c

Method char *U(char *p) {

```
....
29.      switch ((c = p[i++] & 255)) {
```

Buffer Overflow IndexFromInput\Path 26:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=32
Status	New

The size of the buffer used by namelist_add in count, at line 293 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 1947 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1947	308
Object	argv	count

Code Snippet
File Name cosmopolitan/run-test262.c
Method int main(int argc, char **argv)

```
....
1947.  int main(int argc, char **argv)
```

File Name cosmopolitan/run-test262.c
Method void namelist_add(namelist_t *lp, const char *base, const char *name)

```
....
308.      lp->array[lp->count] = s;
```

Buffer Overflow IndexFromInput\Path 27:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=33
Status	New

The size of the buffer used by namelist_add in count, at line 293 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that namelist_load passes to buf, at line 315 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	327	308
Object	buf	count

Code Snippet
File Name cosmopolitan/run-test262.c
Method void namelist_load(namelist_t *lp, const char *filename)

```
....
327.      while (fgets(buf, sizeof(buf), f) != NULL) {
```

File Name cosmopolitan/run-test262.c

Method	void namelist_add(namelist_t *lp, const char *base, const char *name)

```
....
308.      lp->array[lp->count] = s;
```

Buffer Overflow IndexFromInput\Path 28:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=34
Status	New

The size of the buffer used by namelist_add in count, at line 293 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that load_config passes to buf, at line 919 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	940	308
Object	buf	count

Code Snippet

File Name	cosmopolitan/run-test262.c
Method	void load_config(const char *filename)

```
....
940.      while (fgets(buf, sizeof(buf), f) != NULL) {
```

File Name	cosmopolitan/run-test262.c
Method	void namelist_add(namelist_t *lp, const char *base, const char *name)

```
....
308.      lp->array[lp->count] = s;
```

Buffer Overflow IndexFromInput\Path 29:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=35
Status	New

The size of the buffer used by load_config in strcspn, at line 919 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 1947 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c

Line	1947	950
Object	argv	strcspn

Code Snippet
File Name cosmopolitan/run-test262.c
Method int main(int argc, char **argv)

```
....
1947.  int main(int argc, char **argv)
```

File Name cosmopolitan/run-test262.c
Method void load_config(const char *filename)

```
....
950.              p[strcspn(p, "]")] = '\0';
```

Buffer Overflow IndexFromInput\Path 30:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=36
Status	New

The size of the buffer used by load_config in strcspn, at line 919 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that load_config passes to buf, at line 919 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	940	950
Object	buf	strcspn

Code Snippet
File Name cosmopolitan/run-test262.c
Method void load_config(const char *filename)

```
....
940.      while (fgets(buf, sizeof(buf), f) != NULL) {
....
950.              p[strcspn(p, "]")] = '\0';
```

Buffer Overflow IndexFromInput\Path 31:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=37
Status	New

The size of the buffer used by js_os_readlink in res, at line 2666 of cosmopolitan/quickjs-libc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that js_os_readlink passes to buf, at line 2666 of cosmopolitan/quickjs-libc.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	2677	2682
Object	buf	res

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_os_readlink(JSContext *ctx, JSValueConst this_val,

```
....
2677.       res = readlink(path, buf, sizeof(buf) - 1);
....
2682.           buf[res] = '\0';
```

Buffer Overflow StrcpyStrcat

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

Categories

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

Description
Buffer Overflow StrcpyStrcat\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=38
Status	New

The size of the buffer used by *snocString in name, at line 1641 of cosmopolitan/bzip2.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 1688 of cosmopolitan/bzip2.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	1688	1646
Object	argv	name

Code Snippet
File Name cosmopolitan/bzip2.c
Method IntNative main (IntNative argc, Char *argv[])

```
....
1688.  IntNative main ( IntNative argc, Char *argv[] )
```

File Name cosmopolitan/bzip2.c

Method Cell *snocString (Cell *root, Char *name)

```
....
1646.        strcpy ( tmp->name, name );
```

Buffer Overflow StrcpyStrcat\Path 2:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=39
Status	New

The size of the buffer used by *str_append in str, at line 144 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 1947 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1947	156
Object	argv	str

Code Snippet

File Name cosmopolitan/run-test262.c

Method int main(int argc, char **argv)

```
....
1947.  int main(int argc, char **argv)
```

File Name cosmopolitan/run-test262.c

Method char *str_append(char **pp, const char *sep, const char *str) {

```
....
156.      strcpy(res + len, str);
```

Buffer Overflow StrcpyStrcat\Path 3:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=40
Status	New

The size of the buffer used by *str_append in p, at line 144 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *str_append passes to pp, at line 144 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	144	153
Object	pp	p

Code Snippet
File Name cosmopolitan/run-test262.c
Method char *str_append(char **pp, const char *sep, const char *str) {

```
....
144.  char *str_append(char **pp, const char *sep, const char *str) {
....
153.          strcpy(res, p);
```

Buffer Overflow StrcpyStrcat\Path 4:

Severity High
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=41
Status New

The size of the buffer used by *str_append in p, at line 144 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *str_append passes to sep, at line 144 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	144	153
Object	sep	p

Code Snippet
File Name cosmopolitan/run-test262.c
Method char *str_append(char **pp, const char *sep, const char *str) {

```
....
144.  char *str_append(char **pp, const char *sep, const char *str) {
....
153.          strcpy(res, p);
```

Buffer Overflow StrcpyStrcat\Path 5:

Severity High
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=42
Status New

The size of the buffer used by *str_append in p, at line 144 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 1947 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1947	153
Object	argv	p

Code Snippet
File Name cosmopolitan/run-test262.c
Method int main(int argc, char **argv)

```
....
1947.  int main(int argc, char **argv)
```

File Name cosmopolitan/run-test262.c
Method char *str_append(char **pp, const char *sep, const char *str) {

```
....
153.          strcpy(res, p);
```

Buffer Overflow StrcpyStrcat\Path 6:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=43
Status	New

The size of the buffer used by *str_append in len, at line 144 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 1947 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1947	156
Object	argv	len

Code Snippet
File Name cosmopolitan/run-test262.c
Method int main(int argc, char **argv)

```
....
1947.  int main(int argc, char **argv)
```

File Name	cosmopolitan/run-test262.c
Method	char *str_append(char **pp, const char *sep, const char *str) {

```
....
156.      strcpy(res + len, str);
```

Buffer Overflow StrcpyStrcat\Path 7:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=44
Status	New

The size of the buffer used by *str_append in BinaryExpr, at line 144 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *str_append passes to pp, at line 144 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	144	156
Object	pp	BinaryExpr

Code Snippet

File Name	cosmopolitan/run-test262.c
Method	char *str_append(char **pp, const char *sep, const char *str) {

```
....
144.  char *str_append(char **pp, const char *sep, const char *str) {
....
156.      strcpy(res + len, str);
```

Buffer Overflow StrcpyStrcat\Path 8:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=45
Status	New

The size of the buffer used by *str_append in BinaryExpr, at line 144 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *str_append passes to sep, at line 144 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	144	156
Object	sep	BinaryExpr

Code Snippet

File Name cosmopolitan/run-test262.c
Method char *str_append(char **pp, const char *sep, const char *str) {

```
....
144.  char *str_append(char **pp, const char *sep, const char *str) {
....
156.      strcpy(res + len, str);
```

Buffer Overflow StrcpyStrcat\Path 9:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=46
Status	New

The size of the buffer used by *str_append in BinaryExpr, at line 144 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 1947 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1947	156
Object	argv	BinaryExpr

Code Snippet
File Name cosmopolitan/run-test262.c
Method int main(int argc, char **argv)

```
....
1947.  int main(int argc, char **argv)
```

File Name cosmopolitan/run-test262.c
Method char *str_append(char **pp, const char *sep, const char *str) {

```
....
156.      strcpy(res + len, str);
```

Buffer Overflow StrcpyStrcat\Path 10:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=47
Status	New

The size of the buffer used by *str_append in res, at line 144 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *str_append passes to pp, at line 144 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	144	156
Object	pp	res

Code Snippet
File Name cosmopolitan/run-test262.c
Method char *str_append(char **pp, const char *sep, const char *str) {

```
....
144.  char *str_append(char **pp, const char *sep, const char *str) {
....
156.      strcpy(res + len, str);
```

Buffer Overflow StrcpyStrcat\Path 11:

Severity High
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=48
Status New

The size of the buffer used by *str_append in res, at line 144 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *str_append passes to sep, at line 144 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	144	156
Object	sep	res

Code Snippet
File Name cosmopolitan/run-test262.c
Method char *str_append(char **pp, const char *sep, const char *str) {

```
....
144.  char *str_append(char **pp, const char *sep, const char *str) {
....
156.      strcpy(res + len, str);
```

Buffer Overflow StrcpyStrcat\Path 12:

Severity High
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=49
Status New

The size of the buffer used by *str_append in res, at line 144 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 1947 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1947	156
Object	argv	res

Code Snippet
File Name cosmopolitan/run-test262.c
Method int main(int argc, char **argv)

```
....
1947.  int main(int argc, char **argv)
```

File Name cosmopolitan/run-test262.c
Method char *str_append(char **pp, const char *sep, const char *str) {

```
....
156.      strcpy(res + len, str);
```

Buffer Overflow StrcpyStrcat\Path 13:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=50
Status	New

The size of the buffer used by *str_append in res, at line 144 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 1947 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1947	153
Object	argv	res

Code Snippet
File Name cosmopolitan/run-test262.c
Method int main(int argc, char **argv)

```
....
1947.  int main(int argc, char **argv)
```

File Name cosmopolitan/run-test262.c

Method	char *str_append(char **pp, const char *sep, const char *str) {

```
....
153.          strcpy(res, p);
```

Buffer Overflow StrcpyStrcat\Path 14:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=51
Status	New

The size of the buffer used by *str_append in res, at line 144 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *str_append passes to pp, at line 144 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	144	154
Object	pp	res

Code Snippet
File Name cosmopolitan/run-test262.c
Method char *str_append(char **pp, const char *sep, const char *str) {

```
....
144.  char *str_append(char **pp, const char *sep, const char *str) {
....
154.          strcat(res, sep);
```

Buffer Overflow StrcpyStrcat\Path 15:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=52
Status	New

The size of the buffer used by *str_append in res, at line 144 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *str_append passes to sep, at line 144 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	144	154
Object	sep	res

Code Snippet
File Name cosmopolitan/run-test262.c
Method char *str_append(char **pp, const char *sep, const char *str) {

```
....
144.  char *str_append(char **pp, const char *sep, const char *str) {
....
154.          strcat(res, sep);
```

Buffer Overflow StrcpyStrcat\Path 16:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=53
Status	New

The size of the buffer used by *str_append in res, at line 144 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 1947 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1947	154
Object	argv	res

Code Snippet

File Name cosmopolitan/run-test262.c

Method int main(int argc, char **argv)

```
....
1947.  int main(int argc, char **argv)
```

File Name cosmopolitan/run-test262.c

Method char *str_append(char **pp, const char *sep, const char *str) {

```
....
154.          strcat(res, sep);
```

Buffer Overflow StrcpyStrcat\Path 17:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=54
Status	New

The size of the buffer used by *str_append in sep, at line 144 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *str_append passes to sep, at line 144 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c

Line	144	154
Object	sep	sep

Code Snippet
File Name cosmopolitan/run-test262.c
Method char *str_append(char **pp, const char *sep, const char *str) {

```
....
144.  char *str_append(char **pp, const char *sep, const char *str) {
....
154.          strcat(res, sep);
```

Buffer Overflow OutOfBound

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

Categories

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

Description
Buffer Overflow OutOfBound\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=55
Status	New

The size of the buffer used by js_printf_internal in q, at line 137 of cosmopolitan/quickjs-libc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that js_printf_internal passes to q, at line 137 of cosmopolitan/quickjs-libc.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	274	274
Object	q	q

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_printf_internal(JSContext *ctx,

```
....
274.                      q[2] = q[-1];
```

Buffer Overflow OutOfBound\Path 2:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=500

	70&pathid=56
Status	New

The size of the buffer used by js_printf_internal in q, at line 137 of cosmopolitan/quickjs-libc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that js_printf_internal passes to q, at line 137 of cosmopolitan/quickjs-libc.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	275	275
Object	q	q

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_printf_internal(JSContext *ctx,

```
....
275.                        q[-1] = 'I';
```

Buffer Overflow OutOfBound\Path 3:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=57
Status	New

The size of the buffer used by *BF_crypt in i, at line 643 of cosmopolitan/crypt_blowfish.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that BF_set_key passes to tmp, at line 536 of cosmopolitan/crypt_blowfish.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/crypt_blowfish.c	cosmopolitan/crypt_blowfish.c
Line	541	706
Object	tmp	i

Code Snippet
File Name cosmopolitan/crypt_blowfish.c
Method static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

```
....
541.        BF_word safety, sign, diff, tmp[2];
```

File Name cosmopolitan/crypt_blowfish.c
Method static char *BF_crypt(const char *key, const char *setting,

```
....
706.                    data->ctx.s.P[i] ^= data->expanded_key[i];
```

Buffer Overflow OutOfBound\Path 4:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=58
Status	New

The size of the buffer used by pragmaVtabBestIndex in j, at line 2634 of cosmopolitan/pragma.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that pragmaVtabBestIndex passes to seen, at line 2634 of cosmopolitan/pragma.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/pragma.c	cosmopolitan/pragma.c
Line	2638	2651
Object	seen	j

Code Snippet
File Name cosmopolitan/pragma.c
Method static int pragmaVtabBestIndex(sqlite3_vtab *tab, sqlite3_index_info *pIdxInfo){

```
....
2638.    int seen[2];
....
2651.      seen[j] = i+1;
```

Buffer Overflow OutOfBound\Path 5:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=59
Status	New

The size of the buffer used by codes in symbol, at line 471 of cosmopolitan/puff.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that codes passes to lens, at line 471 of cosmopolitan/puff.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/puff.c	cosmopolitan/puff.c
Line	478	512
Object	lens	symbol

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

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

Buffer Overflow OutOfBound\Path 6:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=60
Status	New

The size of the buffer used by codes in symbol, at line 471 of cosmopolitan/puff.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that codes passes to lext, at line 471 of cosmopolitan/puff.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/puff.c	cosmopolitan/puff.c
Line	481	512
Object	lext	symbol

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

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

Buffer Overflow OutOfBound\Path 7:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=61
Status	New

The size of the buffer used by codes in symbol, at line 471 of cosmopolitan/puff.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that codes passes to dists, at line 471 of cosmopolitan/puff.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/puff.c	cosmopolitan/puff.c
Line	484	518
Object	dists	symbol

Code Snippet

File Name	cosmopolitan/puff.c
Method	local int codes(struct state *s,

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

Buffer Overflow OutOfBound\Path 8:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=62
Status	New

The size of the buffer used by codes in symbol, at line 471 of cosmopolitan/puff.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that codes passes to dext, at line 471 of cosmopolitan/puff.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/puff.c	cosmopolitan/puff.c
Line	488	518
Object	dext	symbol

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

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

Buffer Overflow LongString

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

Categories

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

Description

Buffer Overflow LongString\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=1
Status	New

The size of the buffer used by BF_set_key in tmp, at line 536 of cosmopolitan/crypt_blowfish.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *__crypt_blowfish passes to "8b \xd0\xc1\xd2\xcf\xcc\xd8", at line 788 of cosmopolitan/crypt_blowfish.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/crypt_blowfish.c	cosmopolitan/crypt_blowfish.c
Line	790	587
Object	"8b \xd0\xc1\xd2\xcf\xcc\xd8"	tmp

Code Snippet
File Name cosmopolitan/crypt_blowfish.c
Method char *__crypt_blowfish(const char *key, const char *setting, char *output)

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

File Name cosmopolitan/crypt_blowfish.c
Method static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

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

Buffer Overflow LongString\Path 2:

Severity High
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=2
Status New

The size of the buffer used by BF_set_key in tmp, at line 536 of cosmopolitan/crypt_blowfish.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *__crypt_blowfish passes to "8b \xd0\xc1\xd2\xcf\xcc\xd8", at line 788 of cosmopolitan/crypt_blowfish.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/crypt_blowfish.c	cosmopolitan/crypt_blowfish.c
Line	790	589
Object	"8b \xd0\xc1\xd2\xcf\xcc\xd8"	tmp

Code Snippet
File Name cosmopolitan/crypt_blowfish.c
Method char *__crypt_blowfish(const char *key, const char *setting, char *output)

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

File Name	cosmopolitan/crypt_blowfish.c
Method	static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

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

Buffer Overflow LongString\Path 3:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=3
Status	New

The size of the buffer used by BF_set_key in tmp, at line 536 of cosmopolitan/crypt_blowfish.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *__crypt_blowfish passes to "\xff\xa3", at line 788 of cosmopolitan/crypt_blowfish.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/crypt_blowfish.c	cosmopolitan/crypt_blowfish.c
Line	832	589
Object	"\xff\xa3"	tmp

Code Snippet

File Name	cosmopolitan/crypt_blowfish.c
Method	char *__crypt_blowfish(const char *key, const char *setting, char *output)

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

File Name	cosmopolitan/crypt_blowfish.c
Method	static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

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

Buffer Overflow LongString\Path 4:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=4
Status	New

The size of the buffer used by BF_set_key in tmp, at line 536 of cosmopolitan/crypt_blowfish.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that

*__crypt_blowfish passes to "\xff\xa3", at line 788 of cosmopolitan/crypt_blowfish.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/crypt_blowfish.c	cosmopolitan/crypt_blowfish.c
Line	832	587
Object	"\xff\xa3"	tmp

Code Snippet
File Name cosmopolitan/crypt_blowfish.c
Method char *__crypt_blowfish(const char *key, const char *setting, char *output)

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

File Name cosmopolitan/crypt_blowfish.c
Method static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

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

Buffer Overflow Indexes

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

Categories

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

Description
Buffer Overflow Indexes\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=5
Status	New

The size of the buffer used by load_config in strcspn, at line 919 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 1947 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1947	950
Object	argv	strcspn

Code Snippet

File Name	cosmopolitan/run-test262.c
Method	int main(int argc, char **argv)

```
....
1947.  int main(int argc, char **argv)
```

File Name	cosmopolitan/run-test262.c
Method	void load_config(const char *filename)

```
....
950.              p[strcspn(p, "]")] = '\0';
```

Buffer Overflow cpycat

Query Path:
CPP\Cx\CPP Buffer Overflow\Buffer Overflow cpycat 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

Buffer Overflow cpycat\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=6
Status	New

The size of the buffer used by *snocString in name, at line 1641 of cosmopolitan/bzip2.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 1688 of cosmopolitan/bzip2.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	1688	1646
Object	argv	name

Code Snippet

File Name	cosmopolitan/bzip2.c
Method	IntNative main (IntNative argc, Char *argv[])

```
....
1688.  IntNative main ( IntNative argc, Char *argv[] )
```

File Name	cosmopolitan/bzip2.c
Method	Cell *snocString (Cell *root, Char *name)

```
....
1646.        strcpy ( tmp->name, name );
```

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=1050080&projectid=50070&pathid=192
Status	New

The dangerous function, memcpy, was found in use at line 813 in cosmopolitan/compile.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	887	887
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/compile.c
Method compile_tr(char *p, struct s_tr **py)

```
....
887.                        memcpy(y->multis[i].from, op, oclen);
```

Dangerous Functions\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=193
Status	New

The dangerous function, memcpy, was found in use at line 813 in cosmopolitan/compile.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	889	889

Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/compile.c
Method compile_tr(char *p, struct s_tr **py)

```
....
889.                            memcpy(y->multis[i].to, np, nclen);
```

Dangerous Functions\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=194
Status	New

The dangerous function, memcpy, was found in use at line 788 in cosmopolitan/crypt_blowfish.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/crypt_blowfish.c	cosmopolitan/crypt_blowfish.c
Line	815	815
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/crypt_blowfish.c
Method char *__crypt_blowfish(const char *key, const char *setting, char *output)

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

Dangerous Functions\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=195
Status	New

The dangerous function, memcpy, was found in use at line 643 in cosmopolitan/crypt_blowfish.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/crypt_blowfish.c	cosmopolitan/crypt_blowfish.c
Line	679	679
Object	memcpy	memcpy

Code Snippet	
File Name	cosmopolitan/crypt_blowfish.c
Method	static char *BF_crypt(const char *key, const char *setting,

```
....
679.        memcpy(data->ctx.s.S, BF_init_state.s.S, sizeof(data-
>ctx.s.S));
```

Dangerous Functions\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=196
Status	New

The dangerous function, memcpy, was found in use at line 643 in cosmopolitan/crypt_blowfish.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/crypt_blowfish.c	cosmopolitan/crypt_blowfish.c
Line	755	755
Object	memcpy	memcpy

Code Snippet	
File Name	cosmopolitan/crypt_blowfish.c
Method	static char *BF_crypt(const char *key, const char *setting,

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

Dangerous Functions\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=197
Status	New

The dangerous function, memcpy, was found in use at line 173 in cosmopolitan/crypt_md5.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/crypt_md5.c	cosmopolitan/crypt_md5.c
Line	181	181
Object	memcpy	memcpy

Code Snippet	
File Name	cosmopolitan/crypt_md5.c

Method	static void md5_update(struct md5 *s, const void *m, unsigned long len)

```
....
181.                    memcpy(s->buf + r, p, len);
```

Dangerous Functions\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=198
Status	New

The dangerous function, memcpy, was found in use at line 173 in cosmopolitan/crypt_md5.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/crypt_md5.c	cosmopolitan/crypt_md5.c
Line	184	184
Object	memcpy	memcpy

Code Snippet	
File Name	cosmopolitan/crypt_md5.c
Method	static void md5_update(struct md5 *s, const void *m, unsigned long len)

```
....
184.                memcpy(s->buf + r, p, 64 - r);
```

Dangerous Functions\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=199
Status	New

The dangerous function, memcpy, was found in use at line 173 in cosmopolitan/crypt_md5.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/crypt_md5.c	cosmopolitan/crypt_md5.c
Line	191	191
Object	memcpy	memcpy

Code Snippet	
File Name	cosmopolitan/crypt_md5.c
Method	static void md5_update(struct md5 *s, const void *m, unsigned long len)

```
....
191.        memcpy(s->buf, p, len);
```

Dangerous Functions\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=200
Status	New

The dangerous function, memcpy, was found in use at line 236 in cosmopolitan/crypt_md5.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/crypt_md5.c	cosmopolitan/crypt_md5.c
Line	297	297
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/crypt_md5.c
Method static char *md5crypt(const char *key, const char *setting, char *output)

```
....
297.        memcpy(output, setting, 3 + slen);
```

Dangerous Functions\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=201
Status	New

The dangerous function, memcpy, was found in use at line 54 in cosmopolitan/djbsort_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	57	57
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test4) {

```
....
57.     a = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Dangerous Functions\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=202
Status	New

The dangerous function, memcpy, was found in use at line 54 in cosmopolitan/djbsort_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	58	58
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test4) {

```
....
58.     b = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Dangerous Functions\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=203
Status	New

The dangerous function, memcpy, was found in use at line 54 in cosmopolitan/djbsort_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	59	59
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test4) {

```
....
59.     c = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Dangerous Functions\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=204
Status	New

The dangerous function, memcpy, was found in use at line 68 in cosmopolitan/djbsort_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	85	85
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test64) {

```
....
85.     a = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Dangerous Functions\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=205
Status	New

The dangerous function, memcpy, was found in use at line 68 in cosmopolitan/djbsort_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	86	86
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test64) {

```
....
86.     b = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Dangerous Functions\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=206
Status	New

The dangerous function, memcpy, was found in use at line 68 in cosmopolitan/djbsort_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	87	87
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test64) {

```
....
87.     c = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Dangerous Functions\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=207
Status	New

The dangerous function, memcpy, was found in use at line 376 in cosmopolitan/fts3_tokenizer.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/fts3_tokenizer.c	cosmopolitan/fts3_tokenizer.c
Line	396	396
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/fts3_tokenizer.c
Method int queryTokenizer(

```
....
396.        memcpy((void *)pp, sqlite3_column_blob(pStmt, 0),
sizeof(*pp));
```

Dangerous Functions\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=208
Status	New

The dangerous function, memcpy, was found in use at line 192 in cosmopolitan/json.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/json.c	cosmopolitan/json.c
Line	202	202
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/json.c
Method static int jsonGrow(JsonString *p, u32 N){

```
....
202.      memcpy(zNew, p->zBuf, (size_t)p->nUsed);
```

Dangerous Functions\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=209
Status	New

The dangerous function, memcpy, was found in use at line 219 in cosmopolitan/json.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/json.c	cosmopolitan/json.c
Line	222	222
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/json.c
Method static void jsonAppendRaw(JsonString *p, const char *zIn, u32 N){

```
....
222.    memcpy(p->zBuf+p->nUsed, zIn, N);
```

Dangerous Functions\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=210
Status	New

The dangerous function, memcpy, was found in use at line 1013 in cosmopolitan/json.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/json.c	cosmopolitan/json.c
Line	1059	1059
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/json.c
Method static JsonParse *jsonParseCached(

```
....
1059.    memcpy((char*)p->zJson, zJson, nJson+1);
```

Dangerous Functions\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=211
Status	New

The dangerous function, memcpy, was found in use at line 2505 in cosmopolitan/json.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/json.c	cosmopolitan/json.c
Line	2524	2524
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/json.c
Method static int jsonEachFilter(

```
....
2524.    memcpy(p->zJson, z, (size_t)n+1);
```

Dangerous Functions\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=212
Status	New

The dangerous function, memcpy, was found in use at line 2505 in cosmopolitan/json.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/json.c	cosmopolitan/json.c
Line	2546	2546
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/json.c
Method static int jsonEachFilter(

```
....
2546.        memcpy(p->zRoot, zRoot, (size_t)n+1);
```

Dangerous Functions\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=213
Status	New

The dangerous function, memcpy, was found in use at line 168 in cosmopolitan/lstrlib.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/lstrlib.c	cosmopolitan/lstrlib.c
Line	182	182
Object	memcpy	memcpy

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

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

Dangerous Functions\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=214
Status	New

The dangerous function, memcpy, was found in use at line 168 in cosmopolitan/lstrlib.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/lstrlib.c	cosmopolitan/lstrlib.c
Line	184	184
Object	memcpy	memcpy

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

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

Dangerous Functions\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=215
Status	New

The dangerous function, memcpy, was found in use at line 168 in cosmopolitan/lstrlib.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/lstrlib.c	cosmopolitan/lstrlib.c
Line	188	188
Object	memcpy	memcpy

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

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

Dangerous Functions\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=216
Status	New

The dangerous function, memcpy, was found in use at line 1234 in cosmopolitan/lstrlib.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/lstrlib.c	cosmopolitan/lstrlib.c
Line	1249	1249
Object	memcpy	memcpy

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

```
....
1249.    memcpy(form, strfrmt, ((p - strfrmt) + 1) * sizeof(char));
```

Dangerous Functions\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=217
Status	New

The dangerous function, memcpy, was found in use at line 1574 in cosmopolitan/lstrlib.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/lstrlib.c	cosmopolitan/lstrlib.c
Line	1577	1577
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/lstrlib.c
Method static void copywithendian (char *dest, const char *src,

```
....
1577.       memcpy(dest, src, size);
```

Dangerous Functions\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=218
Status	New

The dangerous function, memcpy, was found in use at line 1147 in cosmopolitan/lvm.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/lvm.c	cosmopolitan/lvm.c
Line	1787	1787
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/lvm.c
Method void luaV_execute (lua_State *L, CallInfo *ci) {

```
....
1787.           memcpy(ra + 4, ra, 3 * sizeof(*ra));
```

Dangerous Functions\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=219
Status	New

The dangerous function, memcpy, was found in use at line 642 in cosmopolitan/lvm.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/lvm.c	cosmopolitan/lvm.c
Line	646	646
Object	memcpy	memcpy

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

```
....
646.      memcpy(buff + tl, svalue(s2v(top - n)), l * sizeof(char));
```

Dangerous Functions\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=220
Status	New

The dangerous function, memcpy, was found in use at line 201 in cosmopolitan/main.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/main.c	cosmopolitan/main.c
Line	349	349
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/main.c
Method int sqlite3_initialize(void){

```
....
349.      memcpy(&y, &x, 8);
```

Dangerous Functions\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=221
Status	New

The dangerous function, memcpy, was found in use at line 2908 in cosmopolitan/main.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/main.c	cosmopolitan/main.c
Line	3120	3120
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/main.c
Method int sqlite3ParseUri(

```
....
3120.        memcpy(zFile, zUri, nUri);
```

Dangerous Functions\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=222
Status	New

The dangerous function, memcpy, was found in use at line 3163 in cosmopolitan/main.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/main.c	cosmopolitan/main.c
Line	3251	3251
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/main.c
Method static int openDatabase(

```
....
3251.    memcpy(db->aLimit, aHardLimit, sizeof(db->aLimit));
```

Dangerous Functions\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=223
Status	New

The dangerous function, memcpy, was found in use at line 4499 in cosmopolitan/main.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/main.c	cosmopolitan/main.c
Line	4501	4501
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/main.c
Method static char *appendText(char *p, const char *z){

```
....
4501.    memcpy(p, z, n+1);
```

Dangerous Functions\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=224
Status	New

The dangerous function, memcpy, was found in use at line 657 in cosmopolitan/process.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/process.c	cosmopolitan/process.c
Line	674	674
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/process.c
Method regexec_e(regex_t *preg, const char *string, int eflags, int nomatch,

```
....
674.        (void)memcpy(buf, string, slen);
```

Dangerous Functions\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=225
Status	New

The dangerous function, memcpy, was found in use at line 2689 in cosmopolitan/quickjs-libc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	2725	2725
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static char **build_envp(JSContext *ctx, JSValueConst obj)

```
....
2725.          memcpy(pair, key, key_len);
```

Dangerous Functions\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=226
Status	New

The dangerous function, memcpy, was found in use at line 2689 in cosmopolitan/quickjs-libc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	2727	2727
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static char **build_envp(JSContext *ctx, JSValueConst obj)

```
....
2727.          memcpy(pair + key_len + 1, str, str_len);
```

Dangerous Functions\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=227
Status	New

The dangerous function, memcpy, was found in use at line 3343 in cosmopolitan/quickjs-libc.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	3372	3372
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_worker_postMessage(JSContext *ctx, JSValueConst this_val,

```
....
3372.       memcpy(msg->data, data, data_len);
```

Dangerous Functions\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=228
Status	New

The dangerous function, memcpy, was found in use at line 132 in cosmopolitan/run-test262.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	135	135
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/run-test262.c
Method char *strdup_len(const char *str, int len)

```
....
135.       memcpy(p, str, len);
```

Dangerous Functions\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=229
Status	New

The dangerous function, memcpy, was found in use at line 200 in cosmopolitan/run-test262.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	213	213
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/run-test262.c
Method char *compose_path(const char *path, const char *name)

```
....
213.              memcpy(q, path, path_len);
```

Dangerous Functions\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=230
Status	New

The dangerous function, memcpy, was found in use at line 200 in cosmopolitan/run-test262.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	217	217
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/run-test262.c
Method char *compose_path(const char *path, const char *name)

```
....
217.              memcpy(q, name, name_len + 1);
```

Dangerous Functions\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=231
Status	New

The dangerous function, memcpy, was found in use at line 1379 in cosmopolitan/run-test262.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1398	1398
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/run-test262.c
Method char *extract_desc(const char *buf, char style)

```
....
1398.               memcpy(desc, desc_start, len);
```

Dangerous Functions\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=232
Status	New

The dangerous function, memcpy, was found in use at line 111 in cosmopolitan/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/sds.c	cosmopolitan/sds.c
Line	163	163
Object	memcpy	memcpy

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

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

Dangerous Functions\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=233
Status	New

The dangerous function, memcpy, was found in use at line 225 in cosmopolitan/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/sds.c	cosmopolitan/sds.c
Line	262	262
Object	memcpy	memcpy

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

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

Dangerous Functions\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=234
Status	New

The dangerous function, memcpy, was found in use at line 278 in cosmopolitan/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/sds.c	cosmopolitan/sds.c
Line	294	294
Object	memcpy	memcpy

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

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

Dangerous Functions\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=235
Status	New

The dangerous function, memcpy, was found in use at line 409 in cosmopolitan/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/sds.c	cosmopolitan/sds.c
Line	414	414
Object	memcpy	memcpy

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

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

Dangerous Functions\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=236
Status	New

The dangerous function, memcpy, was found in use at line 438 in cosmopolitan/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/sds.c	cosmopolitan/sds.c
Line	443	443
Object	memcpy	memcpy

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

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

Dangerous Functions\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=237
Status	New

The dangerous function, memcpy, was found in use at line 612 in cosmopolitan/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/sds.c	cosmopolitan/sds.c
Line	644	644
Object	memcpy	memcpy

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

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

Dangerous Functions\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=238
Status	New

The dangerous function, memcpy, was found in use at line 612 in cosmopolitan/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/sds.c	cosmopolitan/sds.c
Line	661	661
Object	memcpy	memcpy

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

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

Dangerous Functions\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=239
Status	New

The dangerous function, memcpy, was found in use at line 612 in cosmopolitan/sds.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/sds.c	cosmopolitan/sds.c
Line	679	679
Object	memcpy	memcpy

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

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

Dangerous Functions\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=240
Status	New

The dangerous function, memcpy, was found in use at line 214 in cosmopolitan/sqlite3expert.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/sqlite3expert.c	cosmopolitan/sqlite3expert.c
Line	233	233
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/sqlite3expert.c
Method static int idxHashAdd(

```
....
233.      memcpy(pEntry->zKey, zKey, nKey);
```

Dangerous Functions\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=241
Status	New

The dangerous function, memcpy, was found in use at line 214 in cosmopolitan/sqlite3expert.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	cosmopolitan/sqlite3expert.c	cosmopolitan/sqlite3expert.c
Line	236	236
Object	memcpy	memcpy

Code Snippet
File Name cosmopolitan/sqlite3expert.c
Method static int idxHashAdd(

```
....
236.        memcpy(pEntry->zVal, zVal, nVal);
```

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=1050080&projectid=50070&pathid=65
Status	New

The size of the buffer used by *__crypt_blowfish in Namespace294156478, at line 788 of cosmopolitan/crypt_blowfish.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *__crypt_blowfish passes to Namespace294156478, at line 788 of cosmopolitan/crypt_blowfish.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/crypt_blowfish.c	cosmopolitan/crypt_blowfish.c
Line	815	815
Object	Namespace294156478	Namespace294156478

Code Snippet
File Name cosmopolitan/crypt_blowfish.c
Method char *__crypt_blowfish(const char *key, const char *setting, char *output)

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

Buffer Overflow boundcpy WrongSizeParam\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=66
Status	New

The size of the buffer used by *BF_crypt in Namespace294156478, at line 643 of cosmopolitan/crypt_blowfish.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *BF_crypt passes to Namespace294156478, at line 643 of cosmopolitan/crypt_blowfish.c, to overwrite the target buffer.

	Source	Destination

File	cosmopolitan/crypt_blowfish.c	cosmopolitan/crypt_blowfish.c
Line	679	679
Object	Namespace294156478	Namespace294156478

Code Snippet
File Name cosmopolitan/crypt_blowfish.c
Method static char *BF_crypt(const char *key, const char *setting,

```
....
679.        memcpy(data->ctx.s.S, BF_init_state.s.S, sizeof(data-
>ctx.s.S));
```

Buffer Overflow boundcpy WrongSizeParam\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=67
Status	New

The size of the buffer used by queryTokenizer in pp, at line 376 of cosmopolitan/fts3_tokenizer.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that queryTokenizer passes to pp, at line 376 of cosmopolitan/fts3_tokenizer.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/fts3_tokenizer.c	cosmopolitan/fts3_tokenizer.c
Line	396	396
Object	pp	pp

Code Snippet
File Name cosmopolitan/fts3_tokenizer.c
Method int queryTokenizer(

```
....
396.        memcpy((void *)pp, sqlite3_column_blob(pStmt, 0),
sizeof(*pp));
```

Buffer Overflow boundcpy WrongSizeParam\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=68
Status	New

The size of the buffer used by openDatabase in ->, at line 3163 of cosmopolitan/main.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that openDatabase passes to ->, at line 3163 of cosmopolitan/main.c, to overwrite the target buffer.

	Source	Destination

File	cosmopolitan/main.c	cosmopolitan/main.c
Line	3251	3251
Object	->	->

Code Snippet
File Name cosmopolitan/main.c
Method static int openDatabase(

```
....
3251.    memcpy(db->aLimit, aHardLimit, sizeof(db->aLimit));
```

Buffer Overflow boundcpy WrongSizeParam\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=69
Status	New

The size of the buffer used by sqlite3rbu_create_vfs in sqlite3_vfs, at line 5281 of cosmopolitan/sqlite3rbu.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sqlite3rbu_create_vfs passes to sqlite3_vfs, at line 5281 of cosmopolitan/sqlite3rbu.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	5331	5331
Object	sqlite3_vfs	sqlite3_vfs

Code Snippet
File Name cosmopolitan/sqlite3rbu.c
Method int sqlite3rbu_create_vfs(const char *zName, const char *zParent){

```
....
5331.        memcpy(&pNew->base, &vfs_template, sizeof(sqlite3_vfs));
```

Buffer Overflow boundcpy WrongSizeParam\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=70
Status	New

The size of the buffer used by ssl_parse_session_ticket_ext in mbedtls_ssl_session, at line 633 of cosmopolitan/ssl_srv.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ssl_parse_session_ticket_ext passes to mbedtls_ssl_session, at line 633 of cosmopolitan/ssl_srv.c, to overwrite the target buffer.

	Source	Destination

File	cosmopolitan/ssl_srv.c	cosmopolitan/ssl_srv.c
Line	690	690
Object	mbedtls_ssl_session	mbedtls_ssl_session

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

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

Buffer Overflow boundcpy WrongSizeParam\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=71
Status	New

The size of the buffer used by *__crypt_blowfish in Namespace294156478, at line 788 of cosmopolitan/crypt_blowfish.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *__crypt_blowfish passes to Namespace294156478, at line 788 of cosmopolitan/crypt_blowfish.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/crypt_blowfish.c	cosmopolitan/crypt_blowfish.c
Line	821	821
Object	Namespace294156478	Namespace294156478

Code Snippet
File Name cosmopolitan/crypt_blowfish.c
Method char *__crypt_blowfish(const char *key, const char *setting, char *output)

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

Buffer Overflow boundcpy WrongSizeParam\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=72
Status	New

The size of the buffer used by sqlite3_config in Namespace1573645336, at line 430 of cosmopolitan/main.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sqlite3_config passes to Namespace1573645336, at line 430 of cosmopolitan/main.c, to overwrite the target buffer.

	Source	Destination

File	cosmopolitan/main.c	cosmopolitan/main.c
Line	596	596
Object	Namespace1573645336	Namespace1573645336

Code Snippet
File Name cosmopolitan/main.c
Method int sqlite3_config(int op, ...){

```
....
596.          memset(&sqlite3GlobalConfig.m, 0,
sizeof(sqlite3GlobalConfig.m));
```

Buffer Overflow boundcpy WrongSizeParam\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=73
Status	New

The size of the buffer used by pragmaVtabConnect in PragmaVtab, at line 2559 of cosmopolitan/pragma.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that pragmaVtabConnect passes to PragmaVtab, at line 2559 of cosmopolitan/pragma.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/pragma.c	cosmopolitan/pragma.c
Line	2604	2604
Object	PragmaVtab	PragmaVtab

Code Snippet
File Name cosmopolitan/pragma.c
Method static int pragmaVtabConnect(

```
....
2604.        memset(pTab, 0, sizeof(PragmaVtab));
```

Buffer Overflow boundcpy WrongSizeParam\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=74
Status	New

The size of the buffer used by pragmaVtabOpen in PragmaVtabCursor, at line 2671 of cosmopolitan/pragma.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that pragmaVtabOpen passes to PragmaVtabCursor, at line 2671 of cosmopolitan/pragma.c, to overwrite the target buffer.

	Source	Destination

File	cosmopolitan/pragma.c	cosmopolitan/pragma.c
Line	2675	2675
Object	PragmaVtabCursor	PragmaVtabCursor

Code Snippet
File Name cosmopolitan/pragma.c
Method static int pragmaVtabOpen(sqlite3_vtab *pVtab, sqlite3_vtab_cursor **ppCursor){

```
....
2675.    memset(pCsr, 0, sizeof(PragmaVtabCursor));
```

Buffer Overflow boundcpy WrongSizeParam\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=75
Status	New

The size of the buffer used by idxHashInit in IdxHash, at line 175 of cosmopolitan/sqlite3expert.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that idxHashInit passes to IdxHash, at line 175 of cosmopolitan/sqlite3expert.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/sqlite3expert.c	cosmopolitan/sqlite3expert.c
Line	176	176
Object	IdxHash	IdxHash

Code Snippet
File Name cosmopolitan/sqlite3expert.c
Method static void idxHashInit(IdxHash *pHash){

```
....
176.    memset(pHash, 0, sizeof(IdxHash));
```

Buffer Overflow boundcpy WrongSizeParam\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=76
Status	New

The size of the buffer used by idxHashClear in IdxHash, at line 182 of cosmopolitan/sqlite3expert.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that idxHashClear passes to IdxHash, at line 182 of cosmopolitan/sqlite3expert.c, to overwrite the target buffer.

	Source	Destination

File	cosmopolitan/sqlite3expert.c	cosmopolitan/sqlite3expert.c
Line	193	193
Object	IdxHash	IdxHash

Code Snippet
File Name cosmopolitan/sqlite3expert.c
Method static void idxHashClear(IdxHash *pHash){

```
....
193.    memset(pHash, 0, sizeof(IdxHash));
```

Buffer Overflow boundcpy WrongSizeParam\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=77
Status	New

The size of the buffer used by rbuObjIterFinalize in RbuObjIter, at line 849 of cosmopolitan/sqlite3rbu.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that rbuObjIterFinalize passes to RbuObjIter, at line 849 of cosmopolitan/sqlite3rbu.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	854	854
Object	RbuObjIter	RbuObjIter

Code Snippet
File Name cosmopolitan/sqlite3rbu.c
Method static void rbuObjIterFinalize(RbuObjIter *pIter){

```
....
854.    memset(pIter, 0, sizeof(RbuObjIter));
```

Buffer Overflow boundcpy WrongSizeParam\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=78
Status	New

The size of the buffer used by rbuObjIterFirst in RbuObjIter, at line 981 of cosmopolitan/sqlite3rbu.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that rbuObjIterFirst passes to RbuObjIter, at line 981 of cosmopolitan/sqlite3rbu.c, to overwrite the target buffer.

	Source	Destination

File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	983	983
Object	RbuObjIter	RbuObjIter

Code Snippet
File Name cosmopolitan/sqlite3rbu.c
Method static int rbuObjIterFirst(sqlite3rbu *p, RbuObjIter *pIter){

```
....
983.    memset(pIter, 0, sizeof(RbuObjIter));
```

Buffer Overflow boundcpy WrongSizeParam\Path 15:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=79
Status New

The size of the buffer used by *openRbuHandle in sqlite3rbu, at line 3970 of cosmopolitan/sqlite3rbu.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *openRbuHandle passes to sqlite3rbu, at line 3970 of cosmopolitan/sqlite3rbu.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	3985	3985
Object	sqlite3rbu	sqlite3rbu

Code Snippet
File Name cosmopolitan/sqlite3rbu.c
Method static sqlite3rbu *openRbuHandle(

```
....
3985.      memset(p, 0, sizeof(sqlite3rbu));
```

Buffer Overflow boundcpy WrongSizeParam\Path 16:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=80
Status New

The size of the buffer used by *rbuMisuseError in sqlite3rbu, at line 4135 of cosmopolitan/sqlite3rbu.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *rbuMisuseError passes to sqlite3rbu, at line 4135 of cosmopolitan/sqlite3rbu.c, to overwrite the target buffer.

	Source	Destination

File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	4139	4139
Object	sqlite3rbu	sqlite3rbu

Code Snippet
File Name cosmopolitan/sqlite3rbu.c
Method static sqlite3rbu *rbuMisuseError(void){

```
....
4139.       memset(pRet, 0, sizeof(sqlite3rbu));
```

Buffer Overflow boundcpy WrongSizeParam\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=81
Status	New

The size of the buffer used by rbuVfsOpen in rbu_file, at line 5025 of cosmopolitan/sqlite3rbu.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that rbuVfsOpen passes to rbu_file, at line 5025 of cosmopolitan/sqlite3rbu.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	5059	5059
Object	rbu_file	rbu_file

Code Snippet
File Name cosmopolitan/sqlite3rbu.c
Method static int rbuVfsOpen(

```
....
5059.     memset(pFd, 0, sizeof(rbu_file));
```

Buffer Overflow boundcpy WrongSizeParam\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=82
Status	New

The size of the buffer used by ssl_parse_client_hello_v2 in ->, at line 1163 of cosmopolitan/ssl_srv.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ssl_parse_client_hello_v2 passes to ->, at line 1163 of cosmopolitan/ssl_srv.c, to overwrite the target buffer.

	Source	Destination

File	cosmopolitan/ssl_srv.c	cosmopolitan/ssl_srv.c
Line	1305	1305
Object	->	->

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

```
....
1305.              sizeof( ssl->session_negotiate->id ) );
```

Buffer Overflow boundcpy WrongSizeParam\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=83
Status	New

The size of the buffer used by vdbePmaReaderClear in PmaReader, at line 469 of cosmopolitan/vdbesort.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that vdbePmaReaderClear passes to PmaReader, at line 469 of cosmopolitan/vdbesort.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/vdbesort.c	cosmopolitan/vdbesort.c
Line	474	474
Object	PmaReader	PmaReader

Code Snippet
File Name cosmopolitan/vdbesort.c
Method static void vdbePmaReaderClear(PmaReader *pReadr){

```
....
474.    memset(pReadr, 0, sizeof(PmaReader));
```

Buffer Overflow boundcpy WrongSizeParam\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=84
Status	New

The size of the buffer used by vdbeSortSubtaskCleanup in SortSubtask, at line 1047 of cosmopolitan/vdbesort.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that vdbeSortSubtaskCleanup passes to SortSubtask, at line 1047 of cosmopolitan/vdbesort.c, to overwrite the target buffer.

	Source	Destination

File	cosmopolitan/vdbesort.c	cosmopolitan/vdbesort.c
Line	1066	1066
Object	SortSubtask	SortSubtask

Code Snippet
File Name cosmopolitan/vdbesort.c
Method static void vdbeSortSubtaskCleanup(sqlite3 *db, SortSubtask *pTask){

```
....
1066.    memset(pTask, 0, sizeof(SortSubtask));
```

Buffer Overflow boundcpy WrongSizeParam\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=85
Status	New

The size of the buffer used by vdbePmaWriterInit in PmaWriter, at line 1453 of cosmopolitan/vdbesort.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that vdbePmaWriterInit passes to PmaWriter, at line 1453 of cosmopolitan/vdbesort.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/vdbesort.c	cosmopolitan/vdbesort.c
Line	1459	1459
Object	PmaWriter	PmaWriter

Code Snippet
File Name cosmopolitan/vdbesort.c
Method static void vdbePmaWriterInit(

```
....
1459.    memset(p, 0, sizeof(PmaWriter));
```

Buffer Overflow boundcpy WrongSizeParam\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=86
Status	New

The size of the buffer used by vdbePmaWriterFinish in PmaWriter, at line 1508 of cosmopolitan/vdbesort.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that vdbePmaWriterFinish passes to PmaWriter, at line 1508 of cosmopolitan/vdbesort.c, to overwrite the target buffer.

	Source	Destination

File	cosmopolitan/vdbesort.c	cosmopolitan/vdbesort.c
Line	1519	1519
Object	PmaWriter	PmaWriter

Code Snippet
File Name cosmopolitan/vdbesort.c
Method static int vdbePmaWriterFinish(PmaWriter *p, i64 *piEof){

```
....
1519.    memset(p, 0, sizeof(PmaWriter));
```

Buffer Overflow boundcpy WrongSizeParam\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=87
Status	New

The size of the buffer used by vdbeSorterListToPMA in PmaWriter, at line 1548 of cosmopolitan/vdbesort.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that vdbeSorterListToPMA passes to PmaWriter, at line 1548 of cosmopolitan/vdbesort.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/vdbesort.c	cosmopolitan/vdbesort.c
Line	1560	1560
Object	PmaWriter	PmaWriter

Code Snippet
File Name cosmopolitan/vdbesort.c
Method static int vdbeSorterListToPMA(SortSubtask *pTask, SorterList *pList){

```
....
1560.    memset(&writer, 0, sizeof(PmaWriter));
```

Buffer Overflow boundcpy WrongSizeParam\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=88
Status	New

The size of the buffer used by str_rep in l, at line 168 of cosmopolitan/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 168 of cosmopolitan/lstrlib.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/lstrlib.c	cosmopolitan/lstrlib.c

Line	182	182
Object	l	l

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

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

Buffer Overflow boundcpy WrongSizeParam\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=89
Status	New

The size of the buffer used by str_rep in char, at line 168 of cosmopolitan/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 168 of cosmopolitan/lstrlib.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/lstrlib.c	cosmopolitan/lstrlib.c
Line	182	182
Object	char	char

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

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

Buffer Overflow boundcpy WrongSizeParam\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=90
Status	New

The size of the buffer used by str_rep in lsep, at line 168 of cosmopolitan/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 168 of cosmopolitan/lstrlib.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/lstrlib.c	cosmopolitan/lstrlib.c
Line	184	184

Object	lsep	lsep

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

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

Buffer Overflow boundcpy WrongSizeParam\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=91
Status	New

The size of the buffer used by str_rep in char, at line 168 of cosmopolitan/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 168 of cosmopolitan/lstrlib.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/lstrlib.c	cosmopolitan/lstrlib.c
Line	184	184
Object	char	char

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

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

Buffer Overflow boundcpy WrongSizeParam\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=92
Status	New

The size of the buffer used by str_rep in l, at line 168 of cosmopolitan/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 168 of cosmopolitan/lstrlib.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/lstrlib.c	cosmopolitan/lstrlib.c
Line	188	188
Object	l	l

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

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

Buffer Overflow boundcpy WrongSizeParam\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=93
Status	New

The size of the buffer used by str_rep in char, at line 168 of cosmopolitan/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 168 of cosmopolitan/lstrlib.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/lstrlib.c	cosmopolitan/lstrlib.c
Line	188	188
Object	char	char

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

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

Buffer Overflow boundcpy WrongSizeParam\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=94
Status	New

The size of the buffer used by *scanformat in char, at line 1234 of cosmopolitan/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 1234 of cosmopolitan/lstrlib.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/lstrlib.c	cosmopolitan/lstrlib.c
Line	1249	1249
Object	char	char

Code Snippet

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

```
....
1249.    memcpy(form, strfrmt, ((p - strfrmt) + 1) * sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=95
Status	New

The size of the buffer used by luaV_execute in ra, at line 1147 of cosmopolitan/lvm.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that luaV_execute passes to ra, at line 1147 of cosmopolitan/lvm.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/lvm.c	cosmopolitan/lvm.c
Line	1787	1787
Object	ra	ra

Code Snippet	
File Name	cosmopolitan/lvm.c
Method	void luaV_execute (lua_State *L, CallInfo *ci) {

```
....
1787.          memcpy(ra + 4, ra, 3 * sizeof(*ra));
```

Buffer Overflow boundcpy WrongSizeParam\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=96
Status	New

The size of the buffer used by copy2buff in l, at line 642 of cosmopolitan/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 642 of cosmopolitan/lvm.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/lvm.c	cosmopolitan/lvm.c
Line	646	646
Object	l	l

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

```
....
646.      memcpy(buff + tl, svalue(s2v(top - n)), l * sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=97
Status	New

The size of the buffer used by copy2buff in char, at line 642 of cosmopolitan/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 642 of cosmopolitan/lvm.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/lvm.c	cosmopolitan/lvm.c
Line	646	646
Object	char	char

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

```
....
646.      memcpy(buff + tl, svalue(s2v(top - n)), l * sizeof(char));
```

Buffer Overflow boundcpy WrongSizeParam\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=98
Status	New

The size of the buffer used by rbuObjIterCacheIndexedCols in pIter, at line 1272 of cosmopolitan/sqlite3rbu.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that rbuObjIterCacheIndexedCols passes to pIter, at line 1272 of cosmopolitan/sqlite3rbu.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	1277	1277
Object	pIter	pIter

Code Snippet
File Name cosmopolitan/sqlite3rbu.c
Method static void rbuObjIterCacheIndexedCols(sqlite3rbu *p, RbuObjIter *pIter){

```
....
1277.       memcpy(pIter->abIndexed, pIter->abTblPk, sizeof(u8)*pIter-
>nTblCol);
```

Buffer Overflow boundcpy WrongSizeParam\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=99
Status	New

The size of the buffer used by rbuObjIterCacheIndexedCols in u8, at line 1272 of cosmopolitan/sqlite3rbu.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that rbuObjIterCacheIndexedCols passes to u8, at line 1272 of cosmopolitan/sqlite3rbu.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	1277	1277
Object	u8	u8

Code Snippet
File Name cosmopolitan/sqlite3rbu.c
Method static void rbuObjIterCacheIndexedCols(sqlite3rbu *p, RbuObjIter *pIter){

```
....
1277.       memcpy(pIter->abIndexed, pIter->abTblPk, sizeof(u8)*pIter-
>nTblCol);
```

Buffer Overflow boundcpy WrongSizeParam\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=100
Status	New

The size of the buffer used by rbuObjIterCacheIndexedCols in pIter, at line 1272 of cosmopolitan/sqlite3rbu.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that rbuObjIterCacheIndexedCols passes to pIter, at line 1272 of cosmopolitan/sqlite3rbu.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	1290	1290
Object	pIter	pIter

Code Snippet
File Name cosmopolitan/sqlite3rbu.c

Method	static void rbuObjIterCacheIndexedCols(sqlite3rbu *p, RbuObjIter *pIter){

```
....
1290.         memset(pIter->abIndexed, 0x01, sizeof(u8)*pIter->nTblCol);
```

Buffer Overflow boundcpy WrongSizeParam\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=101
Status	New

The size of the buffer used by rbuObjIterCacheIndexedCols in u8, at line 1272 of cosmopolitan/sqlite3rbu.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that rbuObjIterCacheIndexedCols passes to u8, at line 1272 of cosmopolitan/sqlite3rbu.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	1290	1290
Object	u8	u8

Code Snippet	
File Name	cosmopolitan/sqlite3rbu.c
Method	static void rbuObjIterCacheIndexedCols(sqlite3rbu *p, RbuObjIter *pIter){

```
....
1290.         memset(pIter->abIndexed, 0x01, sizeof(u8)*pIter->nTblCol);
```

Buffer Overflow boundcpy WrongSizeParam\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=102
Status	New

The size of the buffer used by rbuObjIterCacheIndexedCols in pIter, at line 1272 of cosmopolitan/sqlite3rbu.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that rbuObjIterCacheIndexedCols passes to pIter, at line 1272 of cosmopolitan/sqlite3rbu.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	1299	1299
Object	pIter	pIter

Code Snippet	
File Name	cosmopolitan/sqlite3rbu.c
Method	static void rbuObjIterCacheIndexedCols(sqlite3rbu *p, RbuObjIter *pIter){

```
....
1299.           memset(pIter->abIndexed, 0x01, sizeof(u8)*pIter-
>nTblCol);
```

Buffer Overflow boundcpy WrongSizeParam\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=103
Status	New

The size of the buffer used by rbuObjIterCacheIndexedCols in u8, at line 1272 of cosmopolitan/sqlite3rbu.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that rbuObjIterCacheIndexedCols passes to u8, at line 1272 of cosmopolitan/sqlite3rbu.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	1299	1299
Object	u8	u8

Code Snippet
File Name cosmopolitan/sqlite3rbu.c
Method static void rbuObjIterCacheIndexedCols(sqlite3rbu *p, RbuObjIter *pIter){

```
....
1299.           memset(pIter->abIndexed, 0x01, sizeof(u8)*pIter-
>nTblCol);
```

Buffer Overflow boundcpy WrongSizeParam\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=104
Status	New

The size of the buffer used by rbuWinUtf8ToUnicode in nChar, at line 3175 of cosmopolitan/sqlite3rbu.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that rbuWinUtf8ToUnicode passes to nChar, at line 3175 of cosmopolitan/sqlite3rbu.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	3187	3187
Object	nChar	nChar

Code Snippet
File Name cosmopolitan/sqlite3rbu.c

Method	static LPWSTR rbuWinUtf8ToUnicode(const char *zFilename){

```
....
3187.    memset(zWideFilename, 0, nChar*sizeof(zWideFilename[0]));
```

Buffer Overflow boundcpy WrongSizeParam\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=105
Status	New

The size of the buffer used by rbuWinUtf8ToUnicode in zWideFilename, at line 3175 of cosmopolitan/sqlite3rbu.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that rbuWinUtf8ToUnicode passes to zWideFilename, at line 3175 of cosmopolitan/sqlite3rbu.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	3187	3187
Object	zWideFilename	zWideFilename

Code Snippet	
File Name	cosmopolitan/sqlite3rbu.c
Method	static LPWSTR rbuWinUtf8ToUnicode(const char *zFilename){

```
....
3187.    memset(zWideFilename, 0, nChar*sizeof(zWideFilename[0]));
```

Buffer Overflow boundcpy WrongSizeParam\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=106
Status	New

The size of the buffer used by rbuVfsShmMap in char, at line 4940 of cosmopolitan/sqlite3rbu.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that rbuVfsShmMap passes to char, at line 4940 of cosmopolitan/sqlite3rbu.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	4967	4967
Object	char	char

Code Snippet	
File Name	cosmopolitan/sqlite3rbu.c
Method	static int rbuVfsShmMap(

```
....
4967.        memset(&apNew[p->nShm], 0, sizeof(char*) * (1 + iRegion -
p->nShm));
```

Buffer Overflow boundcpy WrongSizeParam\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=107
Status	New

The size of the buffer used by jsonAppendRaw in N, at line 219 of cosmopolitan/json.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that jsonAppendRaw passes to N, at line 219 of cosmopolitan/json.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/json.c	cosmopolitan/json.c
Line	222	222
Object	N	N

Code Snippet
File Name cosmopolitan/json.c
Method static void jsonAppendRaw(JsonString *p, const char *zIn, u32 N){

```
....
222.    memcpy(p->zBuf+p->nUsed, zIn, N);
```

Buffer Overflow boundcpy WrongSizeParam\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=108
Status	New

The size of the buffer used by **build_envp in key_len, at line 2689 of cosmopolitan/quickjs-libc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that **build_envp passes to key_len, at line 2689 of cosmopolitan/quickjs-libc.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	2725	2725
Object	key_len	key_len

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static char **build_envp(JSContext *ctx, JSValueConst obj)

```
....
2725.          memcpy(pair, key, key_len);
```

Buffer Overflow boundcpy WrongSizeParam\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=109
Status	New

The size of the buffer used by **build_envp in str_len, at line 2689 of cosmopolitan/quickjs-libc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that **build_envp passes to str_len, at line 2689 of cosmopolitan/quickjs-libc.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	2727	2727
Object	str_len	str_len

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static char **build_envp(JSContext *ctx, JSValueConst obj)

```
....
2727.          memcpy(pair + key_len + 1, str, str_len);
```

Buffer Overflow boundcpy WrongSizeParam\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=110
Status	New

The size of the buffer used by js_worker_postMessage in data_len, at line 3343 of cosmopolitan/quickjs-libc.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that js_worker_postMessage passes to data_len, at line 3343 of cosmopolitan/quickjs-libc.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	3372	3372
Object	data_len	data_len

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_worker_postMessage(JSContext *ctx, JSValueConst this_val,

```
....
3372.       memcpy(msg->data, data, data_len);
```

Buffer Overflow boundcpy WrongSizeParam\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=111
Status	New

The size of the buffer used by sdscatlen in len, at line 409 of cosmopolitan/sds.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sdscatlen passes to len, at line 409 of cosmopolitan/sds.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/sds.c	cosmopolitan/sds.c
Line	414	414
Object	len	len

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

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

Buffer Overflow boundcpy WrongSizeParam\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=112
Status	New

The size of the buffer used by sdscpylen in len, at line 438 of cosmopolitan/sds.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sdscpylen passes to len, at line 438 of cosmopolitan/sds.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/sds.c	cosmopolitan/sds.c
Line	443	443
Object	len	len

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

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

Buffer Overflow boundcpy WrongSizeParam\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=113
Status	New

The size of the buffer used by sdscatfmt in l, at line 612 of cosmopolitan/sds.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sdscatfmt passes to l, at line 612 of cosmopolitan/sds.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/sds.c	cosmopolitan/sds.c
Line	644	644
Object	l	l

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

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

Buffer Overflow boundcpy WrongSizeParam\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=114
Status	New

The size of the buffer used by sdscatfmt in l, at line 612 of cosmopolitan/sds.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sdscatfmt passes to l, at line 612 of cosmopolitan/sds.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/sds.c	cosmopolitan/sds.c
Line	661	661
Object	l	l

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

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

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=1050080&projectid=50070&pathid=416
Status	New

	Source	Destination
File	cosmopolitan/process.c	cosmopolitan/process.c
Line	130	130
Object	neW	neW

Code Snippet
File Name cosmopolitan/process.c
Method process(void)

```
....
130.                    goto new;
```

Memory Leak\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=417
Status	New

	Source	Destination
File	cosmopolitan/process.c	cosmopolitan/process.c
Line	133	133
Object	neW	neW

Code Snippet
File Name cosmopolitan/process.c
Method process(void)

```
....
133.                            goto new;
```

Memory Leak\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=418
Status	New

	Source	Destination
File	cosmopolitan/process.c	cosmopolitan/process.c
Line	140	140
Object	neW	neW

Code Snippet
File Name cosmopolitan/process.c
Method process(void)

```
....
140.                                goto new;
```

Memory Leak\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=419
Status	New

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	244	244
Object	ibuf	ibuf

Code Snippet
File Name cosmopolitan/bzip2.c
Method void compressStream (FILE *stream, FILE *zStream)

```
....
244.     UChar   *ibuf = gc(malloc(5000));
```

Memory Leak\Path 5:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=420
Status	New

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	349	349
Object	obuf	obuf

Code Snippet
File Name cosmopolitan/bzip2.c
Method Bool uncompressStream (FILE *zStream, FILE *stream)

```
....
349.     UChar    *obuf = gc(malloc(5000));
```

Memory Leak\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=421
Status	New

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	350	350
Object	unused	unused

Code Snippet
File Name cosmopolitan/bzip2.c
Method Bool uncompressStream (FILE *zStream, FILE *stream)

```
....
350.     UChar    *unused = gc(malloc(BZ_MAX_UNUSED));
```

Memory Leak\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=422
Status	New

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	470	470

Object	obuf	obuf

Code Snippet
File Name cosmopolitan/bzip2.c
Method Bool testStream (FILE *zStream)

```
....
470.     UChar   *obuf = gc(malloc(5000));
```

Memory Leak\Path 8:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=423
Status New

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	471	471
Object	unused	unused

Code Snippet
File Name cosmopolitan/bzip2.c
Method Bool testStream (FILE *zStream)

```
....
471.     UChar   *unused = gc(malloc(BZ_MAX_UNUSED));
```

Memory Leak\Path 9:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=424
Status New

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	107	107
Object	v	v

Code Snippet
File Name cosmopolitan/bzlib.c
Method void* default_bzalloc (void* opaque, Int32 items, Int32 size)

```
....
107.     void* v = malloc ( items * size );
```

Memory Leak\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=425
Status	New

	Source	Destination
File	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	184	184
Object	re	re

Code Snippet
File Name cosmopolitan/compile.c
Method compile_stream(struct s_command **link)

```
....
184.         char *re = gc(malloc(_POSIX2_LINE_MAX + 1));
```

Memory Leak\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=426
Status	New

	Source	Destination
File	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	818	818
Object	old	old

Code Snippet
File Name cosmopolitan/compile.c
Method compile_tr(char *p, struct s_tr **py)

```
....
818.         char *old = gc(malloc(_POSIX2_LINE_MAX + 1));
```

Memory Leak\Path 12:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=427
Status	New

	Source	Destination
File	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	819	819
Object	neW	neW

Code Snippet
File Name cosmopolitan/compile.c
Method compile_tr(char *p, struct s_tr **py)

```
....
819.        char *new = gc(malloc(_POSIX2_LINE_MAX + 1));
```

Memory Leak\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=428
Status	New

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	286	286
Object	new_str	new_str

Code Snippet
File Name cosmopolitan/getopt.c
Method exchange (char **argv)

```
....
286.        char *new_str = malloc (top + 1);
```

Memory Leak\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=429
Status	New

	Source	Destination
File	cosmopolitan/process.c	cosmopolitan/process.c
Line	541	541

Object	buf	buf

Code Snippet
File Name cosmopolitan/process.c
Method flush_appends(void)

```
....
541.        char *buf = gc(malloc(8 * 1024));
```

Memory Leak\Path 15:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=430
Status New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	134	134
Object	p	p

Code Snippet
File Name cosmopolitan/run-test262.c
Method char *strdup_len(const char *str, int len)

```
....
134.      char *p = malloc(len + 1);
```

Memory Leak\Path 16:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=431
Status New

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	1620	1620
Object	p	p

Code Snippet
File Name cosmopolitan/bzip2.c
Method void *myMalloc (Int32 n)

```
....
1620.     p = malloc ( (size_t)n );
```

Memory Leak\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=432
Status	New

	Source	Destination
File	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	794	794
Object	wfile	wfile

Code Snippet
File Name cosmopolitan/compile.c
Method compile_flags(char *p, struct s_subst *s)

```
....
794.                    s->wfile = strdup(wfile);
```

Memory Leak\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=433
Status	New

	Source	Destination
File	cosmopolitan/crypt_blowfish.c	cosmopolitan/crypt_blowfish.c
Line	657	657
Object	data	data

Code Snippet
File Name cosmopolitan/crypt_blowfish.c
Method static char *BF_crypt(const char *key, const char *setting,

```
....
657.          if (!(data = gc(malloc(sizeof(*data))))) return 0;
```

Memory Leak\Path 19:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=434
Status	New

	Source	Destination
File	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	57	57
Object	a	a

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test4) {

```
....
57.     a = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Memory Leak\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=435
Status	New

	Source	Destination
File	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	58	58
Object	b	b

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test4) {

```
....
58.     b = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Memory Leak\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=436
Status	New

	Source	Destination
File	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	59	59

Object	c	c

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test4) {

```
....
59.     c = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Memory Leak\Path 22:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=437
Status New

	Source	Destination
File	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	85	85
Object	a	a

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test64) {

```
....
85.     a = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Memory Leak\Path 23:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=438
Status New

	Source	Destination
File	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	86	86
Object	b	b

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test64) {

```
....
86.     b = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Memory Leak\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=439
Status	New

	Source	Destination
File	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	87	87
Object	c	c

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test64) {

```
....
87.     c = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Memory Leak\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=440
Status	New

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	394	394
Object	__getopt_nonoption_flags	__getopt_nonoption_flags

Code Snippet
File Name cosmopolitan/getopt.c
Method _getopt_initialize (int argc, char *const *argv, const char *optstring)

```
....
394.                __getopt_nonoption_flags =
```

Memory Leak\Path 26:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=441
Status	New

	Source	Destination
File	cosmopolitan/qjsc.c	cosmopolitan/qjsc.c
Line	106	106
Object	name	name

Code Snippet
File Name cosmopolitan/qjsc.c
Method void namelist_add(namelist_t *lp, const char *name, const char *short_name,

```
....
106.      e->name = strdup(name);
```

Memory Leak\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=442
Status	New

	Source	Destination
File	cosmopolitan/qjsc.c	cosmopolitan/qjsc.c
Line	108	108
Object	short_name	short_name

Code Snippet
File Name cosmopolitan/qjsc.c
Method void namelist_add(namelist_t *lp, const char *name, const char *short_name,

```
....
108.          e->short_name = strdup(short_name);
```

Memory Leak\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=443
Status	New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	2426	2426

Object	f	f

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_os_readdir(JSContext *ctx, JSValueConst this_val,

```
....
2426.       f = opendir(path);
```

Memory Leak\Path 29:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=444
Status New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	3069	3069
Object	sab	sab

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static void *js_sab_alloc(void *opaque, size_t size)

```
....
3069.       sab = malloc(sizeof(JSSABHeader) + size);
```

Memory Leak\Path 30:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=445
Status New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	3103	3103
Object	ps	ps

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSWorkerMessagePipe *js_new_message_pipe(void)

```
....
3103.       ps = malloc(sizeof(*ps));
```

Memory Leak\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=446
Status	New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	3299	3299
Object	filename	filename

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_worker_ctor(JSContext *ctx, JSValueConst new_target,

```
....
3299.       args->filename = strdup(filename);
```

Memory Leak\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=447
Status	New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	3300	3300
Object	basename_	basename_

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_worker_ctor(JSContext *ctx, JSValueConst new_target,

```
....
3300.       args->basename_ = strdup(basename);
```

Memory Leak\Path 33:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=448
Status	New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	3369	3369
Object	data	data

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_worker_postMessage(JSContext *ctx, JSValueConst this_val,

```
....
3369.       msg->data = malloc(data_len);
```

Memory Leak\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=449
Status	New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	3690	3690
Object	ts	ts

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method void js_std_init_handlers(JSRuntime *rt)

```
....
3690.       ts = malloc(sizeof(*ts));
```

Memory Leak\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=450
Status	New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	553	553

Object	agent	agent

Code Snippet
File Name cosmopolitan/run-test262.c
Method static JSValue js_agent_start(JSContext *ctx, JSValue this_val,

```
....
553.        agent = malloc(sizeof(*agent));
```

Memory Leak\Path 36:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=451
Status New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	557	557
Object	script	script

Code Snippet
File Name cosmopolitan/run-test262.c
Method static JSValue js_agent_start(JSContext *ctx, JSValue this_val,

```
....
557.        agent->script = strdup(script);
```

Memory Leak\Path 37:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=452
Status New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	703	703
Object	rep	rep

Code Snippet
File Name cosmopolitan/run-test262.c
Method static JSValue js_agent_report(JSContext *ctx, JSValue this_val,

```
....
703.        rep = malloc(sizeof(*rep));
```

Memory Leak\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=453
Status	New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	704	704
Object	str	str

Code Snippet
File Name cosmopolitan/run-test262.c
Method static JSValue js_agent_report(JSContext *ctx, JSValue this_val,

```
....
704.        rep->str = strdup(str);
```

Memory Leak\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=454
Status	New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1397	1397
Object	desc	desc

Code Snippet
File Name cosmopolitan/run-test262.c
Method char *extract_desc(const char *buf, char style)

```
....
1397.               desc = malloc(len + 1);
```

Memory Leak\Path 40:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=455
Status	New

	Source	Destination
File	cosmopolitan/maketab.c	cosmopolitan/maketab.c
Line	174	174
Object	names	names

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

```
....
174.              names[tok-FIRSTTOKEN] = strdup(name);
```

Use of Zero Initialized Pointer

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

Categories

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

Description
Use of Zero Initialized Pointer\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=604
Status	New

The variable declared in args at cosmopolitan/quickjs-libc.c in line 3262 is not initialized when it is used by args at cosmopolitan/quickjs-libc.c in line 3262.

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	3266	3295
Object	args	args

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_worker_ctor(JSContext *ctx, JSValueConst new_target,

```
....
3266.      WorkerFuncArgs *args = NULL;
....
3295.      args = malloc(sizeof(*args));
```

Use of Zero Initialized Pointer\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=605
Status	New

The variable declared in vector at cosmopolitan/sds.c in line 971 is not initialized when it is used by vector at cosmopolitan/sds.c in line 971.

	Source	Destination
File	cosmopolitan/sds.c	cosmopolitan/sds.c
Line	974	1068
Object	vector	vector

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

```
....
974.      char **vector = NULL;
....
1068.                  vector = new_vector;
```

Use of Zero Initialized Pointer\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=606
Status	New

The variable declared in vector at cosmopolitan/sds.c in line 971 is not initialized when it is used by vector at cosmopolitan/sds.c in line 971.

	Source	Destination
File	cosmopolitan/sds.c	cosmopolitan/sds.c
Line	974	1082
Object	vector	vector

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

```
....
974.      char **vector = NULL;
....
1082.          sdsfree(vector[*argc]);
```

Use of Zero Initialized Pointer\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=607
Status	New

The variable declared in dig_signed at cosmopolitan/ssl_srv.c in line 3250 is not initialized when it is used by dig_signed at cosmopolitan/ssl_srv.c in line 3250.

	Source	Destination
File	cosmopolitan/ssl_srv.c	cosmopolitan/ssl_srv.c
Line	3257	3417
Object	dig_signed	dig_signed

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

```
....
3257.      unsigned char *dig_signed = NULL;
....
3417.          size_t dig_signed_len = ssl->out_msg + ssl->out_msglen -
dig_signed;
```

Use of Zero Initialized Pointer\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=608
Status	New

The variable declared in curve at cosmopolitan/ssl_srv.c in line 3250 is not initialized when it is used by curve at cosmopolitan/ssl_srv.c in line 3250.

	Source	Destination
File	cosmopolitan/ssl_srv.c	cosmopolitan/ssl_srv.c
Line	3368	3384
Object	curve	curve

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

```
....
3368.          const mbedtls_ecp_curve_info **curve = NULL;
....
3384.          ssl->curve = *curve;
```

Use of Zero Initialized Pointer\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=609
Status	New

The variable declared in name at cosmopolitan/bzip2.c in line 1628 is not initialized when it is used by link at cosmopolitan/bzip2.c in line 1641.

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	1633	1651
Object	name	link

Code Snippet

File Name cosmopolitan/bzip2.c

Method Cell *mkCell (void)

```
....
1633.     c->name = NULL;
```

File Name cosmopolitan/bzip2.c

Method Cell *snocString (Cell *root, Char *name)

```
....
1651.         tmp->link = snocString ( tmp->link, name );
```

Use of Zero Initialized Pointer\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=610
Status	New

The variable declared in link at cosmopolitan/bzip2.c in line 1628 is not initialized when it is used by link at cosmopolitan/bzip2.c in line 1641.

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	1634	1651
Object	link	link

Code Snippet

File Name cosmopolitan/bzip2.c

Method	Cell *mkCell (void)

```
....
1634.      c->link = NULL;
```

File Name	cosmopolitan/bzip2.c
Method	Cell *snocString (Cell *root, Char *name)

```
....
1651.          tmp->link = snocString ( tmp->link, name );
```

Use of Zero Initialized Pointer\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=611
Status	New

The variable declared in argList at cosmopolitan/bzip2.c in line 1688 is not initialized when it is used by link at cosmopolitan/bzip2.c in line 1641.

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	1739	1651
Object	argList	link

Code Snippet

File Name	cosmopolitan/bzip2.c
Method	IntNative main (IntNative argc, Char *argv[])

```
....
1739.      argList = NULL;
```

File Name	cosmopolitan/bzip2.c
Method	Cell *snocString (Cell *root, Char *name)

```
....
1651.          tmp->link = snocString ( tmp->link, name );
```

Use of Zero Initialized Pointer\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=612
Status	New

The variable declared in opaque at cosmopolitan/bzlib.c in line 918 is not initialized when it is used by bzfp at cosmopolitan/bzlib.c in line 1384.

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	950	1441
Object	opaque	bzfp

Code Snippet
File Name cosmopolitan/bzlib.c
Method BZFILE* BZ2_bzWriteOpen

```
....
950.     bzf->strm.opaque   = NULL;
```

File Name cosmopolitan/bzlib.c
Method BZFILE * bzopen_or_bzdopen

```
....
1441.        bzfp = BZ2_bzWriteOpen(&bzerr,fp,blockSize100k,
```

Use of Zero Initialized Pointer\Path 10:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=613
Status New

The variable declared in bzfree at cosmopolitan/bzlib.c in line 918 is not initialized when it is used by bzfp at cosmopolitan/bzlib.c in line 1384.

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	949	1441
Object	bzfree	bzfp

Code Snippet
File Name cosmopolitan/bzlib.c
Method BZFILE* BZ2_bzWriteOpen

```
....
949.     bzf->strm.bzfree   = NULL;
```

File Name cosmopolitan/bzlib.c

Method	BZFILE * bzopen_or_bzdopen

```
....
1441.        bzfp = BZ2_bzWriteOpen(&bzerr,fp,blockSize100k,
```

Use of Zero Initialized Pointer\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=614
Status	New

The variable declared in bzalloc at cosmopolitan/bzlib.c in line 918 is not initialized when it is used by bzfp at cosmopolitan/bzlib.c in line 1384.

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	948	1441
Object	bzalloc	bzfp

Code Snippet

File Name	cosmopolitan/bzlib.c
Method	BZFILE* BZ2_bzWriteOpen

```
....
948.     bzf->strm.bzalloc  = NULL;
```

File Name	cosmopolitan/bzlib.c
Method	BZFILE * bzopen_or_bzdopen

```
....
1441.        bzfp = BZ2_bzWriteOpen(&bzerr,fp,blockSize100k,
```

Use of Zero Initialized Pointer\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=615
Status	New

The variable declared in bzalloc at cosmopolitan/bzlib.c in line 1089 is not initialized when it is used by bzfp at cosmopolitan/bzlib.c in line 1384.

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c

Line	1122	1444
Object	bzalloc	bzfp

Code Snippet
File Name cosmopolitan/bzlib.c
Method BZFILE* BZ2_bzReadOpen

```
....
1122.     bzf->strm.bzalloc  = NULL;
```

File Name cosmopolitan/bzlib.c

Method BZFILE * bzopen_or_bzdopen

```
....
1444.         bzfp = BZ2_bzReadOpen(&bzerr,fp,verbosity,smallMode,
```

Use of Zero Initialized Pointer\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=616
Status	New

The variable declared in bzfree at cosmopolitan/bzlib.c in line 1089 is not initialized when it is used by bzfp at cosmopolitan/bzlib.c in line 1384.

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	1123	1444
Object	bzfree	bzfp

Code Snippet
File Name cosmopolitan/bzlib.c
Method BZFILE* BZ2_bzReadOpen

```
....
1123.     bzf->strm.bzfree   = NULL;
```

File Name cosmopolitan/bzlib.c

Method BZFILE * bzopen_or_bzdopen

```
....
1444.         bzfp = BZ2_bzReadOpen(&bzerr,fp,verbosity,smallMode,
```

Use of Zero Initialized Pointer\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=617
Status	New

The variable declared in opaque at cosmopolitan/bzlib.c in line 1089 is not initialized when it is used by bzfp at cosmopolitan/bzlib.c in line 1384.

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	1124	1444
Object	opaque	bzfp

Code Snippet

File Name cosmopolitan/bzlib.c

Method BZFILE* BZ2_bzReadOpen

```
....
1124.     bzf->strm.opaque   = NULL;
```

File Name cosmopolitan/bzlib.c

Method BZFILE * bzopen_or_bzdopen

```
....
1444.        bzfp = BZ2_bzReadOpen(&bzerr,fp,verbosity,smallMode,
```

Use of Zero Initialized Pointer\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=618
Status	New

The variable declared in opaque at cosmopolitan/bzlib.c in line 1300 is not initialized when it is used by state at cosmopolitan/bzlib.c in line 495.

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	1319	514
Object	opaque	state

Code Snippet

File Name cosmopolitan/bzlib.c

Method int BZ2_bzBuffToBuffDecompress

```
....
1319.     strm.opaque = NULL;
```

File Name cosmopolitan/bzlib.c

Method int BZ2_bzDecompressInit

```
....
514.     strm->state              = s;
```

Use of Zero Initialized Pointer\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=619
Status	New

The variable declared in bzfree at cosmopolitan/bzlib.c in line 1300 is not initialized when it is used by state at cosmopolitan/bzlib.c in line 495.

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	1318	514
Object	bzfree	state

Code Snippet

File Name cosmopolitan/bzlib.c

Method int BZ2_bzBuffToBuffDecompress

```
....
1318.     strm.bzfree = NULL;
```

File Name cosmopolitan/bzlib.c

Method int BZ2_bzDecompressInit

```
....
514.     strm->state              = s;
```

Use of Zero Initialized Pointer\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=620
Status	New

The variable declared in bzalloc at cosmopolitan/bzlib.c in line 1300 is not initialized when it is used by state at cosmopolitan/bzlib.c in line 495.

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	1317	514
Object	bzalloc	state

Code Snippet
File Name cosmopolitan/bzlib.c
Method int BZ2_bzBuffToBuffDecompress

```
....
1317.     strm.bzalloc = NULL;
```

File Name cosmopolitan/bzlib.c

Method int BZ2_bzDecompressInit

```
....
514.     strm->state              = s;
```

Use of Zero Initialized Pointer\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=621
Status	New

The variable declared in opaque at cosmopolitan/bzlib.c in line 1089 is not initialized when it is used by state at cosmopolitan/bzlib.c in line 495.

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	1124	514
Object	opaque	state

Code Snippet
File Name cosmopolitan/bzlib.c
Method BZFILE* BZ2_bzReadOpen

```
....
1124.     bzf->strm.opaque   = NULL;
```

File Name cosmopolitan/bzlib.c

Method int BZ2_bzDecompressInit

```
....
514.     strm->state              = s;
```

Use of Zero Initialized Pointer\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=622
Status	New

The variable declared in bzalloc at cosmopolitan/bzlib.c in line 1089 is not initialized when it is used by state at cosmopolitan/bzlib.c in line 495.

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	1122	514
Object	bzalloc	state

Code Snippet
File Name cosmopolitan/bzlib.c
Method BZFILE* BZ2_bzReadOpen

```
....
1122.     bzf->strm.bzalloc  = NULL;
```

File Name cosmopolitan/bzlib.c

Method int BZ2_bzDecompressInit

```
....
514.     strm->state              = s;
```

Use of Zero Initialized Pointer\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=623
Status	New

The variable declared in bzfree at cosmopolitan/bzlib.c in line 1089 is not initialized when it is used by state at cosmopolitan/bzlib.c in line 495.

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c

Line	1123	514
Object	bzfree	state

Code Snippet
File Name cosmopolitan/bzlib.c
Method BZFILE* BZ2_bzReadOpen

```
....
1123.     bzf->strm.bzfree   = NULL;
```

File Name cosmopolitan/bzlib.c
Method int BZ2_bzDecompressInit

```
....
514.     strm->state              = s;
```

Use of Zero Initialized Pointer\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=624
Status	New

The variable declared in opaque at cosmopolitan/bzlib.c in line 1248 is not initialized when it is used by zbits at cosmopolitan/bzlib.c in line 337.

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	1270	350
Object	opaque	zbits

Code Snippet
File Name cosmopolitan/bzlib.c
Method int BZ2_bzBuffToBuffCompress

```
....
1270.     strm.opaque = NULL;
```

File Name cosmopolitan/bzlib.c
Method Bool copy_output_until_stop (EState* s)

```
....
350.        *(s->strm->next_out) = s->zbits[s->state_out_pos];
```

Use of Zero Initialized Pointer\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=625
Status	New

The variable declared in bzfree at cosmopolitan/bzlib.c in line 1248 is not initialized when it is used by zbits at cosmopolitan/bzlib.c in line 337.

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	1269	350
Object	bzfree	zbits

Code Snippet

File Name cosmopolitan/bzlib.c

Method int BZ2_bzBuffToBuffCompress

```
....
1269.     strm.bzfree = NULL;
```

File Name cosmopolitan/bzlib.c

Method Bool copy_output_until_stop (EState* s)

```
....
350.        *(s->strm->next_out) = s->zbits[s->state_out_pos];
```

Use of Zero Initialized Pointer\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=626
Status	New

The variable declared in bzalloc at cosmopolitan/bzlib.c in line 1248 is not initialized when it is used by zbits at cosmopolitan/bzlib.c in line 337.

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	1268	350
Object	bzalloc	zbits

Code Snippet

File Name cosmopolitan/bzlib.c

Method int BZ2_bzBuffToBuffCompress

```
....
1268.     strm.bzalloc = NULL;
```

File Name cosmopolitan/bzlib.c

Method Bool copy_output_until_stop (EState* s)

```
....
350.         *(s->strm->next_out) = s->zbits[s->state_out_pos];
```

Use of Zero Initialized Pointer\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=627
Status	New

The variable declared in fp at cosmopolitan/bzlib.c in line 1384 is not initialized when it is used by bzf at cosmopolitan/bzlib.c in line 918.

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	1430	946
Object	fp	bzf

Code Snippet

File Name cosmopolitan/bzlib.c

Method BZFILE * bzopen_or_bzdopen

```
....
1430.         fp = NULL;
```

File Name cosmopolitan/bzlib.c

Method BZFILE* BZ2_bzWriteOpen

```
....
946.     bzf->handle         = f;
```

Use of Zero Initialized Pointer\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=628
Status	New

The variable declared in fp at cosmopolitan/bzlib.c in line 1384 is not initialized when it is used by bzf at cosmopolitan/bzlib.c in line 1089.

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	1430	1119
Object	fp	bzf

Code Snippet
File Name cosmopolitan/bzlib.c
Method BZFILE * bzopen_or_bzdopen

```
....
1430.         fp = NULL;
```

File Name cosmopolitan/bzlib.c
Method BZFILE* BZ2_bzReadOpen

```
....
1119.     bzf->handle        = f;
```

Use of Zero Initialized Pointer\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=629
Status	New

The variable declared in multis at cosmopolitan/compile.c in line 813 is not initialized when it is used by multis at cosmopolitan/compile.c in line 813.

	Source	Destination
File	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	824	884
Object	multis	multis

Code Snippet
File Name cosmopolitan/compile.c
Method compile_tr(char *p, struct s_tr **py)

```
....
824.         y->multis = NULL;
....
884.                                  (y->nmultis + 1) * sizeof(*y-
>multis));
```

Use of Zero Initialized Pointer\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=630
Status	New

The variable declared in multis at cosmopolitan/compile.c in line 813 is not initialized when it is used by multis at cosmopolitan/compile.c in line 813.

	Source	Destination
File	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	824	883
Object	multis	multis

Code Snippet
File Name cosmopolitan/compile.c
Method compile_tr(char *p, struct s_tr **py)

```
....
824.        y->multis = NULL;
....
883.                          y->multis = xrealloc(y->multis,
```

Use of Zero Initialized Pointer\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=631
Status	New

The variable declared in nextchar at cosmopolitan/getopt.c in line 469 is not initialized when it is used by optarg at cosmopolitan/getopt.c in line 469.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	864	646
Object	nextchar	optarg

Code Snippet
File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
864.          nextchar = NULL;
....
646.              optarg = nameend + 1;
```

Use of Zero Initialized Pointer\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=632
Status	New

The variable declared in nextchar at cosmopolitan/getopt.c in line 469 is not initialized when it is used by optarg at cosmopolitan/getopt.c in line 469.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	879	646
Object	nextchar	optarg

Code Snippet
File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
879.            nextchar = NULL;
....
646.              optarg = nameend + 1;
```

Use of Zero Initialized Pointer\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=633
Status	New

The variable declared in nextchar at cosmopolitan/getopt.c in line 469 is not initialized when it is used by optarg at cosmopolitan/getopt.c in line 469.

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	910	646
Object	nextchar	optarg

Code Snippet
File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
910.            nextchar = NULL;
....
646.              optarg = nameend + 1;
```

Use of Zero Initialized Pointer\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=634
Status	New

The variable declared in sab_tab at cosmopolitan/quickjs-libc.c in line 3343 is not initialized when it is used by sab_tab at cosmopolitan/quickjs-libc.c in line 3343.

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	3366	3386
Object	sab_tab	sab_tab

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_worker_postMessage(JSContext *ctx, JSValueConst this_val,

```
....
3366.       msg->sab_tab = NULL;
....
3386.           js_sab_dup(NULL, msg->sab_tab[i]);
```

Use of Zero Initialized Pointer\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=635
Status	New

The variable declared in array at cosmopolitan/run-test262.c in line 352 is not initialized when it is used by array at cosmopolitan/run-test262.c in line 352.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	358	355
Object	array	array

Code Snippet
File Name cosmopolitan/run-test262.c
Method void namelist_free(namelist_t *lp)

```
....
358.       lp->array = NULL;
....
355.           free(lp->array[--lp->count]);
```

Use of Zero Initialized Pointer\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=636
Status	New

The variable declared in array at cosmopolitan/run-test262.c in line 352 is not initialized when it is used by array at cosmopolitan/run-test262.c in line 1883.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	358	1889
Object	array	array

Code Snippet

File Name cosmopolitan/run-test262.c

Method void namelist_free(namelist_t *lp)

```
....
358.      lp->array = NULL;
```

File Name cosmopolitan/run-test262.c

Method void run_test_dir_list(namelist_t *lp, int start_index, int stop_index)

```
....
1889.          const char *p = lp->array[i];
```

Use of Zero Initialized Pointer\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=637
Status	New

The variable declared in array at cosmopolitan/run-test262.c in line 352 is not initialized when it is used by array at cosmopolitan/run-test262.c in line 255.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	358	259
Object	array	array

Code Snippet

File Name cosmopolitan/run-test262.c

Method void namelist_free(namelist_t *lp)

```
....
358.      lp->array = NULL;
```

File Name cosmopolitan/run-test262.c

Method void namelist_sort(namelist_t *lp)

```
....
259.          qsort(lp->array, lp->count, sizeof(*lp->array),
namelist_cmp_indirect);
```

Use of Zero Initialized Pointer\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=638
Status	New

The variable declared in array at cosmopolitan/run-test262.c in line 352 is not initialized when it is used by array at cosmopolitan/run-test262.c in line 879.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	358	903
Object	array	array

Code Snippet

File Name cosmopolitan/run-test262.c

Method void namelist_free(namelist_t *lp)

```
....
358.      lp->array = NULL;
```

File Name cosmopolitan/run-test262.c

Method void update_exclude_dirs(void)

```
....
903.          name = lp->array[i];
```

Use of Zero Initialized Pointer\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=639
Status	New

The variable declared in current at cosmopolitan/sds.c in line 971 is not initialized when it is used by vector at cosmopolitan/sds.c in line 971.

	Source	Destination
File	cosmopolitan/sds.c	cosmopolitan/sds.c
Line	1071	1069
Object	current	vector

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

```
....
1071.                   current = NULL;
....
1069.                   vector[*argc] = current;
```

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=1050080&projectid=50070&pathid=141
Status New

Calling free() (line 1688) on a variable that was not dynamically allocated (line 1688) in file cosmopolitan/bzip2.c may result with a crash.

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	1938	1938
Object	aa	aa

Code Snippet
File Name cosmopolitan/bzip2.c
Method IntNative main (IntNative argc, Char *argv[])

```
....
1938.          free(aa);
```

MemoryFree on StackVariable\Path 2:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=142
Status	New

Calling free() (line 1023) on a variable that was not dynamically allocated (line 1023) in file cosmopolitan/bzlib.c may result with a crash.

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	1084	1084
Object	bzf	bzf

Code Snippet
File Name cosmopolitan/bzlib.c
Method void BZ2_bzWriteClose64

```
....
1084.     free ( bzf );
```

MemoryFree on StackVariable\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=143
Status	New

Calling free() (line 1145) on a variable that was not dynamically allocated (line 1145) in file cosmopolitan/bzlib.c may result with a crash.

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	1158	1158
Object	bzf	bzf

Code Snippet
File Name cosmopolitan/bzlib.c
Method void BZ2_bzReadClose (int *bzerror, BZFILE *b)

```
....
1158.     free ( bzf );
```

MemoryFree on StackVariable\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=144
Status	New

Calling free() (line 1097) on a variable that was not dynamically allocated (line 1097) in file cosmopolitan/compile.c may result with a crash.

	Source	Destination
File	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	1108	1108
Object	lh	lh

Code Snippet
File Name cosmopolitan/compile.c
Method uselabel(void)

```
....
1108.                    free(lh);
```

MemoryFree on StackVariable\Path 5:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=145
Status New

Calling free() (line 214) on a variable that was not dynamically allocated (line 214) in file cosmopolitan/qjsc.c may result with a crash.

	Source	Destination
File	cosmopolitan/qjsc.c	cosmopolitan/qjsc.c
Line	229	229
Object	cname1	cname1

Code Snippet
File Name cosmopolitan/qjsc.c
Method static void find_unique_cname(char *cname, size_t cname_size)

```
....
229.          free(cname1);
```

MemoryFree on StackVariable\Path 6:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=146
Status New

Calling free() (line 214) on a variable that was not dynamically allocated (line 214) in file cosmopolitan/qjsc.c may result with a crash.

	Source	Destination
File	cosmopolitan/qjsc.c	cosmopolitan/qjsc.c
Line	236	236
Object	cname1	cname1

Code Snippet
File Name cosmopolitan/qjsc.c
Method static void find_unique_cname(char *cname, size_t cname_size)

```
....
236.        free(cname1);
```

MemoryFree on StackVariable\Path 7:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=147
Status New

Calling free() (line 3076) on a variable that was not dynamically allocated (line 3076) in file cosmopolitan/quickjs-libc.c may result with a crash.

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	3084	3084
Object	sab	sab

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static void js_sab_free(void *opaque, void *ptr)

```
....
3084.           free(sab);
```

MemoryFree on StackVariable\Path 8:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=148
Status New

Calling free() (line 3184) on a variable that was not dynamically allocated (line 3184) in file cosmopolitan/quickjs-libc.c may result with a crash.

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c

Line	3220	3220
Object	args	args

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static void *worker_func(void *opaque)

```
....
3220.       free(args);
```

MemoryFree on StackVariable\Path 9:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=149
Status New

Calling free() (line 3716) on a variable that was not dynamically allocated (line 3716) in file cosmopolitan/quickjs-libc.c may result with a crash.

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	3744	3744
Object	ts	ts

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method void js_std_free_handlers(JSRuntime *rt)

```
....
3744.       free(ts);
```

MemoryFree on StackVariable\Path 10:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=150
Status New

Calling free() (line 315) on a variable that was not dynamically allocated (line 315) in file cosmopolitan/run-test262.c may result with a crash.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	334	334
Object	base_name	base_name

Code Snippet	
File Name	cosmopolitan/run-test262.c
Method	void namelist_load(namelist_t *lp, const char *filename)

```
....
334.        free(base_name);
```

MemoryFree on StackVariable\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=151
Status	New

Calling free() (line 338) on a variable that was not dynamically allocated (line 338) in file cosmopolitan/run-test262.c may result with a crash.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	348	348
Object	pp	pp

Code Snippet	
File Name	cosmopolitan/run-test262.c
Method	void namelist_add_from_error_file(namelist_t *lp, const char *file)

```
....
348.            free(pp);
```

MemoryFree on StackVariable\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=152
Status	New

Calling free() (line 564) on a variable that was not dynamically allocated (line 564) in file cosmopolitan/run-test262.c may result with a crash.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	574	574
Object	agent	agent

Code Snippet	
File Name	cosmopolitan/run-test262.c

Method	static void js_agent_free(JSContext *ctx)

```
....
574.            free(agent);
```

MemoryFree on StackVariable\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=153
Status	New

Calling free() (line 670) on a variable that was not dynamically allocated (line 670) in file cosmopolitan/run-test262.c may result with a crash.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	687	687
Object	rep	rep

Code Snippet	
File Name	cosmopolitan/run-test262.c
Method	static JSValue js_agent_getReport(JSContext *ctx, JSValue this_val,

```
....
687.            free(rep);
```

MemoryFree on StackVariable\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=154
Status	New

Calling free() (line 1177) on a variable that was not dynamically allocated (line 1177) in file cosmopolitan/run-test262.c may result with a crash.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1265	1265
Object	error_class	error_class

Code Snippet	
File Name	cosmopolitan/run-test262.c
Method	static int eval_buf(JSContext *ctx, const char *buf, size_t buf_len,

```
....
1265.                   free(error_class);
```

MemoryFree on StackVariable\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=155
Status	New

Calling free() (line 1177) on a variable that was not dynamically allocated (line 1177) in file cosmopolitan/run-test262.c may result with a crash.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1344	1344
Object	s	s

Code Snippet
File Name cosmopolitan/run-test262.c
Method static int eval_buf(JSContext *ctx, const char *buf, size_t buf_len,

```
....
1344.           free(s);
```

MemoryFree on StackVariable\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=156
Status	New

Calling free() (line 1566) on a variable that was not dynamically allocated (line 1566) in file cosmopolitan/run-test262.c may result with a crash.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1611	1611
Object	ifile	ifile

Code Snippet
File Name cosmopolitan/run-test262.c
Method int run_test(const char *filename, int index)

```
....
1611.                       free(ifile);
```

MemoryFree on StackVariable\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=157
Status	New

Calling free() (line 1566) on a variable that was not dynamically allocated (line 1566) in file cosmopolitan/run-test262.c may result with a crash.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1637	1637
Object	option	option

Code Snippet
File Name cosmopolitan/run-test262.c
Method int run_test(const char *filename, int index)

```
....
1637.                       free(option);
```

MemoryFree on StackVariable\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=158
Status	New

Calling free() (line 1566) on a variable that was not dynamically allocated (line 1566) in file cosmopolitan/run-test262.c may result with a crash.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1664	1664
Object	option	option

Code Snippet
File Name cosmopolitan/run-test262.c
Method int run_test(const char *filename, int index)

```
....
1664.                    free(option);
```

MemoryFree on StackVariable\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=159
Status	New

Calling free() (line 1566) on a variable that was not dynamically allocated (line 1566) in file cosmopolitan/run-test262.c may result with a crash.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1667	1667
Object	desc	desc

Code Snippet
File Name cosmopolitan/run-test262.c
Method int run_test(const char *filename, int index)

```
....
1667.              free(desc);
```

MemoryFree on StackVariable\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=160
Status	New

Calling free() (line 1566) on a variable that was not dynamically allocated (line 1566) in file cosmopolitan/run-test262.c may result with a crash.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1697	1697
Object	ifile	ifile

Code Snippet
File Name cosmopolitan/run-test262.c
Method int run_test(const char *filename, int index)

```
....
1697.               free(ifile);
```

MemoryFree on StackVariable\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=161
Status	New

Calling free() (line 1566) on a variable that was not dynamically allocated (line 1566) in file cosmopolitan/run-test262.c may result with a crash.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1715	1715
Object	desc	desc

Code Snippet
File Name cosmopolitan/run-test262.c
Method int run_test(const char *filename, int index)

```
....
1715.               free(desc);
```

MemoryFree on StackVariable\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=162
Status	New

Calling free() (line 1566) on a variable that was not dynamically allocated (line 1566) in file cosmopolitan/run-test262.c may result with a crash.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1796	1796
Object	error_type	error_type

Code Snippet
File Name cosmopolitan/run-test262.c
Method int run_test(const char *filename, int index)

```
....
1796.       free(error_type);
```

MemoryFree on StackVariable\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=163
Status	New

Calling free() (line 1566) on a variable that was not dynamically allocated (line 1566) in file cosmopolitan/run-test262.c may result with a crash.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1797	1797
Object	buf	buf

Code Snippet
File Name cosmopolitan/run-test262.c
Method int run_test(const char *filename, int index)

```
....
1797.       free(buf);
```

MemoryFree on StackVariable\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=164
Status	New

Calling free() (line 1803) on a variable that was not dynamically allocated (line 1803) in file cosmopolitan/run-test262.c may result with a crash.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1859	1859
Object	buf	buf

Code Snippet
File Name cosmopolitan/run-test262.c
Method int run_test262_harness_test(const char *filename, BOOL is_module)

```
....
1859.       free(buf);
```

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=1050080&projectid=50070&pathid=165
Status	New

The function data_len in cosmopolitan/quickjs-libc.c at line 3343 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	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	3369	3369
Object	data_len	data_len

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_worker_postMessage(JSContext *ctx, JSValueConst this_val,

```
....
3369.       msg->data = malloc(data_len);
```

Wrong Size t Allocation\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=166
Status	New

The function newsize in cosmopolitan/stb_image_write_png.c at line 141 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	cosmopolitan/stb_image_write_png.c	cosmopolitan/stb_image_write_png.c
Line	150	150
Object	newsize	newsize

Code Snippet	
File Name	cosmopolitan/stb_image_write_png.c
Method	static unsigned char *stbi_zlib_compress(unsigned char *data, int size,

```
....
150.      if ((trimdata = realloc(newdata, newsize))) {
```

Wrong Size t Allocation\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=167
Status	New

The function asize in cosmopolitan/compile.c at line 634 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	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	649	649
Object	asize	asize

Code Snippet	
File Name	cosmopolitan/compile.c
Method	compile_subst(char *p, struct s_subst *s)

```
....
649.        text = xmalloc(asize);
```

Wrong Size t Allocation\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=168
Status	New

The function asize in cosmopolitan/compile.c at line 902 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	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	910	910
Object	asize	asize

Code Snippet	
File Name	cosmopolitan/compile.c
Method	compile_text(void)

```
....
910.        text = xmalloc(asize);
```

Wrong Size t Allocation\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=169
Status	New

The function len in cosmopolitan/compile.c at line 994 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	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	1007	1007
Object	len	len

Code Snippet
File Name cosmopolitan/compile.c
Method duptoeol(char *s, const char *ctype)

```
....
1007.        p = xmalloc(len);
```

Wrong Size t Allocation\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=170
Status	New

The function size in cosmopolitan/compile.c at line 634 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	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	705	705
Object	size	size

Code Snippet
File Name cosmopolitan/compile.c
Method compile_subst(char *p, struct s_subst *s)

```
....
705.                            s->new = xrealloc(text, size);
```

Wrong Size t Allocation\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=171
Status	New

The function asize in cosmopolitan/compile.c at line 634 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	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	717	717
Object	asize	asize

Code Snippet
File Name cosmopolitan/compile.c
Method compile_subst(char *p, struct s_subst *s)

```
....
717.                        text = xrealloc(text, asize);
```

Wrong Size t Allocation\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=172
Status	New

The function asize in cosmopolitan/compile.c at line 902 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	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	927	927
Object	asize	asize

Code Snippet
File Name cosmopolitan/compile.c
Method compile_text(void)

```
....
927.                    text = xrealloc(text, asize);
```

Wrong Size t Allocation\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=173
Status	New

The function n in cosmopolitan/djbsort_test.c at line 54 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	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	57	57
Object	n	n

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test4) {

```
....
57.    a = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Wrong Size t Allocation\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=174
Status	New

The function n in cosmopolitan/djbsort_test.c at line 54 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	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	58	58
Object	n	n

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test4) {

```
....
58.     b = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Wrong Size t Allocation\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=175
Status	New

The function n in cosmopolitan/djbsort_test.c at line 54 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	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	59	59
Object	n	n

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test4) {

```
....
59.     c = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Wrong Size t Allocation\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=176
Status	New

The function n in cosmopolitan/djbsort_test.c at line 68 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	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	85	85
Object	n	n

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test64) {

```
....
85.     a = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Wrong Size t Allocation\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=177
Status	New

The function n in cosmopolitan/djbsort_test.c at line 68 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	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	86	86
Object	n	n

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test64) {

```
....
86.     b = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Wrong Size t Allocation\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=178
Status	New

The function n in cosmopolitan/djbsort_test.c at line 68 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	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	87	87
Object	n	n

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test64) {

```
....
87.     c = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Wrong Size t Allocation\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=179
Status	New

The function buf_len in cosmopolitan/quickjs-libc.c at line 353 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	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	379	379
Object	buf_len	buf_len

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method uint8_t *js_load_file(JSContext *ctx, size_t *pbuf_len, const char *filename)

```
....
379.          buf = malloc(buf_len + 1);
```

Wrong Size t Allocation\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=180
Status	New

The function newsize in cosmopolitan/stb_image_write_png.c at line 141 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	cosmopolitan/stb_image_write_png.c	cosmopolitan/stb_image_write_png.c
Line	146	146
Object	newsize	newsize

Code Snippet
File Name cosmopolitan/stb_image_write_png.c
Method static unsigned char *stbi_zlib_compress(unsigned char *data, int size,

```
....
146.    if ((newdata = malloc((newsize = compressBound(size)))) &&
```

Wrong Size t Allocation\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=181
Status	New

The function slen in cosmopolitan/process.c at line 657 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	cosmopolitan/process.c	cosmopolitan/process.c
Line	673	673
Object	slen	slen

Code Snippet
File Name cosmopolitan/process.c
Method regexec_e(regex_t *preg, const char *string, int eflags, int nomatch,

```
....
673.        buf = xmalloc(slen + 1);
```

Wrong Size t Allocation\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=182
Status	New

The function size in cosmopolitan/compile.c at line 902 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	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	931	931
Object	size	size

Code Snippet
File Name cosmopolitan/compile.c
Method compile_text(void)

```
....
931.        p = xrealloc(text, size + 1);
```

Wrong Size t Allocation\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=183
Status	New

The function len in cosmopolitan/run-test262.c at line 144 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	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	151	151
Object	len	len

Code Snippet
File Name cosmopolitan/run-test262.c
Method char *str_append(char **pp, const char *sep, const char *str) {

```
....
151.      res = malloc(len + strlen(str) + 1);
```

Wrong Size t Allocation\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=184
Status	New

The function appendnum in cosmopolitan/process.c at line 88 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	cosmopolitan/process.c	cosmopolitan/process.c
Line	116	116
Object	appendnum	appendnum

Code Snippet
File Name cosmopolitan/process.c
Method process(void)

```
....
116.                                    (appendnum *= 2));
```

Wrong Size t Allocation\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=185
Status	New

The function appendnum in cosmopolitan/process.c at line 88 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	cosmopolitan/process.c	cosmopolitan/process.c
Line	209	209
Object	appendnum	appendnum

Code Snippet
File Name cosmopolitan/process.c
Method process(void)

```
....
209.                                    (appendnum *= 2));
```

Use of Uninitialized Variable

Query Path:
CPP\Cx\CPP Medium Threat\Use of Uninitialized Variable Version:0

Categories

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

Description

Use of Uninitialized Variable\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=598
Status	New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	76	2074
Object	dump_memory	dump_memory

Code Snippet

File Name cosmopolitan/run-test262.c
Method int dump_memory;

```
....
76.  int dump_memory;
```

File Name cosmopolitan/run-test262.c

Method int main(int argc, char **argv)

```
....
2074.       if (dump_memory) {
```

Use of Uninitialized Variable\Path 2:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=599
Status New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	76	1975
Object	dump_memory	dump_memory

Code Snippet
File Name cosmopolitan/run-test262.c
Method int dump_memory;

```
....
76.  int dump_memory;
```

File Name cosmopolitan/run-test262.c

Method int main(int argc, char **argv)

```
....
1975.               dump_memory++;
```

Use of Uninitialized Variable\Path 3:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=600
Status New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	76	2075
Object	dump_memory	dump_memory

Code Snippet
File Name cosmopolitan/run-test262.c
Method int dump_memory;

```
....
76.  int dump_memory;
```

File Name cosmopolitan/run-test262.c

Method int main(int argc, char **argv)

```
....
2075.          if (dump_memory > 1 && stats_count > 1) {
```

Use of Uninitialized Variable\Path 4:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=601
Status New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	76	1546
Object	dump_memory	dump_memory

Code Snippet
File Name cosmopolitan/run-test262.c
Method int dump_memory;

```
....
76.  int dump_memory;
```

File Name cosmopolitan/run-test262.c

Method int run_test_buf(const char *filename, char *harness, namelist_t *ip,

```
....
1546.       if (dump_memory) {
```

Use of Uninitialized Variable\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=602
Status	New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	90	2026
Object	update_errors	update_errors

Code Snippet

File Name cosmopolitan/run-test262.c

Method int update_errors;

```
....
90.  int update_errors;
```

File Name cosmopolitan/run-test262.c

Method int main(int argc, char **argv)

```
....
2026.          if (update_errors) {
```

Use of Uninitialized Variable\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=603
Status	New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	90	1983
Object	update_errors	update_errors

Code Snippet

File Name cosmopolitan/run-test262.c

Method int update_errors;

```
....
90.  int update_errors;
```

File Name	cosmopolitan/run-test262.c
Method	int main(int argc, char **argv)

```
....
1983.               update_errors++;
```

Short Overflow

Query Path:
CPP\Cx\CPP Integer Overflow\Short Overflow Version:1

Categories

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

Description

Short Overflow\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=189
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 700 of cosmopolitan/puff.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	cosmopolitan/puff.c	cosmopolitan/puff.c
Line	746	746
Object	AssignExpr	AssignExpr

Code Snippet

File Name	cosmopolitan/puff.c
Method	local int dynamic(struct state *s)

```
....
746.               lengths[index++] = symbol;
```

Short Overflow\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=190
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 700 of cosmopolitan/puff.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	cosmopolitan/puff.c	cosmopolitan/puff.c
Line	762	762
Object	AssignExpr	AssignExpr

Code Snippet
File Name cosmopolitan/puff.c
Method local int dynamic(struct state *s)

```
....
762.                    lengths[index++] = len;
```

Short Overflow\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=191
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 465 of cosmopolitan/stb_image_write.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	cosmopolitan/stb_image_write.c	cosmopolitan/stb_image_write.c
Line	472	472
Object	AssignExpr	AssignExpr

Code Snippet
File Name cosmopolitan/stb_image_write.c
Method static void stbiw__jpg_calcBits(int val, unsigned short bits[2]) {

```
....
472.    bits[0] = val & ((1u << bits[1]) - 1);
```

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=1050080&projectid=50070&pathid=413

Status	New

	Source	Destination
File	cosmopolitan/stb_image_write_png.c	cosmopolitan/stb_image_write_png.c
Line	312	156
Object	filt	newdata

Code Snippet
File Name cosmopolitan/stb_image_write_png.c
Method unsigned char *stbi_write_png_to_mem(const unsigned char *pixels,

```
....
312.    free(filt);
```

File Name cosmopolitan/stb_image_write_png.c
Method static unsigned char *stbi_zlib_compress(unsigned char *data, int size,

```
....
156.    free(newdata);
```

Double Free\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=414
Status	New

	Source	Destination
File	cosmopolitan/stb_image_write_png.c	cosmopolitan/stb_image_write_png.c
Line	338	338
Object	zlib	zlib

Code Snippet
File Name cosmopolitan/stb_image_write_png.c
Method unsigned char *stbi_write_png_to_mem(const unsigned char *pixels,

```
....
338.    free(zlib);
```

Double Free\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=415

Status	New

	Source	Destination
File	cosmopolitan/stb_image_write_png.c	cosmopolitan/stb_image_write_png.c
Line	312	150
Object	filt	newdata

Code Snippet
File Name cosmopolitan/stb_image_write_png.c
Method unsigned char *stbi_write_png_to_mem(const unsigned char *pixels,

```
....
312.    free(filt);
```

File Name cosmopolitan/stb_image_write_png.c
Method static unsigned char *stbi_zlib_compress(unsigned char *data, int size,

```
....
150.      if ((trimdata = realloc(newdata, newsize))) {
```

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=1050080&projectid=50070&pathid=63
Status	New

The pointer option at cosmopolitan/run-test262.c in line 1419 is being used after it has been freed.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1458	1458
Object	option	option

Code Snippet
File Name cosmopolitan/run-test262.c
Method static char *get_option(char **pp, int *state)

```
....
1458.       return option;
```

Buffer Overflow AddressOfLocalVarReturned\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=64
Status	New

The pointer aRes at cosmopolitan/sqlite3rbu.c in line 4316 is being used after it has been freed.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	4340	4340
Object	aRes	aRes

Code Snippet
File Name cosmopolitan/sqlite3rbu.c
Method int sqlite3rbu_state(sqlite3rbu *p){

```
....
4340.       return aRes[p->eStage];
```

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=1050080&projectid=50070&pathid=187
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 195 of cosmopolitan/lstrlib.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	cosmopolitan/lstrlib.c	cosmopolitan/lstrlib.c
Line	205	205

Object	AssignExpr	AssignExpr

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

```
....
205.    n = (int)(pose -  posi) + 1;
```

Integer Overflow\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=188
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 3830 of cosmopolitan/ssl_srv.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	cosmopolitan/ssl_srv.c	cosmopolitan/ssl_srv.c
Line	3866	3866
Object	AssignExpr	AssignExpr

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

```
....
3866.        * padding, to protect against timing-based Bleichenbacher-
type
```

Char Overflow

Query Path:
CPP\Cx\CPP Integer Overflow\Char Overflow Version:1

Categories

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

Description

Char Overflow\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=186
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 2908 of cosmopolitan/main.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	cosmopolitan/main.c	cosmopolitan/main.c
Line	3015	3015
Object	AssignExpr	AssignExpr

Code Snippet
File Name cosmopolitan/main.c
Method int sqlite3ParseUri(

```
....
3015.           c = octet;
```

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=1050080&projectid=50070&pathid=597
Status	New

The pointer port at cosmopolitan/quickjs-libc.c in line 3158 is being used after it has been freed.

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	3154	3164
Object	ps	port

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static void js_free_message_pipe(JSWorkerMessagePipe *ps)

```
....
3154.           free(ps);
```

File Name cosmopolitan/quickjs-libc.c

Method static void js_free_port(JSRuntime *rt, JSWorkerMessageHandler *port)

```
....
3164.           js_free_rt(rt, port);
```

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=1050080&projectid=50070&pathid=640
Status	New

The size of the buffer used by *strdup_len in len, at line 132 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that load_config passes to buf, at line 919 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	940	135
Object	buf	len

Code Snippet
File Name cosmopolitan/run-test262.c
Method void load_config(const char *filename)

```
....
940.      while (fgets(buf, sizeof(buf), f) != NULL) {
```

File Name cosmopolitan/run-test262.c
Method char *strdup_len(const char *str, int len)

```
....
135.      memcpy(p, str, len);
```

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=1050080&projectid=50070&pathid=641
Status	New

	Source	Destination
File	cosmopolitan/maketab.c	cosmopolitan/maketab.c
Line	147	147
Object	fgets	fgets

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

```
....
147.        while (fgets(buf, sizeof buf, fp) != NULL) {
```

Improper Resource Access Authorization\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=642
Status	New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	327	327
Object	fgets	fgets

Code Snippet
File Name cosmopolitan/run-test262.c
Method void namelist_load(namelist_t *lp, const char *filename)

```
....
327.      while (fgets(buf, sizeof(buf), f) != NULL) {
```

Improper Resource Access Authorization\Path 3:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=643
Status	New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	940	940
Object	fgets	fgets

Code Snippet
File Name cosmopolitan/run-test262.c
Method void load_config(const char *filename)

```
....
940.      while (fgets(buf, sizeof(buf), f) != NULL) {
```

Improper Resource Access Authorization\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=644
Status	New

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	232	232
Object	fgetc	fgetc

Code Snippet
File Name cosmopolitan/bzip2.c
Method Bool myfeof (FILE* f)

```
....
232.     Int32 c = fgetc ( f );
```

Improper Resource Access Authorization\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=645
Status	New

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	910	910

Object	fgetc	fgetc

Code Snippet
File Name cosmopolitan/bzlib.c
Method static Bool myfeof (FILE* f)

```
....
910.     Int32 c = fgetc ( f );
```

Improper Resource Access Authorization\Path 6:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=646
Status New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	1158	1158
Object	fgetc	fgetc

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_std_file_getline(JSContext *ctx, JSValueConst this_val,

```
....
1158.          c = fgetc(f);
```

Improper Resource Access Authorization\Path 7:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=647
Status New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	1209	1209
Object	fgetc	fgetc

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_std_file_readAsString(JSContext *ctx, JSValueConst this_val,

```
....
1209.           c = fgetc(f);
```

Improper Resource Access Authorization\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=648
Status	New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	1229	1229
Object	fgetc	fgetc

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_std_file_getByte(JSContext *ctx, JSValueConst this_val,

```
....
1229.       return JS_NewInt32(ctx, fgetc(f));
```

Improper Resource Access Authorization\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=649
Status	New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	1258	1258
Object	fgetc	fgetc

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static int http_get_header_line(FILE *f, char *buf, size_t buf_size,

```
....
1258.           c = fgetc(f);
```

Improper Resource Access Authorization\Path 10:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=650
Status	New

	Source	Destination
File	cosmopolitan/maketab.c	cosmopolitan/maketab.c
Line	147	147
Object	buf	buf

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

```
....
147.        while (fgets(buf, sizeof buf, fp) != NULL) {
```

Improper Resource Access Authorization\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=651
Status	New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	327	327
Object	buf	buf

Code Snippet
File Name cosmopolitan/run-test262.c
Method void namelist_load(namelist_t *lp, const char *filename)

```
....
327.      while (fgets(buf, sizeof(buf), f) != NULL) {
```

Improper Resource Access Authorization\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=652
Status	New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	940	940

Object	buf	buf

Code Snippet
File Name cosmopolitan/run-test262.c
Method void load_config(const char *filename)

```
....
940.      while (fgets(buf, sizeof(buf), f) != NULL) {
```

Improper Resource Access Authorization\Path 13:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=653
Status New

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	265	265
Object	ibuf	ibuf

Code Snippet
File Name cosmopolitan/bzip2.c
Method void compressStream (FILE *stream, FILE *zStream)

```
....
265.        nIbuf = fread ( ibuf, sizeof(UChar), 5000, stream );
```

Improper Resource Access Authorization\Path 14:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=654
Status New

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	421	421
Object	obuf	obuf

Code Snippet
File Name cosmopolitan/bzip2.c
Method Bool uncompressStream (FILE *zStream, FILE *stream)

```
....
421.                nread = fread ( obuf, sizeof(UChar), 5000, zStream );
```

Improper Resource Access Authorization\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=655
Status	New

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	1192	1192
Object	buf	buf

Code Snippet
File Name cosmopolitan/bzlib.c
Method int BZ2_bzRead

```
....
1192.            n = fread ( bzf->buf, sizeof(UChar),
```

Improper Resource Access Authorization\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=656
Status	New

	Source	Destination
File	cosmopolitan/process.c	cosmopolitan/process.c
Line	560	560
Object	buf	buf

Code Snippet
File Name cosmopolitan/process.c
Method flush_appends(void)

```
....
560.                    while ((count = fread(buf, sizeof(char),
sizeof(buf), f)))
```

Improper Resource Access Authorization\Path 17:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=657
Status	New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	382	382
Object	buf	buf

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method uint8_t *js_load_file(JSContext *ctx, size_t *pbuf_len, const char *filename)

```
....
382.      if (fread(buf, 1, buf_len, f) != buf_len) {
```

Improper Resource Access Authorization\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=658
Status	New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	1140	1140
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_std_file_read_write(JSContext *ctx, JSValueConst this_val,

```
....
1140.          ret = fread(buf + pos, 1, len, f);
```

Improper Resource Access Authorization\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=659
Status	New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c

Line	1375	1375
Object	buf	buf

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_std_urlGet(JSContext *ctx, JSValueConst this_val,

```
....
1375.           len = fread(buf, 1, URL_GET_BUF_SIZE, f);
```

Improper Resource Access Authorization\Path 20:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=660
Status New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	1632	1632
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_os_read_write(JSContext *ctx, JSValueConst this_val,

```
....
1632.           ret = js_get_errno(read(fd, buf + pos, len));
```

Improper Resource Access Authorization\Path 21:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=661
Status New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	2160	2160
Object	buf	buf

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static int handle_posted_message(JSRuntime *rt, JSContext *ctx,

```
....
2160.                   ret = read(ps->read_fd, buf, sizeof(buf));
```

Improper Resource Access Authorization\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=662
Status	New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	2677	2677
Object	buf	buf

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_os_readlink(JSContext *ctx, JSValueConst this_val,

```
....
2677.       res = readlink(path, buf, sizeof(buf) - 1);
```

Improper Resource Access Authorization\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=663
Status	New

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	46	46
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/bzlib.c
Method void BZ2_bz__AssertH__fail (int errcode)

```
....
46.     fprintf(stderr,
```

Improper Resource Access Authorization\Path 24:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=664
Status	New

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	60	60
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/bzlib.c
Method void BZ2_bz__AssertH__fail (int errcode)

```
....
60.     fprintf(stderr,
```

Improper Resource Access Authorization\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=665
Status	New

	Source	Destination
File	cosmopolitan/common.c	cosmopolitan/common.c
Line	5	5
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/common.c
Method void Assert(long expected, long actual, char *code) {

```
....
5.      fprintf(stderr, "%s => %ld expected but got %ld\n", code,
expected, actual);
```

Improper Resource Access Authorization\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=666
Status	New

	Source	Destination
File	cosmopolitan/common.c	cosmopolitan/common.c

Line	12	12
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/common.c
Method void Assert2(long expected, long actual, char *code, char *func, int line) {

```
....
12.      fprintf(stderr, "%s:%d: %s => expected %ld but got %ld\n",
func, line, code,
```

Improper Resource Access Authorization\Path 27:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=667
Status New

	Source	Destination
File	cosmopolitan/common.c	cosmopolitan/common.c
Line	20	20
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/common.c
Method void Assert128(__int128 k, __int128 x, char *code, char *func, int line) {

```
....
20.      fprintf(stderr, "%s:%d: %s => want %jjd but got %jjd\n", func,
line, code,
```

Improper Resource Access Authorization\Path 28:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=668
Status New

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	629	629
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
629.            fprintf (stderr, _("%s: option '%s' is ambiguous\n"),
```

Improper Resource Access Authorization\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=669
Status	New

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	652	652
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
652.                fprintf (stderr,
```

Improper Resource Access Authorization\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=670
Status	New

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	657	657
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
657.                fprintf (stderr,
```

Improper Resource Access Authorization\Path 31:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=671
Status	New

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	674	674
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
674.                  fprintf (stderr,
```

Improper Resource Access Authorization\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=672
Status	New

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	704	704
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
704.              fprintf (stderr, _("%s: unrecognized option '--
%s'\n"),
```

Improper Resource Access Authorization\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=673
Status	New

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c

Line	708	708
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
708.              fprintf (stderr, _("%s: unrecognized option
'%c%s'\n"),
```

Improper Resource Access Authorization\Path 34:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=674
Status New

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	734	734
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
734.              fprintf (stderr, _("%s: illegal option -- %c\n"),
```

Improper Resource Access Authorization\Path 35:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=675
Status New

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	737	737
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
737.              fprintf (stderr, _("%s: invalid option -- %c\n"),
```

Improper Resource Access Authorization\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=676
Status	New

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	767	767
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
767.              fprintf (stderr, _("%s: option requires an argument --
%c\n"),
```

Improper Resource Access Authorization\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=677
Status	New

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	814	814
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
814.              fprintf (stderr, _("%s: option '-W %s' is
ambiguous\n"),
```

Improper Resource Access Authorization\Path 38:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=678
Status	New

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	847	847
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
847.                    fprintf (stderr,
```

Improper Resource Access Authorization\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=679
Status	New

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	896	896
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
896.                    fprintf (stderr,
```

Improper Resource Access Authorization\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=680
Status	New

	Source	Destination
File	cosmopolitan/maketab.c	cosmopolitan/maketab.c

Line	138	138
Object	fprintf	fprintf

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

```
....
138.              fprintf(stderr, "usage: maketab YTAB_H\n");
```

Improper Resource Access Authorization\Path 41:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=681
Status New

	Source	Destination
File	cosmopolitan/maketab.c	cosmopolitan/maketab.c
Line	142	142
Object	fprintf	fprintf

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

```
....
142.              fprintf(stderr, "maketab can't open %s!\n", argv[1]);
```

Improper Resource Access Authorization\Path 42:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=682
Status New

	Source	Destination
File	cosmopolitan/maketab.c	cosmopolitan/maketab.c
Line	176	176
Object	fprintf	fprintf

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

```
....
176.                    fprintf(stderr, "maketab out of space copying
%s", name);
```

Improper Resource Access Authorization\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=683
Status	New

	Source	Destination
File	cosmopolitan/printf.c	cosmopolitan/printf.c
Line	123	123
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/printf.c
Method int main(int argc, char *argv[]) {

```
....
123.          fprintf(stderr, "%s: %s format [arguments]\n", argv[0],
argv[0]);
```

Improper Resource Access Authorization\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=684
Status	New

	Source	Destination
File	cosmopolitan/process.c	cosmopolitan/process.c
Line	129	129
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/process.c
Method process(void)

```
....
129.                                (void)fprintf(outfile, "%s", cp->t);
```

Improper Resource Access Authorization\Path 45:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=685
Status	New

	Source	Destination
File	cosmopolitan/process.c	cosmopolitan/process.c
Line	161	161
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/process.c
Method process(void)

```
....
161.                            (void)fprintf(outfile, "%s", cp->t);
```

Improper Resource Access Authorization\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=686
Status	New

	Source	Destination
File	cosmopolitan/process.c	cosmopolitan/process.c
Line	259	259
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/process.c
Method process(void)

```
....
259.                            (void)fprintf(outfile, "%lu\n", linenum);
```

Improper Resource Access Authorization\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=687
Status	New

	Source	Destination
File	cosmopolitan/process.c	cosmopolitan/process.c

Line	615	615
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/process.c
Method lputs(char *s, size_t len)

```
....
615.                            fprintf(outfile, "\\\n");
```

Improper Resource Access Authorization\Path 48:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=688
Status New

	Source	Destination
File	cosmopolitan/process.c	cosmopolitan/process.c
Line	622	622
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/process.c
Method lputs(char *s, size_t len)

```
....
622.                            fprintf(outfile, "\\\n");
```

Improper Resource Access Authorization\Path 49:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=689
Status New

	Source	Destination
File	cosmopolitan/process.c	cosmopolitan/process.c
Line	630	630
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/process.c
Method lputs(char *s, size_t len)

```
....
630.                            fprintf(outfile, "\\\n");
```

Improper Resource Access Authorization\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=690
Status	New

	Source	Destination
File	cosmopolitan/process.c	cosmopolitan/process.c
Line	633	633
Object	fprintf	fprintf

Code Snippet
File Name cosmopolitan/process.c
Method lputs(char *s, size_t len)

```
....
633.                    fprintf(outfile, "\\%c", "\\abfrtv"[p -
escapes]);
```

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=1050080&projectid=50070&pathid=467
Status	New

The compress method calls the remove function, at line 1045 of cosmopolitan/bzip2.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	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	1115	1115
Object	remove	remove

Code Snippet
File Name cosmopolitan/bzip2.c
Method void compress (Char *name)

```
....
1115.         remove(outName);
```

Unchecked Return Value\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=468
Status	New

The uncompress method calls the remove function, at line 1226 of cosmopolitan/bzip2.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	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	1301	1301
Object	remove	remove

Code Snippet
File Name cosmopolitan/bzip2.c
Method void uncompress (Char *name)

```
....
1301.         remove(outName);
```

Unchecked Return Value\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=469
Status	New

The main method calls the snprintf function, at line 478 of cosmopolitan/qjsc.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	cosmopolitan/qjsc.c	cosmopolitan/qjsc.c
Line	609	609
Object	snprintf	snprintf

Code Snippet

File Name	cosmopolitan/qjsc.c
Method	int main(int argc, char **argv)

```
....
609.          snprintf(cfilename, 1024, "/tmp/out%d.c", getpid());
```

Unchecked Return Value\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=470
Status	New

The output_executable method calls the snprintf function, at line 398 of cosmopolitan/qjsc.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	cosmopolitan/qjsc.c	cosmopolitan/qjsc.c
Line	416	416
Object	snprintf	snprintf

Code Snippet	
File Name	cosmopolitan/qjsc.c
Method	static int output_executable(const char *out_filename, const char *cfilename,

```
....
416.      snprintf(buf, sizeof(buf), "%s/quickjs.h", exe_dir);
```

Unchecked Return Value\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=471
Status	New

The output_executable method calls the snprintf function, at line 398 of cosmopolitan/qjsc.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	cosmopolitan/qjsc.c	cosmopolitan/qjsc.c
Line	421	421
Object	snprintf	snprintf

Code Snippet	
File Name	cosmopolitan/qjsc.c

Method	static int output_executable(const char *out_filename, const char *cfilename,

```
....
421.          snprintf(inc_dir, sizeof(inc_dir), "%s/include/quickjs",
CONFIG_PREFIX);
```

Unchecked Return Value\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=472
Status	New

The output_executable method calls the snprintf function, at line 398 of cosmopolitan/qjsc.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	cosmopolitan/qjsc.c	cosmopolitan/qjsc.c
Line	422	422
Object	snprintf	snprintf

Code Snippet	
File Name	cosmopolitan/qjsc.c
Method	static int output_executable(const char *out_filename, const char *cfilename,

```
....
422.          snprintf(lib_dir, sizeof(lib_dir), "%s/lib/quickjs",
CONFIG_PREFIX);
```

Unchecked Return Value\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=473
Status	New

The output_executable method calls the snprintf function, at line 398 of cosmopolitan/qjsc.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	cosmopolitan/qjsc.c	cosmopolitan/qjsc.c
Line	446	446
Object	snprintf	snprintf

Code Snippet

File Name	cosmopolitan/qjsc.c
Method	static int output_executable(const char *out_filename, const char *cfilename,

```
....
446.      snprintf(libjsname, sizeof(libjsname), "%s/libquickjs%s%s.a",
```

Unchecked Return Value\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=474
Status	New

The run_test method calls the snprintf function, at line 1566 of cosmopolitan/run-test262.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	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1590	1590
Object	snprintf	snprintf

Code Snippet	
File Name	cosmopolitan/run-test262.c
Method	int run_test(const char *filename, int index)

```
....
1590.                   snprintf(harnessbuf, sizeof(harnessbuf),
"%.*s%s",
```

Unchecked Return Value\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=475
Status	New

The run_test method calls the snprintf function, at line 1566 of cosmopolitan/run-test262.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	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1677	1677
Object	snprintf	snprintf

Code Snippet

File Name	cosmopolitan/run-test262.c
Method	int run_test(const char *filename, int index)

```
....
1677.                    snprintf(harnessbuf, sizeof(harnessbuf),
"%.*s%s",
```

Unchecked Return Value\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=476
Status	New

The compile_flags method calls the wfile function, at line 729 of cosmopolitan/compile.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	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	794	794
Object	wfile	wfile

Code Snippet	
File Name	cosmopolitan/compile.c
Method	compile_flags(char *p, struct s_subst *s)

```
....
794.                    s->wfile = strdup(wfile);
```

Unchecked Return Value\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=477
Status	New

The BENCH method calls the a function, at line 105 of cosmopolitan/djbsort_test.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	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	107	107
Object	a	a

Code Snippet

File Name	cosmopolitan/djbsort_test.c
Method	BENCH(djbsort, bench) {

```
....
107.    a = _gc(memalign(32, n * 4));
```

Unchecked Return Value\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=478
Status	New

The TEST method calls the a function, at line 54 of cosmopolitan/djbsort_test.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	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	57	57
Object	a	a

Code Snippet	
File Name	cosmopolitan/djbsort_test.c
Method	TEST(djbsort, test4) {

```
....
57.    a = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Unchecked Return Value\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=479
Status	New

The TEST method calls the b function, at line 54 of cosmopolitan/djbsort_test.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	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	58	58
Object	b	b

Code Snippet	
File Name	cosmopolitan/djbsort_test.c
Method	TEST(djbsort, test4) {

```
....
58.     b = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Unchecked Return Value\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=480
Status	New

The TEST method calls the c function, at line 54 of cosmopolitan/djbsort_test.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	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	59	59
Object	c	c

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test4) {

```
....
59.     c = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Unchecked Return Value\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=481
Status	New

The TEST method calls the a function, at line 68 of cosmopolitan/djbsort_test.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	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	85	85
Object	a	a

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test64) {

```
....
85.     a = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Unchecked Return Value\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=482
Status	New

The TEST method calls the b function, at line 68 of cosmopolitan/djbsort_test.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	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	86	86
Object	b	b

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test64) {

```
....
86.     b = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Unchecked Return Value\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=483
Status	New

The TEST method calls the c function, at line 68 of cosmopolitan/djbsort_test.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	cosmopolitan/djbsort_test.c	cosmopolitan/djbsort_test.c
Line	87	87
Object	c	c

Code Snippet
File Name cosmopolitan/djbsort_test.c
Method TEST(djbsort, test64) {

```
....
87.     c = memcpy(_gc(malloc(n * 4)), kA, n * 4);
```

Unchecked Return Value\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=484
Status	New

The namelist_add method calls the short_name function, at line 93 of cosmopolitan/qjsc.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	cosmopolitan/qjsc.c	cosmopolitan/qjsc.c
Line	108	108
Object	short_name	short_name

Code Snippet
File Name cosmopolitan/qjsc.c
Method void namelist_add(namelist_t *lp, const char *name, const char *short_name,

```
....
108.          e->short_name = strdup(short_name);
```

Unchecked Return Value\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=485
Status	New

The *str_append method calls the res function, at line 144 of cosmopolitan/run-test262.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	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	151	151
Object	res	res

Code Snippet
File Name cosmopolitan/run-test262.c
Method char *str_append(char **pp, const char *sep, const char *str) {

```
....
151.      res = malloc(len + strlen(str) + 1);
```

Unchecked Return Value\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=486
Status	New

The js_agent_start method calls the agent function, at line 541 of cosmopolitan/run-test262.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	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	553	553
Object	agent	agent

Code Snippet
File Name cosmopolitan/run-test262.c
Method static JSValue js_agent_start(JSContext *ctx, JSValue this_val,

```
....
553.      agent = malloc(sizeof(*agent));
```

Unchecked Return Value\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=487
Status	New

The js_agent_start method calls the script function, at line 541 of cosmopolitan/run-test262.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	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	557	557
Object	script	script

Code Snippet
File Name cosmopolitan/run-test262.c
Method static JSValue js_agent_start(JSContext *ctx, JSValue this_val,

```
....
557.      agent->script = strdup(script);
```

Unchecked Return Value\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=488
Status	New

The js_agent_report method calls the rep function, at line 694 of cosmopolitan/run-test262.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	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	703	703
Object	rep	rep

Code Snippet
File Name cosmopolitan/run-test262.c
Method static JSValue js_agent_report(JSContext *ctx, JSValue this_val,

```
....
703.      rep = malloc(sizeof(*rep));
```

Unchecked Return Value\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=489
Status	New

The js_agent_report method calls the str function, at line 694 of cosmopolitan/run-test262.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	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	704	704
Object	str	str

Code Snippet
File Name cosmopolitan/run-test262.c
Method static JSValue js_agent_report(JSContext *ctx, JSValue this_val,

```
....
704.      rep->str = strdup(str);
```

Unchecked Return Value\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=490
Status	New

The load_config method calls the base_name function, at line 919 of cosmopolitan/run-test262.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	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	938	938
Object	base_name	base_name

Code Snippet
File Name cosmopolitan/run-test262.c
Method void load_config(const char *filename)

```
....
938.      base_name = strdup("");
```

Unchecked Return Value\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=491
Status	New

The *extract_desc method calls the desc function, at line 1379 of cosmopolitan/run-test262.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	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1397	1397
Object	desc	desc

Code Snippet
File Name cosmopolitan/run-test262.c
Method char *extract_desc(const char *buf, char style)

```
....
1397.              desc = malloc(len + 1);
```

Unchecked Return Value\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=492
Status	New

The update_stats method calls the stats_min_filename function, at line 1461 of cosmopolitan/run-test262.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	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1466	1466
Object	stats_min_filename	stats_min_filename

Code Snippet
File Name cosmopolitan/run-test262.c
Method void update_stats(JSRuntime *rt, const char *filename) {

```
....
1466.          stats_min_filename = strdup(filename);
```

Unchecked Return Value\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=493
Status	New

The update_stats method calls the stats_max_filename function, at line 1461 of cosmopolitan/run-test262.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	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1467	1467
Object	stats_max_filename	stats_max_filename

Code Snippet
File Name cosmopolitan/run-test262.c
Method void update_stats(JSRuntime *rt, const char *filename) {

```
....
1467.          stats_max_filename = strdup(filename);
```

Unchecked Return Value\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=494
Status	New

The update_stats method calls the stats_max_filename function, at line 1461 of cosmopolitan/run-test262.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	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1472	1472
Object	stats_max_filename	stats_max_filename

Code Snippet
File Name cosmopolitan/run-test262.c
Method void update_stats(JSRuntime *rt, const char *filename) {

```
....
1472.              stats_max_filename = strdup(filename);
```

Unchecked Return Value\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=495
Status	New

The update_stats method calls the stats_min_filename function, at line 1461 of cosmopolitan/run-test262.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	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1477	1477
Object	stats_min_filename	stats_min_filename

Code Snippet
File Name cosmopolitan/run-test262.c
Method void update_stats(JSRuntime *rt, const char *filename) {

```
....
1477.               stats_min_filename = strdup(filename);
```

Unchecked Return Value\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=496
Status	New

The compressStream method calls the ibuf function, at line 241 of cosmopolitan/bzip2.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	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	244	244
Object	ibuf	ibuf

Code Snippet
File Name cosmopolitan/bzip2.c
Method void compressStream (FILE *stream, FILE *zStream)

```
....
244.     UChar   *ibuf = gc(malloc(5000));
```

Unchecked Return Value\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=497
Status	New

The uncompressStream method calls the obuf function, at line 345 of cosmopolitan/bzip2.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	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	349	349
Object	obuf	obuf

Code Snippet
File Name cosmopolitan/bzip2.c
Method Bool uncompressStream (FILE *zStream, FILE *stream)

```
....
349.     UChar    *obuf = gc(malloc(5000));
```

Unchecked Return Value\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=498
Status	New

The uncompressStream method calls the unused function, at line 345 of cosmopolitan/bzip2.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	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	350	350
Object	unused	unused

Code Snippet
File Name cosmopolitan/bzip2.c
Method Bool uncompressStream (FILE *zStream, FILE *stream)

```
....
350.     UChar    *unused = gc(malloc(BZ_MAX_UNUSED));
```

Unchecked Return Value\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=499
Status	New

The testStream method calls the obuf function, at line 466 of cosmopolitan/bzip2.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	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	470	470
Object	obuf	obuf

Code Snippet
File Name cosmopolitan/bzip2.c
Method Bool testStream (FILE *zStream)

```
....
470.     UChar    *obuf = gc(malloc(5000));
```

Unchecked Return Value\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=500
Status	New

The testStream method calls the unused function, at line 466 of cosmopolitan/bzip2.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	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	471	471
Object	unused	unused

Code Snippet
File Name cosmopolitan/bzip2.c
Method Bool testStream (FILE *zStream)

```
....
471.     UChar    *unused = gc(malloc(BZ_MAX_UNUSED));
```

Unchecked Return Value\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=501
Status	New

The default_bzalloc method calls the v function, at line 105 of cosmopolitan/bzlib.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	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	107	107
Object	v	v

Code Snippet
File Name cosmopolitan/bzlib.c
Method void* default_bzalloc (void* opaque, Int32 items, Int32 size)

```
....
107.     void* v = malloc ( items * size );
```

Unchecked Return Value\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=502
Status	New

The compile_tr method calls the old function, at line 813 of cosmopolitan/compile.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	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	818	818
Object	old	old

Code Snippet
File Name cosmopolitan/compile.c
Method compile_tr(char *p, struct s_tr **py)

```
....
818.         char *old = gc(malloc(_POSIX2_LINE_MAX + 1));
```

Unchecked Return Value\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=503
Status	New

The compile_tr method calls the neW function, at line 813 of cosmopolitan/compile.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	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	819	819
Object	neW	neW

Code Snippet
File Name cosmopolitan/compile.c
Method compile_tr(char *p, struct s_tr **py)

```
....
819.        char *new = gc(malloc(_POSIX2_LINE_MAX + 1));
```

Unchecked Return Value\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=504
Status	New

The main method calls the cfilename function, at line 478 of cosmopolitan/qjsc.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	cosmopolitan/qjsc.c	cosmopolitan/qjsc.c
Line	482	482
Object	cfilename	cfilename

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

```
....
482.      char *cfilename = _gc(malloc(1024));
```

Unchecked Return Value\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=505
Status	New

The *strdup_len method calls the p function, at line 132 of cosmopolitan/run-test262.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	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	134	134
Object	p	p

Code Snippet
File Name cosmopolitan/run-test262.c
Method char *strdup_len(const char *str, int len)

```
....
134.      char *p = malloc(len + 1);
```

Unchecked Return Value\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=506
Status	New

The idxFindIndexes method calls the hIdx function, at line 1111 of cosmopolitan/sqlite3expert.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	cosmopolitan/sqlite3expert.c	cosmopolitan/sqlite3expert.c
Line	1119	1119
Object	hIdx	hIdx

Code Snippet
File Name cosmopolitan/sqlite3expert.c
Method int idxFindIndexes(

```
....
1119.    IdxHash *hIdx = malloc(sizeof(IdxHash));
```

Unchecked Return Value\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=507
Status	New

The stbi_write_hdr_core method calls the scratch function, at line 1002 of cosmopolitan/stb_image_write.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	cosmopolitan/stb_image_write.c	cosmopolitan/stb_image_write.c
Line	1009	1009
Object	scratch	scratch

Code Snippet
File Name cosmopolitan/stb_image_write.c
Method static int stbi_write_hdr_core(stbi__write_context *s, int x, int y, int comp,

```
....
1009.      unsigned char *scratch = malloc(x * 4);
```

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=1050080&projectid=50070&pathid=528
Status	New

The variable declared in null at cosmopolitan/ssl_srv.c in line 3250 is not initialized when it is used by curve at cosmopolitan/ssl_srv.c in line 3250.

	Source	Destination
File	cosmopolitan/ssl_srv.c	cosmopolitan/ssl_srv.c
Line	3368	3384
Object	null	curve

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

```
....
3368.          const mbedtls_ecp_curve_info **curve = NULL;
....
3384.          ssl->curve = *curve;
```

NULL Pointer Dereference\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=529
Status	New

The variable declared in 0 at cosmopolitan/fts3_tokenizer.c in line 376 is not initialized when it is used by Pointer at cosmopolitan/fts3_tokenizer.c in line 376.

	Source	Destination
File	cosmopolitan/fts3_tokenizer.c	cosmopolitan/fts3_tokenizer.c

Line	385	396
Object	0	Pointer

Code Snippet
File Name cosmopolitan/fts3_tokenizer.c
Method int queryTokenizer(

```
....
385.     *pp = 0;
....
396.         memcpy((void *)pp, sqlite3_column_blob(pStmt, 0),
sizeof(*pp));
```

NULL Pointer Dereference\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=530
Status	New

The variable declared in 0 at cosmopolitan/fts3_tokenizer.c in line 376 is not initialized when it is used by Pointer at cosmopolitan/fts3_tokenizer.c in line 376.

	Source	Destination
File	cosmopolitan/fts3_tokenizer.c	cosmopolitan/fts3_tokenizer.c
Line	385	394
Object	0	Pointer

Code Snippet
File Name cosmopolitan/fts3_tokenizer.c
Method int queryTokenizer(

```
....
385.     *pp = 0;
....
394.        && sqlite3_column_bytes(pStmt, 0)==sizeof(*pp)
```

NULL Pointer Dereference\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=531
Status	New

The variable declared in 0 at cosmopolitan/json.c in line 2505 is not initialized when it is used by eU at cosmopolitan/json.c in line 2505.

	Source	Destination

File	cosmopolitan/json.c	cosmopolitan/json.c
Line	2538	2566
Object	0	eU

Code Snippet
File Name cosmopolitan/json.c
Method static int jsonEachFilter(

```
....
2538.       JsonNode *pNode = 0;
....
2566.         assert( pNode->eU==0 );
```

NULL Pointer Dereference\Path 5:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=532
Status New

The variable declared in 0 at cosmopolitan/lgc.c in line 146 is not initialized when it is used by gray at cosmopolitan/lgc.c in line 673.

	Source	Destination
File	cosmopolitan/lgc.c	cosmopolitan/lgc.c
Line	158	676
Object	0	gray

Code Snippet
File Name cosmopolitan/lgc.c
Method static GCObject **getgclist (GCObject *o) {

```
....
158.       default: lua_assert(0); return 0;
```

File Name cosmopolitan/lgc.c
Method static lu_mem propagatemark (global_State *g) {

```
....
676.     g->gray = *getgclist(o);  /* remove from 'gray' list */
```

NULL Pointer Dereference\Path 6:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=500

	70&pathid=533
Status	New

The variable declared in 0 at cosmopolitan/main.c in line 748 is not initialized when it is used by bMalloced at cosmopolitan/main.c in line 748.

	Source	Destination
File	cosmopolitan/main.c	cosmopolitan/main.c
Line	825	825
Object	0	bMalloced

Code Snippet
File Name cosmopolitan/main.c
Method static int setupLookaside(sqlite3 *db, void *pBuf, int sz, int cnt){

```
....
825.      db->lookaside.bMalloced = pBuf==0 ?1:0;
```

NULL Pointer Dereference\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=534
Status	New

The variable declared in 0 at cosmopolitan/main.c in line 748 is not initialized when it is used by lookaside at cosmopolitan/main.c in line 748.

	Source	Destination
File	cosmopolitan/main.c	cosmopolitan/main.c
Line	825	808
Object	0	lookaside

Code Snippet
File Name cosmopolitan/main.c
Method static int setupLookaside(sqlite3 *db, void *pBuf, int sz, int cnt){

```
....
825.      db->lookaside.bMalloced = pBuf==0 ?1:0;
....
808.        p->pNext = db->lookaside.pInit;
```

NULL Pointer Dereference\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=535

Status	New

The variable declared in 0 at cosmopolitan/main.c in line 748 is not initialized when it is used by lookaside at cosmopolitan/main.c in line 748.

	Source	Destination
File	cosmopolitan/main.c	cosmopolitan/main.c
Line	825	817
Object	0	lookaside

Code Snippet
File Name cosmopolitan/main.c
Method static int setupLookaside(sqlite3 *db, void *pBuf, int sz, int cnt){

```
....
825.      db->lookaside.bMalloced = pBuf==0 ?1:0;
....
817.        p->pNext = db->lookaside.pSmallInit;
```

NULL Pointer Dereference\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=536
Status	New

The variable declared in 0 at cosmopolitan/main.c in line 748 is not initialized when it is used by lookaside at cosmopolitan/main.c in line 748.

	Source	Destination
File	cosmopolitan/main.c	cosmopolitan/main.c
Line	825	840
Object	0	lookaside

Code Snippet
File Name cosmopolitan/main.c
Method static int setupLookaside(sqlite3 *db, void *pBuf, int sz, int cnt){

```
....
825.      db->lookaside.bMalloced = pBuf==0 ?1:0;
....
840.    db->lookaside.pTrueEnd = db->lookaside.pEnd;
```

NULL Pointer Dereference\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=537

Status	New

The variable declared in 0 at cosmopolitan/main.c in line 748 is not initialized when it is used by lookaside at cosmopolitan/main.c in line 748.

	Source	Destination
File	cosmopolitan/main.c	cosmopolitan/main.c
Line	825	763
Object	0	lookaside

Code Snippet
File Name cosmopolitan/main.c
Method static int setupLookaside(sqlite3 *db, void *pBuf, int sz, int cnt){

```
....
825.      db->lookaside.bMalloced = pBuf==0 ?1:0;
....
763.      sqlite3_free(db->lookaside.pStart);
```

NULL Pointer Dereference\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=538
Status	New

The variable declared in 0 at cosmopolitan/main.c in line 748 is not initialized when it is used by lookaside at cosmopolitan/main.c in line 748.

	Source	Destination
File	cosmopolitan/main.c	cosmopolitan/main.c
Line	825	762
Object	0	lookaside

Code Snippet
File Name cosmopolitan/main.c
Method static int setupLookaside(sqlite3 *db, void *pBuf, int sz, int cnt){

```
....
825.      db->lookaside.bMalloced = pBuf==0 ?1:0;
....
762.    if( db->lookaside.bMalloced ){
```

NULL Pointer Dereference\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=539

Status	New

The variable declared in 0 at cosmopolitan/main.c in line 2138 is not initialized when it is used by mTrace at cosmopolitan/main.c in line 2138.

	Source	Destination
File	cosmopolitan/main.c	cosmopolitan/main.c
Line	2149	2149
Object	0	mTrace

Code Snippet
File Name cosmopolitan/main.c
Method void *sqlite3_trace(sqlite3 *db, void(*xTrace)(void*,const char*), void *pArg){

```
....
2149.    db->mTrace = xTrace ? SQLITE_TRACE_LEGACY : 0;
```

NULL Pointer Dereference\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=540
Status	New

The variable declared in 0 at cosmopolitan/main.c in line 3163 is not initialized when it is used by nVdbeActive at cosmopolitan/main.c in line 2696.

	Source	Destination
File	cosmopolitan/main.c	cosmopolitan/main.c
Line	3235	2729
Object	0	nVdbeActive

Code Snippet
File Name cosmopolitan/main.c
Method static int openDatabase(

```
....
3235.        db = 0;
```

File Name cosmopolitan/main.c
Method static int createCollation(

```
....
2729.      if( db->nVdbeActive ){
```

NULL Pointer Dereference\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=541
Status	New

The variable declared in 0 at cosmopolitan/sqlite3rbu.c in line 2115 is not initialized when it is used by zTbl at cosmopolitan/sqlite3rbu.c in line 1326.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	2119	1337
Object	0	zTbl

Code Snippet

File Name cosmopolitan/sqlite3rbu.c

Method static void rbuCreateImposterTable(sqlite3rbu *p, RbuObjIter *pIter){

```
....
2119.       char *zSql = 0;
```

File Name cosmopolitan/sqlite3rbu.c

Method static int rbuObjIterCacheTableInfo(sqlite3rbu *p, RbuObjIter *pIter){

```
....
1337.       rbuTableType(p, pIter->zTbl, &pIter->eType, &iTnum, &pIter-
>iPkTnum);
```

NULL Pointer Dereference\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=542
Status	New

The variable declared in 0 at cosmopolitan/sqlite3rbu.c in line 2115 is not initialized when it is used by zIdx at cosmopolitan/sqlite3rbu.c in line 1326.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	2119	1343
Object	0	zIdx

Code Snippet

File Name cosmopolitan/sqlite3rbu.c

Method static void rbuCreateImposterTable(sqlite3rbu *p, RbuObjIter *pIter){

```
....
2119.       char *zSql = 0;
```

File Name cosmopolitan/sqlite3rbu.c

Method static int rbuObjIterCacheTableInfo(sqlite3rbu *p, RbuObjIter *pIter){

```
....
1343.       if( pIter->zIdx==0 ) pIter->iTnum = iTnum;
```

NULL Pointer Dereference\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=543
Status	New

The variable declared in 0 at cosmopolitan/sqlite3rbu.c in line 2115 is not initialized when it is used by iPkTnum at cosmopolitan/sqlite3rbu.c in line 1326.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	2119	1337
Object	0	iPkTnum

Code Snippet

File Name cosmopolitan/sqlite3rbu.c

Method static void rbuCreateImposterTable(sqlite3rbu *p, RbuObjIter *pIter){

```
....
2119.       char *zSql = 0;
```

File Name cosmopolitan/sqlite3rbu.c

Method static int rbuObjIterCacheTableInfo(sqlite3rbu *p, RbuObjIter *pIter){

```
....
1337.       rbuTableType(p, pIter->zTbl, &pIter->eType, &iTnum, &pIter-
>iPkTnum);
```

NULL Pointer Dereference\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=544
Status	New

The variable declared in 0 at cosmopolitan/sqlite3rbu.c in line 2115 is not initialized when it is used by azTblCol at cosmopolitan/sqlite3rbu.c in line 1326.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	2119	1327
Object	0	azTblCol

Code Snippet
File Name cosmopolitan/sqlite3rbu.c
Method static void rbuCreateImposterTable(sqlite3rbu *p, RbuObjIter *pIter){

```
....
2119.       char *zSql = 0;
```

File Name cosmopolitan/sqlite3rbu.c
Method static int rbuObjIterCacheTableInfo(sqlite3rbu *p, RbuObjIter *pIter){

```
....
1327.     if( pIter->azTblCol==0 ){
```

NULL Pointer Dereference\Path 18:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=545
Status New

The variable declared in 0 at cosmopolitan/sqlite3rbu.c in line 3150 is not initialized when it is used by pMethods at cosmopolitan/sqlite3rbu.c in line 3150.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	3152	3156
Object	0	pMethods

Code Snippet
File Name cosmopolitan/sqlite3rbu.c
Method static int rbuLockDatabase(sqlite3 *db){

```
....
3152.     sqlite3_file *fd = 0;
....
3156.       rc = fd->pMethods->xLock(fd, SQLITE_LOCK_SHARED);
```

NULL Pointer Dereference\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=546
Status	New

The variable declared in 0 at cosmopolitan/sqlite3rbu.c in line 3150 is not initialized when it is used by pMethods at cosmopolitan/sqlite3rbu.c in line 3150.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	3152	3155
Object	0	pMethods

Code Snippet
File Name cosmopolitan/sqlite3rbu.c
Method static int rbuLockDatabase(sqlite3 *db){

```
....
3152.     sqlite3_file *fd = 0;
....
3155.     if( fd->pMethods ){
```

NULL Pointer Dereference\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=547
Status	New

The variable declared in 0 at cosmopolitan/sqlite3rbu.c in line 3970 is not initialized when it is used by zTbl at cosmopolitan/sqlite3rbu.c in line 3798.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	3982	3800
Object	0	zTbl

Code Snippet
File Name cosmopolitan/sqlite3rbu.c
Method static sqlite3rbu *openRbuHandle(

```
....
3982.       RbuState *pState = 0;
```

File Name	cosmopolitan/sqlite3rbu.c
Method	static void rbuSetupOal(sqlite3rbu *p, RbuState *pState){

```
....
3800.    if( pState->zTbl ){
```

NULL Pointer Dereference\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=548
Status	New

The variable declared in 0 at cosmopolitan/vdbesort.c in line 2500 is not initialized when it is used by pIncr at cosmopolitan/vdbesort.c in line 2500.

	Source	Destination
File	cosmopolitan/vdbesort.c	cosmopolitan/vdbesort.c
Line	2519	2530
Object	0	pIncr

Code Snippet

File Name	cosmopolitan/vdbesort.c
Method	static int vdbeSorterSetupMerge(VdbeSorter *pSorter){

```
....
2519.        PmaReader *pReadr = 0;
....
2530.            vdbeIncrMergerSetThreads(pReadr->pIncr);
```

NULL Pointer Dereference\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=549
Status	New

The variable declared in pNew at cosmopolitan/sqlite3expert.c in line 676 is not initialized when it is used by nCol at cosmopolitan/sqlite3expert.c in line 676.

	Source	Destination
File	cosmopolitan/sqlite3expert.c	cosmopolitan/sqlite3expert.c
Line	686	711
Object	pNew	nCol

Code Snippet

File Name	cosmopolitan/sqlite3expert.c
Method	static int idxGetTableInfo(

```
....
686.    IdxTable *pNew = 0;
....
711.      pNew->nCol = nCol;
```

NULL Pointer Dereference\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=550
Status	New

The variable declared in pNew at cosmopolitan/sqlite3expert.c in line 676 is not initialized when it is used by aCol at cosmopolitan/sqlite3expert.c in line 676.

	Source	Destination
File	cosmopolitan/sqlite3expert.c	cosmopolitan/sqlite3expert.c
Line	686	710
Object	pNew	aCol

Code Snippet

File Name	cosmopolitan/sqlite3expert.c
Method	static int idxGetTableInfo(

```
....
686.    IdxTable *pNew = 0;
....
710.      pNew->aCol = (IdxColumn*)&pNew[1];
```

NULL Pointer Dereference\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=551
Status	New

The variable declared in pNew at cosmopolitan/sqlite3expert.c in line 676 is not initialized when it is used by zColl at cosmopolitan/sqlite3expert.c in line 676.

	Source	Destination
File	cosmopolitan/sqlite3expert.c	cosmopolitan/sqlite3expert.c
Line	686	729
Object	pNew	zColl

Code Snippet

File Name	cosmopolitan/sqlite3expert.c
Method	static int idxGetTableInfo(

```
....
686.    IdxTable *pNew = 0;
....
729.        pNew->aCol[nCol].zColl = pCsr;
```

NULL Pointer Dereference\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=552
Status	New

The variable declared in pNew at cosmopolitan/sqlite3expert.c in line 676 is not initialized when it is used by iPk at cosmopolitan/sqlite3expert.c in line 676.

	Source	Destination
File	cosmopolitan/sqlite3expert.c	cosmopolitan/sqlite3expert.c
Line	686	720
Object	pNew	iPk

Code Snippet

File Name	cosmopolitan/sqlite3expert.c
Method	static int idxGetTableInfo(

```
....
686.    IdxTable *pNew = 0;
....
720.      pNew->aCol[nCol].iPk = (sqlite3_column_int(p1, 5)==1 &&
nPk==1);
```

NULL Pointer Dereference\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=553
Status	New

The variable declared in pNew at cosmopolitan/sqlite3expert.c in line 676 is not initialized when it is used by zName at cosmopolitan/sqlite3expert.c in line 676.

	Source	Destination
File	cosmopolitan/sqlite3expert.c	cosmopolitan/sqlite3expert.c
Line	686	742
Object	pNew	zName

Code Snippet
File Name cosmopolitan/sqlite3expert.c
Method static int idxGetTableInfo(

```
....
686.    IdxTable *pNew = 0;
....
742.      pNew->zName = pCsr;
```

NULL Pointer Dereference\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=554
Status	New

The variable declared in pNew at cosmopolitan/sqlite3expert.c in line 676 is not initialized when it is used by zName at cosmopolitan/sqlite3expert.c in line 676.

	Source	Destination
File	cosmopolitan/sqlite3expert.c	cosmopolitan/sqlite3expert.c
Line	686	719
Object	pNew	zName

Code Snippet
File Name cosmopolitan/sqlite3expert.c
Method static int idxGetTableInfo(

```
....
686.    IdxTable *pNew = 0;
....
719.      pNew->aCol[nCol].zName = pCsr;
```

NULL Pointer Dereference\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=555
Status	New

The variable declared in pNew at cosmopolitan/sqlite3expert.c in line 676 is not initialized when it is used by aCol at cosmopolitan/sqlite3expert.c in line 676.

	Source	Destination
File	cosmopolitan/sqlite3expert.c	cosmopolitan/sqlite3expert.c
Line	686	712
Object	pNew	aCol

Code Snippet
File Name cosmopolitan/sqlite3expert.c
Method static int idxGetTableInfo(

```
....
686.    IdxTable *pNew = 0;
....
712.      pCsr = (char*)&pNew->aCol[nCol];
```

NULL Pointer Dereference\Path 29:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=556
Status New

The variable declared in pNew at cosmopolitan/sqlite3expert.c in line 676 is not initialized when it is used by zName at cosmopolitan/sqlite3expert.c in line 676.

	Source	Destination
File	cosmopolitan/sqlite3expert.c	cosmopolitan/sqlite3expert.c
Line	686	743
Object	pNew	zName

Code Snippet
File Name cosmopolitan/sqlite3expert.c
Method static int idxGetTableInfo(

```
....
686.    IdxTable *pNew = 0;
....
743.      memcpy(pNew->zName, zTab, nTab+1);
```

NULL Pointer Dereference\Path 30:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=557
Status New

The variable declared in fd at cosmopolitan/sqlite3rbu.c in line 3150 is not initialized when it is used by pMethods at cosmopolitan/sqlite3rbu.c in line 3150.

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	3152	3158
Object	fd	pMethods

Code Snippet	
File Name	cosmopolitan/sqlite3rbu.c
Method	static int rbuLockDatabase(sqlite3 *db){

```
....
3152.    sqlite3_file *fd = 0;
....
3158.        rc = fd->pMethods->xLock(fd, SQLITE_LOCK_EXCLUSIVE);
```

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=1050080&projectid=50070&pathid=811
Status	New

The fileExists method in cosmopolitan/bzip2.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	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	854	854
Object	fopen	fopen

Code Snippet	
File Name	cosmopolitan/bzip2.c
Method	Bool fileExists (Char* name)

```
....
854.     FILE *tmp   = fopen ( name, "rb" );
```

TOCTOU\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=812
Status	New

The compress method in cosmopolitan/bzip2.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	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	1154	1154

Object	fopen	fopen

Code Snippet
File Name cosmopolitan/bzip2.c
Method void compress (Char *name)

```
....
1154.           inStr = fopen ( inName, "rb" );
```

TOCTOU\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=813
Status	New

The compress method in cosmopolitan/bzip2.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	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	1175	1175
Object	fopen	fopen

Code Snippet
File Name cosmopolitan/bzip2.c
Method void compress (Char *name)

```
....
1175.           inStr = fopen ( inName, "rb" );
```

TOCTOU\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=814
Status	New

The uncompress method in cosmopolitan/bzip2.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	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	1340	1340
Object	fopen	fopen

Code Snippet	
File Name	cosmopolitan/bzip2.c
Method	void uncompress (Char *name)

```
....
1340.           inStr = fopen ( inName, "rb" );
```

TOCTOU\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=815
Status	New

The uncompress method in cosmopolitan/bzip2.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	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	1352	1352
Object	fopen	fopen

Code Snippet	
File Name	cosmopolitan/bzip2.c
Method	void uncompress (Char *name)

```
....
1352.           inStr = fopen ( inName, "rb" );
```

TOCTOU\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=816
Status	New

The testf method in cosmopolitan/bzip2.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	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	1482	1482
Object	fopen	fopen

Code Snippet	
File Name	cosmopolitan/bzip2.c
Method	void testf (Char *name)

```
....
1482.           inStr = fopen ( inName, "rb" );
```

TOCTOU\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=817
Status	New

The bzopen_or_bzdopen method in cosmopolitan/bzlib.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	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	1426	1426
Object	fopen	fopen

Code Snippet
File Name cosmopolitan/bzlib.c
Method BZFILE * bzopen_or_bzdopen

```
....
1426.           fp = fopen(path,mode2);
```

TOCTOU\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=818
Status	New

The main method in cosmopolitan/maketab.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	cosmopolitan/maketab.c	cosmopolitan/maketab.c
Line	141	141
Object	fopen	fopen

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

```
....
141.        if ((fp = fopen(argv[1], "r")) == NULL) {
```

TOCTOU\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=819
Status	New

The flush_appends method in cosmopolitan/process.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	cosmopolitan/process.c	cosmopolitan/process.c
Line	558	558
Object	fopen	fopen

Code Snippet
File Name cosmopolitan/process.c
Method flush_appends(void)

```
....
558.                    if ((f = fopen(appends_[i].s, "r")) == NULL)
```

TOCTOU\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=820
Status	New

The main method in cosmopolitan/qjsc.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	cosmopolitan/qjsc.c	cosmopolitan/qjsc.c
Line	613	613
Object	fopen	fopen

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

```
....
613.      fo = fopen(cfilename, "w");
```

TOCTOU\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=821
Status	New

The *js_load_file method in cosmopolitan/quickjs-libc.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	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	360	360
Object	fopen	fopen

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method uint8_t *js_load_file(JSContext *ctx, size_t *pbuf_len, const char *filename)

```
....
360.      f = fopen(filename, "rb");
```

TOCTOU\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=822
Status	New

The js_std_open method in cosmopolitan/quickjs-libc.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	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	856	856
Object	fopen	fopen

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_std_open(JSContext *ctx, JSValueConst this_val,

```
....
856.      f = fopen(filename, mode);
```

TOCTOU\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=823
Status	New

The main method in cosmopolitan/run-test262.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	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	2029	2029
Object	fopen	fopen

Code Snippet
File Name cosmopolitan/run-test262.c
Method int main(int argc, char **argv)

```
....
2029.              error_out = fopen(error_filename, "w");
```

TOCTOU\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=824
Status	New

The main method in cosmopolitan/run-test262.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	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	2056	2056
Object	fopen	fopen

Code Snippet
File Name cosmopolitan/run-test262.c
Method int main(int argc, char **argv)

```
....
2056.               outfile = fopen(report_filename, "wb");
```

TOCTOU\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=825
Status	New

The namelist_load method in cosmopolitan/run-test262.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	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	321	321
Object	fopen	fopen

Code Snippet
File Name cosmopolitan/run-test262.c
Method void namelist_load(namelist_t *lp, const char *filename)

```
....
321.       f = fopen(filename, "rb");
```

TOCTOU\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=826
Status	New

The load_config method in cosmopolitan/run-test262.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	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	933	933
Object	fopen	fopen

Code Snippet
File Name cosmopolitan/run-test262.c
Method void load_config(const char *filename)

```
....
933.      f = fopen(filename, "rb");
```

TOCTOU\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=827
Status	New

The stbi__start_write_file method in cosmopolitan/stb_image_write.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	cosmopolitan/stb_image_write.c	cosmopolitan/stb_image_write.c
Line	163	163
Object	fopen	fopen

Code Snippet
File Name cosmopolitan/stb_image_write.c
Method static int stbi__start_write_file(stbi__write_context *s,

```
....
163.    FILE *f = fopen(filename, "wb");
```

TOCTOU\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=828
Status	New

The stbi_write_png method in cosmopolitan/stb_image_write_png.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	cosmopolitan/stb_image_write_png.c	cosmopolitan/stb_image_write_png.c
Line	357	357
Object	fopen	fopen

Code Snippet
File Name cosmopolitan/stb_image_write_png.c
Method int stbi_write_png(const char *filename, int x, int y, int comp,

```
....
357.    f = fopen(filename, "wb");
```

TOCTOU\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=829
Status	New

The compile_stream method in cosmopolitan/compile.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	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	311	311
Object	open	open

Code Snippet
File Name cosmopolitan/compile.c
Method compile_stream(struct s_command **link)

```
....
311.                    else if ((cmd->u.fd = open(p,
```

TOCTOU\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=830
Status	New

The compile_flags method in cosmopolitan/compile.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	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	795	795
Object	open	open

Code Snippet
File Name cosmopolitan/compile.c
Method compile_flags(char *p, struct s_subst *s)

```
....
795.                    if (!aflag && (s->wfd = open(wfile,
```

TOCTOU\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=831
Status	New

The process method in cosmopolitan/process.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	cosmopolitan/process.c	cosmopolitan/process.c
Line	228	228
Object	open	open

Code Snippet
File Name cosmopolitan/process.c
Method process(void)

```
....
228.                        if (cp->u.fd == -1 && (cp->u.fd = open(cp->t,
```

TOCTOU\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=832
Status	New

The substitute method in cosmopolitan/process.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	cosmopolitan/process.c	cosmopolitan/process.c
Line	466	466
Object	open	open

Code Snippet
File Name cosmopolitan/process.c
Method substitute(struct s_command *cp)

```
....
466.              if (cp->u.s->wfd == -1 && (cp->u.s->wfd = open(cp-
>u.s->wfile,
```

TOCTOU\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=833
Status	New

The js_os_open method in cosmopolitan/quickjs-libc.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	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	1571	1571
Object	open	open

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_os_open(JSContext *ctx, JSValueConst this_val,

```
....
1571.      ret = js_get_errno(open(filename, flags, mode));
```

TOCTOU\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=834
Status	New

The FixIrregularFds method in cosmopolitan/testmain.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	cosmopolitan/testmain.c	cosmopolitan/testmain.c
Line	113	113
Object	open	open

Code Snippet
File Name cosmopolitan/testmain.c
Method static void FixIrregularFds(void) {

```
....
113.        fd = open("/dev/null", O_RDWR);
```

TOCTOU\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=835
Status	New

The main method in cosmopolitan/zip2.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	cosmopolitan/zip2.c	cosmopolitan/zip2.c
Line	275	275
Object	open	open

Code Snippet
File Name cosmopolitan/zip2.c
Method int main(int argc, char *argv[]) {

```
....
275.    CHECK_NE(-1, (fd = open(argv[1], O_RDONLY)));
```

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=1050080&projectid=50070&pathid=576
Status	New

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	1127	1127
Object	bufN	bufN

Code Snippet

File Name	cosmopolitan/bzlib.c
Method	BZFILE* BZ2_bzReadOpen

```
....
1127.         bzf->buf[bzf->bufN] = *((UChar*)(unused)); bzf->bufN++;
```

Unchecked Array Index\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=577
Status	New

	Source	Destination
File	cosmopolitan/json.c	cosmopolitan/json.c
Line	674	674
Object	j	j

Code Snippet	
File Name	cosmopolitan/json.c
Method	static void jsonReturn(

```
....
674.           zOut[j] = 0;
```

Unchecked Array Index\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=578
Status	New

	Source	Destination
File	cosmopolitan/json.c	cosmopolitan/json.c
Line	963	963
Object	i	i

Code Snippet	
File Name	cosmopolitan/json.c
Method	static void jsonParseFillInParentage(JsonParse *pParse, u32 i, u32 iParent){

```
....
963.     pParse->aUp[i] = iParent;
```

Unchecked Array Index\Path 4:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=579
Status	New

	Source	Destination
File	cosmopolitan/json.c	cosmopolitan/json.c
Line	2473	2473
Object	iCol	iCol

Code Snippet
File Name cosmopolitan/json.c
Method static int jsonEachBestIndex(

```
....
2473.         aIdx[iCol] = i;
```

Unchecked Array Index\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=580
Status	New

	Source	Destination
File	cosmopolitan/lstrlib.c	cosmopolitan/lstrlib.c
Line	1061	1061
Object	n	n

Code Snippet
File Name cosmopolitan/lstrlib.c
Method static lua_Number adddigit (char *buff, int n, lua_Number x) {

```
....
1061.     buff[n] = (d < 10 ? d + '0' : d - 10 + 'a');  /* add to buffer
*/
```

Unchecked Array Index\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=581
Status	New

	Source	Destination

File	cosmopolitan/printf.c	cosmopolitan/printf.c
Line	92	92
Object	j	j

Code Snippet
File Name cosmopolitan/printf.c
Method char *U(char *p) {

```
....
92.           p[j] = 0;
```

Unchecked Array Index\Path 7:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=582
Status New

	Source	Destination
File	cosmopolitan/process.c	cosmopolitan/process.c
Line	675	675
Object	slen	slen

Code Snippet
File Name cosmopolitan/process.c
Method regexec_e(regex_t *preg, const char *string, int eflags, int nomatch,

```
....
675.          buf[slen] = '\0';
```

Unchecked Array Index\Path 8:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=583
Status New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	1840	1840
Object	magic	magic

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_os_setReadHandler(JSContext *ctx, JSValueConst this_val,

```
....
1840.               rh->rw_func[magic] = JS_NULL;
```

Unchecked Array Index\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=584
Status	New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	1861	1861
Object	magic	magic

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_os_setReadHandler(JSContext *ctx, JSValueConst this_val,

```
....
1861.           rh->rw_func[magic] = JS_DupValue(ctx, func);
```

Unchecked Array Index\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=585
Status	New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	2726	2726
Object	key_len	key_len

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static char **build_envp(JSContext *ctx, JSValueConst obj)

```
....
2726.           pair[key_len] = '=';
```

Unchecked Array Index\Path 11:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=586
Status	New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	136	136
Object	len	len

Code Snippet
File Name cosmopolitan/run-test262.c
Method char *strdup_len(const char *str, int len)

```
....
136.      p[len] = '\0';
```

Unchecked Array Index\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=587
Status	New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1399	1399
Object	len	len

Code Snippet
File Name cosmopolitan/run-test262.c
Method char *extract_desc(const char *buf, char style)

```
....
1399.              desc[len] = '\0';
```

Unchecked Array Index\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=588
Status	New

	Source	Destination
File	cosmopolitan/sds.c	cosmopolitan/sds.c
Line	383	383

Object	len	len

Code Snippet
File Name cosmopolitan/sds.c
Method void sdsIncrLen(sds s, int incr) {

```
....
383.        s[len] = '\0';
```

Unchecked Array Index\Path 14:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=589
Status New

	Source	Destination
File	cosmopolitan/sds.c	cosmopolitan/sds.c
Line	700	700
Object	i	i

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

```
....
700.        s[i] = '\0';
```

Unchecked Array Index\Path 15:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=590
Status New

	Source	Destination
File	cosmopolitan/sds.c	cosmopolitan/sds.c
Line	732	732
Object	len	len

Code Snippet
File Name cosmopolitan/sds.c
Method sds sdstrim(sds s, const char *cset) {

```
....
732.        s[len] = '\0';
```

Unchecked Array Index\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=591
Status	New

	Source	Destination
File	cosmopolitan/sqlite3expert.c	cosmopolitan/sqlite3expert.c
Line	381	381
Object	iOut	iOut

Code Snippet
File Name cosmopolitan/sqlite3expert.c
Method static char *expertDequote(const char *zIn){

```
....
381.        zRet[iOut] = '\0';
```

Unchecked Array Index\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=592
Status	New

	Source	Destination
File	cosmopolitan/stb_image_write.c	cosmopolitan/stb_image_write.c
Line	738	738
Object	k	k

Code Snippet
File Name cosmopolitan/stb_image_write.c
Method static int stbi_write_jpg_core(stbi__write_context *s, int width, int height,

```
....
738.        fdtbl_Y[k] = 1 / (YTable[stbiw__jpg_ZigZag[k]] * aasf[row] *
aasf[col]);
```

Unchecked Array Index\Path 18:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=593
Status	New

	Source	Destination
File	cosmopolitan/stb_image_write.c	cosmopolitan/stb_image_write.c
Line	739	739
Object	k	k

Code Snippet
File Name cosmopolitan/stb_image_write.c
Method static int stbi_write_jpg_core(stbi__write_context *s, int width, int height,

```
....
739.         fdtbl_UV[k] = 1 / (UVTable[stbiw__jpg_ZigZag[k]] * aasf[row]
* aasf[col]);
```

Unchecked Array Index\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=594
Status	New

	Source	Destination
File	cosmopolitan/stb_image_write.c	cosmopolitan/stb_image_write.c
Line	830	830
Object	pos	pos

Code Snippet
File Name cosmopolitan/stb_image_write.c
Method static int stbi_write_jpg_core(stbi__write_context *s, int width, int height,

```
....
830.             YDU[pos] = +0.29900f * r + 0.58700f * g + 0.11400f * b
- 128;
```

Unchecked Array Index\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=595
Status	New

	Source	Destination

File	cosmopolitan/stb_image_write.c	cosmopolitan/stb_image_write.c
Line	831	831
Object	pos	pos

Code Snippet
File Name cosmopolitan/stb_image_write.c
Method static int stbi_write_jpg_core(stbi__write_context *s, int width, int height,

```
....
831.                UDU[pos] = -0.16874f * r - 0.33126f * g + 0.50000f *
b;
```

Unchecked Array Index\Path 21:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=596
Status New

	Source	Destination
File	cosmopolitan/stb_image_write.c	cosmopolitan/stb_image_write.c
Line	832	832
Object	pos	pos

Code Snippet
File Name cosmopolitan/stb_image_write.c
Method static int stbi_write_jpg_core(stbi__write_context *s, int width, int height,

```
....
832.                VDU[pos] = +0.50000f * r - 0.41869f * g - 0.08131f *
b;
```

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=1050080&projectid=50070&pathid=788
Status New

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	1154	1154
Object	inStr	inStr

Code Snippet
File Name cosmopolitan/bzip2.c
Method void compress (Char *name)

```
....
1154.           inStr = fopen ( inName, "rb" );
```

Incorrect Permission Assignment For Critical Resources\Path 2:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=789
Status New

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	1175	1175
Object	inStr	inStr

Code Snippet
File Name cosmopolitan/bzip2.c
Method void compress (Char *name)

```
....
1175.           inStr = fopen ( inName, "rb" );
```

Incorrect Permission Assignment For Critical Resources\Path 3:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=790
Status New

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	1340	1340
Object	inStr	inStr

Code Snippet
File Name cosmopolitan/bzip2.c
Method void uncompress (Char *name)

```
....
1340.           inStr = fopen ( inName, "rb" );
```

Incorrect Permission Assignment For Critical Resources\Path 4:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=791
Status New

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	1352	1352
Object	inStr	inStr

Code Snippet
File Name cosmopolitan/bzip2.c
Method void uncompress (Char *name)

```
....
1352.           inStr = fopen ( inName, "rb" );
```

Incorrect Permission Assignment For Critical Resources\Path 5:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=792
Status New

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	1482	1482
Object	inStr	inStr

Code Snippet
File Name cosmopolitan/bzip2.c
Method void testf (Char *name)

```
....
1482.           inStr = fopen ( inName, "rb" );
```

Incorrect Permission Assignment For Critical Resources\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=793
Status	New

	Source	Destination
File	cosmopolitan/bzlib.c	cosmopolitan/bzlib.c
Line	1426	1426
Object	fp	fp

Code Snippet
File Name cosmopolitan/bzlib.c
Method BZFILE * bzopen_or_bzdopen

```
....
1426.           fp = fopen(path,mode2);
```

Incorrect Permission Assignment For Critical Resources\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=794
Status	New

	Source	Destination
File	cosmopolitan/maketab.c	cosmopolitan/maketab.c
Line	141	141
Object	fp	fp

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

```
....
141.        if ((fp = fopen(argv[1], "r")) == NULL) {
```

Incorrect Permission Assignment For Critical Resources\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=795
Status	New

	Source	Destination

File	cosmopolitan/process.c	cosmopolitan/process.c
Line	558	558
Object	f	f

Code Snippet
File Name cosmopolitan/process.c
Method flush_appends(void)

```
....
558.                    if ((f = fopen(appends_[i].s, "r")) == NULL)
```

Incorrect Permission Assignment For Critical Resources\Path 9:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=796
Status New

	Source	Destination
File	cosmopolitan/qjsc.c	cosmopolitan/qjsc.c
Line	613	613
Object	fo	fo

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

```
....
613.      fo = fopen(cfilename, "w");
```

Incorrect Permission Assignment For Critical Resources\Path 10:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=797
Status New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	360	360
Object	f	f

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method uint8_t *js_load_file(JSContext *ctx, size_t *pbuf_len, const char *filename)

```
....
360.      f = fopen(filename, "rb");
```

Incorrect Permission Assignment For Critical Resources\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=798
Status	New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	856	856
Object	f	f

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_std_open(JSContext *ctx, JSValueConst this_val,

```
....
856.      f = fopen(filename, mode);
```

Incorrect Permission Assignment For Critical Resources\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=799
Status	New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	2029	2029
Object	error_out	error_out

Code Snippet
File Name cosmopolitan/run-test262.c
Method int main(int argc, char **argv)

```
....
2029.             error_out = fopen(error_filename, "w");
```

Incorrect Permission Assignment For Critical Resources\Path 13:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=800
Status	New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	2056	2056
Object	outfile	outfile

Code Snippet
File Name cosmopolitan/run-test262.c
Method int main(int argc, char **argv)

```
....
2056.              outfile = fopen(report_filename, "wb");
```

Incorrect Permission Assignment For Critical Resources\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=801
Status	New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	321	321
Object	f	f

Code Snippet
File Name cosmopolitan/run-test262.c
Method void namelist_load(namelist_t *lp, const char *filename)

```
....
321.      f = fopen(filename, "rb");
```

Incorrect Permission Assignment For Critical Resources\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=802
Status	New

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	933	933

Object	f	f

Code Snippet
File Name cosmopolitan/run-test262.c
Method void load_config(const char *filename)

```
....
933.      f = fopen(filename, "rb");
```

Incorrect Permission Assignment For Critical Resources\Path 16:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=803
Status New

	Source	Destination
File	cosmopolitan/stb_image_write_png.c	cosmopolitan/stb_image_write_png.c
Line	357	357
Object	f	f

Code Snippet
File Name cosmopolitan/stb_image_write_png.c
Method int stbi_write_png(const char *filename, int x, int y, int comp,

```
....
357.    f = fopen(filename, "wb");
```

Incorrect Permission Assignment For Critical Resources\Path 17:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=804
Status New

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	854	854
Object	tmp	tmp

Code Snippet
File Name cosmopolitan/bzip2.c
Method Bool fileExists (Char* name)

```
....
854.     FILE *tmp   = fopen ( name, "rb" );
```

Incorrect Permission Assignment For Critical Resources\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=805
Status	New

	Source	Destination
File	cosmopolitan/stb_image_write.c	cosmopolitan/stb_image_write.c
Line	163	163
Object	f	f

Code Snippet
File Name cosmopolitan/stb_image_write.c
Method static int stbi__start_write_file(stbi__write_context *s,

```
....
163.    FILE *f = fopen(filename, "wb");
```

Incorrect Permission Assignment For Critical Resources\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=806
Status	New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	2399	2399
Object	mkdir	mkdir

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_os_mkdir(JSContext *ctx, JSValueConst this_val,

```
....
2399.       ret = js_get_errno(mkdir(path));
```

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=1050080&projectid=50070&pathid=508
Status	New

	Source	Destination
File	cosmopolitan/fts3_tokenizer.c	cosmopolitan/fts3_tokenizer.c
Line	70	110
Object	pPtr	sizeof

Code Snippet
File Name cosmopolitan/fts3_tokenizer.c
Method static void fts3TokenizerFunc(

```
....
70.     void *pPtr = 0;
....
110.        sqlite3_result_blob(context, (void *)&pPtr, sizeof(pPtr),
SQLITE_TRANSIENT);
```

Use of Sizeof On a Pointer Type\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=509
Status	New

	Source	Destination
File	cosmopolitan/fts3_tokenizer.c	cosmopolitan/fts3_tokenizer.c
Line	70	85
Object	pPtr	sizeof

Code Snippet
File Name cosmopolitan/fts3_tokenizer.c
Method static void fts3TokenizerFunc(

```
....
70.     void *pPtr = 0;
....
85.         if( zName==0 || n!=sizeof(pPtr) ){
```

Use of Sizeof On a Pointer Type\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=500

	70&pathid=510
Status	New

	Source	Destination
File	cosmopolitan/fts3_tokenizer.c	cosmopolitan/fts3_tokenizer.c
Line	356	368
Object	p	sizeof

Code Snippet
File Name cosmopolitan/fts3_tokenizer.c
Method int registerTokenizer(

```
....
356.    const sqlite3_tokenizer_module *p
....
368.    sqlite3_bind_blob(pStmt, 2, &p, sizeof(p), SQLITE_STATIC);
```

Use of Sizeof On a Pointer Type\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=511
Status	New

	Source	Destination
File	cosmopolitan/process.c	cosmopolitan/process.c
Line	541	560
Object	buf	sizeof

Code Snippet
File Name cosmopolitan/process.c
Method flush_appends(void)

```
....
541.        char *buf = gc(malloc(8 * 1024));
....
560.                    while ((count = fread(buf, sizeof(char),
sizeof(buf), f)))
```

Use of Sizeof On a Pointer Type\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=512
Status	New

	Source	Destination

File	cosmopolitan/fts3_tokenizer.c	cosmopolitan/fts3_tokenizer.c
Line	199	199
Object	sizeof	sizeof

Code Snippet
File Name cosmopolitan/fts3_tokenizer.c
Method int sqlite3Fts3InitTokenizer(

```
....
199.        sqlite3_int64 nNew = sizeof(char *)*(iArg+1);
```

Use of Sizeof On a Pointer Type\Path 6:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=513
Status New

	Source	Destination
File	cosmopolitan/main.c	cosmopolitan/main.c
Line	218	218
Object	sizeof	sizeof

Code Snippet
File Name cosmopolitan/main.c
Method int sqlite3_initialize(void){

```
....
218.    assert( SQLITE_PTRSIZE==sizeof(char*) );
```

Use of Sizeof On a Pointer Type\Path 7:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=514
Status New

	Source	Destination
File	cosmopolitan/main.c	cosmopolitan/main.c
Line	769	769
Object	sizeof	sizeof

Code Snippet
File Name cosmopolitan/main.c
Method static int setupLookaside(sqlite3 *db, void *pBuf, int sz, int cnt){

```
....
769.    if( sz<=(int)sizeof(LookasideSlot*) ) sz = 0;
```

Use of Sizeof On a Pointer Type\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=515
Status	New

	Source	Destination
File	cosmopolitan/main.c	cosmopolitan/main.c
Line	805	805
Object	sizeof	sizeof

Code Snippet
File Name cosmopolitan/main.c
Method static int setupLookaside(sqlite3 *db, void *pBuf, int sz, int cnt){

```
....
805.      assert( sz > (int)sizeof(LookasideSlot*) );
```

Use of Sizeof On a Pointer Type\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=516
Status	New

	Source	Destination
File	cosmopolitan/rsa_internal.c	cosmopolitan/rsa_internal.c
Line	110	110
Object	sizeof	sizeof

Code Snippet
File Name cosmopolitan/rsa_internal.c
Method int mbedtls_rsa_deduce_primes(mbedtls_mpi const *N,

```
....
110.      const size_t num_primes = sizeof( primes ) / sizeof( *primes
);
```

Use of Sizeof On a Pointer Type\Path 10:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=517
Status	New

	Source	Destination
File	cosmopolitan/sds.c	cosmopolitan/sds.c
Line	1062	1062
Object	sizeof	sizeof

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

```
....
1062.                    char **new_vector =
s_realloc(vector,((*argc)+1)*sizeof(char*));
```

Use of Sizeof On a Pointer Type\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=518
Status	New

	Source	Destination
File	cosmopolitan/sds.c	cosmopolitan/sds.c
Line	1075	1075
Object	sizeof	sizeof

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

```
....
1075.              if (vector == NULL) vector = s_malloc(sizeof(void*));
```

Use of Sizeof On a Pointer Type\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=519
Status	New

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c

Line	1089	1089
Object	sizeof	sizeof

Code Snippet
File Name cosmopolitan/sqlite3rbu.c
Method static void rbuAllocateIterArrays(sqlite3rbu *p, RbuObjIter *pIter, int nCol){

```
....
1089.    sqlite3_int64 nByte = (2*sizeof(char*) + sizeof(int) +
3*sizeof(u8)) * nCol;
```

Use of Sizeof On a Pointer Type\Path 13:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=520
Status New

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	4956	4956
Object	sizeof	sizeof

Code Snippet
File Name cosmopolitan/sqlite3rbu.c
Method static int rbuVfsShmMap(

```
....
4956.      sqlite3_int64 nByte = (iRegion+1) * sizeof(char*);
```

Use of Sizeof On a Pointer Type\Path 14:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=521
Status New

	Source	Destination
File	cosmopolitan/sqlite3rbu.c	cosmopolitan/sqlite3rbu.c
Line	4967	4967
Object	sizeof	sizeof

Code Snippet
File Name cosmopolitan/sqlite3rbu.c
Method static int rbuVfsShmMap(

```
....
4967.         memset(&apNew[p->nShm], 0, sizeof(char*) * (1 + iRegion -
p->nShm));
```

Use of Sizeof On a Pointer Type\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=522
Status	New

	Source	Destination
File	cosmopolitan/vdbesort.c	cosmopolitan/vdbesort.c
Line	966	966
Object	sizeof	sizeof

Code Snippet
File Name cosmopolitan/vdbesort.c
Method int sqlite3VdbeSorterInit(

```
....
966.     szKeyInfo = sizeof(KeyInfo) + (pCsr->pKeyInfo->nKeyField-
1)*sizeof(CollSeq*);
```

Sizeof Pointer Argument

Query Path:
CPP\Cx\CPP Low Visibility\Sizeof Pointer Argument Version:0

Description

Sizeof Pointer Argument\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=568
Status	New

	Source	Destination
File	cosmopolitan/json.c	cosmopolitan/json.c
Line	1052	1057
Object	Pointer	sizeof

Code Snippet
File Name cosmopolitan/json.c
Method static JsonParse *jsonParseCached(

```
....
1052.    p = sqlite3_malloc64( sizeof(*p) + nJson + 1 );
....
1057.    memset(p, 0, sizeof(*p));
```

Sizeof Pointer Argument\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=569
Status	New

	Source	Destination
File	cosmopolitan/json.c	cosmopolitan/json.c
Line	1052	1057
Object	Pointer	sizeof

Code Snippet
File Name cosmopolitan/json.c
Method static JsonParse *jsonParseCached(

```
....
1052.    p = sqlite3_malloc64( sizeof(*p) + nJson + 1 );
....
1057.    memset(p, 0, sizeof(*p));
```

Sizeof Pointer Argument\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=570
Status	New

	Source	Destination
File	cosmopolitan/json.c	cosmopolitan/json.c
Line	1057	1057
Object	Pointer	sizeof

Code Snippet
File Name cosmopolitan/json.c
Method static JsonParse *jsonParseCached(

```
....
1057.    memset(p, 0, sizeof(*p));
```

Sizeof Pointer Argument\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=571
Status	New

	Source	Destination
File	cosmopolitan/main.c	cosmopolitan/main.c
Line	3250	3250
Object	aHardLimit	sizeof

Code Snippet
File Name cosmopolitan/main.c
Method static int openDatabase(

```
....
3250.    assert( sizeof(db->aLimit)==sizeof(aHardLimit) );
```

Sizeof Pointer Argument\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=572
Status	New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	2677	2677
Object	buf	sizeof

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_os_readlink(JSContext *ctx, JSValueConst this_val,

```
....
2677.      res = readlink(path, buf, sizeof(buf) - 1);
```

Sizeof Pointer Argument\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=573
Status	New

	Source	Destination
File	cosmopolitan/crypt_blowfish.c	cosmopolitan/crypt_blowfish.c
Line	839	839
Object	ai	sizeof

Code Snippet
File Name cosmopolitan/crypt_blowfish.c
Method char *__crypt_blowfish(const char *key, const char *setting, char *output)

```
....
839.                    !memcmp(ai, yi, sizeof(ai));
```

Sizeof Pointer Argument\Path 7:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=574
Status New

	Source	Destination
File	cosmopolitan/json.c	cosmopolitan/json.c
Line	1052	1052
Object	Pointer	sizeof

Code Snippet
File Name cosmopolitan/json.c
Method static JsonParse *jsonParseCached(

```
....
1052.    p = sqlite3_malloc64( sizeof(*p) + nJson + 1 );
```

Sizeof Pointer Argument\Path 8:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=575
Status New

	Source	Destination
File	cosmopolitan/json.c	cosmopolitan/json.c
Line	1052	1052
Object	Pointer	sizeof

Code Snippet

File Name	cosmopolitan/json.c
Method	static JsonParse *jsonParseCached(

```
....
1052.    p = sqlite3_malloc64( sizeof(*p) + nJson + 1 );
```

Potential Path Traversal

Query Path:
CPP\Cx\CPP Low Visibility\Potential Path Traversal Version:0

Categories

OWASP Top 10 2013: A4-Insecure Direct Object References
OWASP Top 10 2017: A5-Broken Access Control

Description

Potential Path Traversal\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=460
Status	New

Method main at line 120 of cosmopolitan/maketab.c gets user input from the argv element. This element's value then flows through the code and is eventually used in a file path for local disk access in main at line 120 of cosmopolitan/maketab.c. This may cause a Path Traversal vulnerability.

	Source	Destination
File	cosmopolitan/maketab.c	cosmopolitan/maketab.c
Line	120	141
Object	argv	argv

Code Snippet

File Name	cosmopolitan/maketab.c
Method	int main(int argc, char *argv[])

```
....
120.  int main(int argc, char *argv[])
....
141.        if ((fp = fopen(argv[1], "r")) == NULL) {
```

Potential Path Traversal\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=461
Status	New

Method main at line 1947 of cosmopolitan/run-test262.c gets user input from the argv element. This element's value then flows through the code and is eventually used in a file path for local disk access in namelist_load at line 315 of cosmopolitan/run-test262.c. This may cause a Path Traversal vulnerability.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1947	321
Object	argv	filename

Code Snippet
File Name cosmopolitan/run-test262.c
Method int main(int argc, char **argv)

```
....
1947.  int main(int argc, char **argv)
```

File Name cosmopolitan/run-test262.c
Method void namelist_load(namelist_t *lp, const char *filename)

```
....
321.       f = fopen(filename, "rb");
```

Potential Path Traversal\Path 3:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=462
Status New

Method main at line 1947 of cosmopolitan/run-test262.c gets user input from the argv element. This element's value then flows through the code and is eventually used in a file path for local disk access in main at line 1947 of cosmopolitan/run-test262.c. This may cause a Path Traversal vulnerability.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1947	2056
Object	argv	report_filename

Code Snippet
File Name cosmopolitan/run-test262.c
Method int main(int argc, char **argv)

```
....
1947.  int main(int argc, char **argv)
....
2056.              outfile = fopen(report_filename, "wb");
```

Potential Path Traversal\Path 4:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=463
Status	New

Method main at line 1947 of cosmopolitan/run-test262.c gets user input from the argv element. This element's value then flows through the code and is eventually used in a file path for local disk access in main at line 1947 of cosmopolitan/run-test262.c. This may cause a Path Traversal vulnerability.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1947	2029
Object	argv	error_filename

Code Snippet
File Name cosmopolitan/run-test262.c
Method int main(int argc, char **argv)

```
....
1947.  int main(int argc, char **argv)
....
2029.              error_out = fopen(error_filename, "w");
```

Potential Path Traversal\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=464
Status	New

Method main at line 1947 of cosmopolitan/run-test262.c gets user input from the argv element. This element's value then flows through the code and is eventually used in a file path for local disk access in load_config at line 919 of cosmopolitan/run-test262.c. This may cause a Path Traversal vulnerability.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1947	933
Object	argv	filename

Code Snippet
File Name cosmopolitan/run-test262.c
Method int main(int argc, char **argv)

```
....
1947.  int main(int argc, char **argv)
```

File Name cosmopolitan/run-test262.c
Method void load_config(const char *filename)

```
....
933.      f = fopen(filename, "rb");
```

Potential Path Traversal\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=465
Status	New

Method main at line 269 of cosmopolitan/zip2.c gets user input from the argv element. This element's value then flows through the code and is eventually used in a file path for local disk access in main at line 269 of cosmopolitan/zip2.c. This may cause a Path Traversal vulnerability.

	Source	Destination
File	cosmopolitan/zip2.c	cosmopolitan/zip2.c
Line	269	275
Object	argv	argv

Code Snippet
File Name cosmopolitan/zip2.c
Method int main(int argc, char *argv[]) {

```
....
269.  int main(int argc, char *argv[]) {
....
275.    CHECK_NE(-1, (fd = open(argv[1], O_RDONLY)));
```

Potential Path Traversal\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=466
Status	New

Method main at line 269 of cosmopolitan/zip2.c gets user input from the argv element. This element's value then flows through the code and is eventually used in a file path for local disk access in main at line 269 of cosmopolitan/zip2.c. This may cause a Path Traversal vulnerability.

	Source	Destination
File	cosmopolitan/zip2.c	cosmopolitan/zip2.c
Line	269	275
Object	argv	argv

Code Snippet
File Name cosmopolitan/zip2.c
Method int main(int argc, char *argv[]) {

```
....
269.  int main(int argc, char *argv[]) {
....
275.    CHECK_NE(-1, (fd = open(argv[1], O_RDONLY)));
```

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=1050080&projectid=50070&pathid=456
Status	New

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	943	943
Object	getopt	getopt

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

```
....
943.        c = getopt (argc, argv, "abc:d:0123456789");
```

Inconsistent Implementations\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=457
Status	New

	Source	Destination
File	cosmopolitan/qjsc.c	cosmopolitan/qjsc.c
Line	514	514
Object	getopt	getopt

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

```
....
514.          c = getopt(argc, argv, "ho:cN:f:mxevM:p:S:D:");
```

Inconsistent Implementations\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=458
Status	New

	Source	Destination
File	cosmopolitan/testmain.c	cosmopolitan/testmain.c
Line	83	83
Object	getopt	getopt

Code Snippet
File Name cosmopolitan/testmain.c
Method void GetOpts(int argc, char *argv[]) {

```
....
83.    while ((opt = getopt(argc, argv, "?hbv")) != -1) {
```

Inconsistent Implementations\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=459
Status	New

	Source	Destination
File	cosmopolitan/getopt1.c	cosmopolitan/getopt1.c
Line	88	88
Object	getopt_long	getopt_long

Code Snippet
File Name cosmopolitan/getopt1.c
Method main (int argc, char **argv)

```
....
88.        c = getopt_long (argc, argv, "abc:d:0123456789",
```

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=1050080&projectid=50070&pathid=523
Status	New

The buffer allocated by <= in cosmopolitan/compile.c at line 813 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	856	856
Object	<=	<=

Code Snippet
File Name cosmopolitan/compile.c
Method compile_tr(char *p, struct s_tr **py)

```
....
856.              for (i = 0; i <= UCHAR_MAX; i++)
```

Potential Off by One Error in Loops\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=524
Status	New

The buffer allocated by <= in cosmopolitan/compile.c at line 813 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	cosmopolitan/compile.c	cosmopolitan/compile.c
Line	870	870
Object	<=	<=

Code Snippet
File Name cosmopolitan/compile.c
Method compile_tr(char *p, struct s_tr **py)

```
....
870.              for (i = 0; i <= UCHAR_MAX; i++)
```

Potential Off by One Error in Loops\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=525
Status	New

The buffer allocated by <= in cosmopolitan/puff.c at line 375 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	cosmopolitan/puff.c	cosmopolitan/puff.c
Line	383	383
Object	<=	<=

Code Snippet
File Name cosmopolitan/puff.c
Method local int construct(struct huffman *h, const short *length, int n)

```
....
383.      for (len = 0; len <= MAXBITS; len++)
```

Potential Off by One Error in Loops\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=526
Status	New

The buffer allocated by <= in cosmopolitan/sqlite3expert.c at line 1521 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	cosmopolitan/sqlite3expert.c	cosmopolitan/sqlite3expert.c
Line	1579	1579
Object	<=	<=

Code Snippet
File Name cosmopolitan/sqlite3expert.c
Method static int idxPopulateOneStat1(

```
....
1579.       for(i=0; i<=nCol; i++) aStat[i] = 1;
```

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=1050080&projectid=50070&pathid=807
Status	New

The system data read by compressedStreamEOF in the file cosmopolitan/bzip2.c at line 686 is potentially exposed by compressedStreamEOF found in cosmopolitan/bzip2.c at line 686.

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	693	693
Object	perror	perror

Code Snippet
File Name cosmopolitan/bzip2.c
Method void compressedStreamEOF (void)

```
....
693.       perror ( progName );
```

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=1050080&projectid=50070&pathid=808
Status	New

The system data read by ioError in the file cosmopolitan/bzip2.c at line 703 is potentially exposed by ioError found in cosmopolitan/bzip2.c at line 703.

	Source	Destination
File	cosmopolitan/bzip2.c	cosmopolitan/bzip2.c
Line	709	709

Object	perror	perror

Code Snippet
File Name cosmopolitan/bzip2.c
Method void ioError (void)

```
....
709.     perror ( progName );
```

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=1050080&projectid=50070&pathid=809
Status	New

The system data read by main in the file cosmopolitan/qjsc.c at line 478 is potentially exposed by main found in cosmopolitan/qjsc.c at line 478.

	Source	Destination
File	cosmopolitan/qjsc.c	cosmopolitan/qjsc.c
Line	615	615
Object	perror	perror

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

```
....
615.          perror(cfilename);
```

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=1050080&projectid=50070&pathid=810
Status	New

The system data read by perror_exit in the file cosmopolitan/run-test262.c at line 124 is potentially exposed by perror_exit found in cosmopolitan/run-test262.c at line 124.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	128	128
Object	perror	perror

Code Snippet
File Name cosmopolitan/run-test262.c
Method void perror_exit(int errcode, const char *s)

```
....
128.      perror(s);
```

Heuristic 2nd Order Buffer Overflow malloc

Query Path:
CPP\Cx\CPP Heuristic\Heuristic 2nd Order Buffer Overflow malloc 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 malloc\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=558
Status	New

The size of the buffer used by *str_append in str, at line 144 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that load_config passes to buf, at line 919 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	940	151
Object	buf	str

Code Snippet
File Name cosmopolitan/run-test262.c
Method void load_config(const char *filename)

```
....
940.      while (fgets(buf, sizeof(buf), f) != NULL) {
```

File Name cosmopolitan/run-test262.c
Method char *str_append(char **pp, const char *sep, const char *str) {

```
....
151.      res = malloc(len + strlen(str) + 1);
```

Heuristic 2nd Order Buffer Overflow malloc\Path 2:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=559
Status	New

The size of the buffer used by *strdup_len in len, at line 132 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that load_config passes to buf, at line 919 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	940	134
Object	buf	len

Code Snippet
File Name cosmopolitan/run-test262.c
Method void load_config(const char *filename)

```
....
940.      while (fgets(buf, sizeof(buf), f) != NULL) {
```

File Name cosmopolitan/run-test262.c
Method char *strdup_len(const char *str, int len)

```
....
134.      char *p = malloc(len + 1);
```

Heuristic 2nd Order Buffer Overflow malloc\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=560
Status	New

The size of the buffer used by *strdup_len in BinaryExpr, at line 132 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that load_config passes to buf, at line 919 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	940	134
Object	buf	BinaryExpr

Code Snippet
File Name cosmopolitan/run-test262.c
Method void load_config(const char *filename)

```
....
940.      while (fgets(buf, sizeof(buf), f) != NULL) {
```

File Name cosmopolitan/run-test262.c

Method char *strdup_len(const char *str, int len)

```
....
134.      char *p = malloc(len + 1);
```

Heuristic Buffer Overflow malloc

Query Path:
CPP\Cx\CPP Heuristic\Heuristic Buffer Overflow malloc 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 Buffer Overflow malloc\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=561
Status	New

The size of the buffer used by *strdup_len in len, at line 132 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 1947 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1947	134
Object	argv	len

Code Snippet
File Name cosmopolitan/run-test262.c
Method int main(int argc, char **argv)

```
....
1947.  int main(int argc, char **argv)
```

File Name cosmopolitan/run-test262.c

Method char *strdup_len(const char *str, int len)

```
....
134.      char *p = malloc(len + 1);
```

Heuristic Buffer Overflow malloc\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=562
Status	New

The size of the buffer used by *strdup_len in BinaryExpr, at line 132 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 1947 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1947	134
Object	argv	BinaryExpr

Code Snippet
File Name cosmopolitan/run-test262.c
Method int main(int argc, char **argv)

```
....
1947.  int main(int argc, char **argv)
```

File Name cosmopolitan/run-test262.c
Method char *strdup_len(const char *str, int len)

```
....
134.      char *p = malloc(len + 1);
```

Heuristic Buffer Overflow malloc\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=563
Status	New

The size of the buffer used by *str_append in str, at line 144 of cosmopolitan/run-test262.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that main passes to argv, at line 1947 of cosmopolitan/run-test262.c, to overwrite the target buffer.

	Source	Destination
File	cosmopolitan/run-test262.c	cosmopolitan/run-test262.c
Line	1947	151

Object	argv	str

Code Snippet
File Name cosmopolitan/run-test262.c
Method int main(int argc, char **argv)

```
....
1947.  int main(int argc, char **argv)
```

File Name cosmopolitan/run-test262.c

Method char *str_append(char **pp, const char *sep, const char *str) {

```
....
151.      res = malloc(len + strlen(str) + 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=1050080&projectid=50070&pathid=564
Status	New

	Source	Destination
File	cosmopolitan/getopt.c	cosmopolitan/getopt.c
Line	568	568
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name cosmopolitan/getopt.c
Method _getopt_internal (int argc, char *const *argv, const char *optstring,

```
....
568.                + (longopts != NULL && argv[optind][1] == '-'));
```

Arithmenic Operation On Boolean\Path 2:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=565
Status	New

	Source	Destination
File	cosmopolitan/lstrlib.c	cosmopolitan/lstrlib.c
Line	175	175
Object	BinaryExpr	BinaryExpr

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

```
....
175.    else if (l_unlikely(l + lsep < l || l + lsep > MAXSIZE / n))
```

Arithmenic Operation On Boolean\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=566
Status	New

	Source	Destination
File	cosmopolitan/lstrlib.c	cosmopolitan/lstrlib.c
Line	1696	1696
Object	BinaryExpr	BinaryExpr

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

```
....
1696.      luaL_argcheck(L, totalsize <= MAXSIZE - size, 1,
```

Unreleased Resource Leak

Query Path:
CPP\Cx\CPP Low Visibility\Unreleased Resource Leak Version:0

Categories

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

Description

Unreleased Resource Leak\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=500

	70&pathid=527
Status	New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	3111	3111
Object	ps	ps

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSWorkerMessagePipe *js_new_message_pipe(void)

```
....
3111.       pthread_mutex_init(&ps->mutex, NULL);
```

Insecure Temporary File

Query Path:
CPP\Cx\CPP Low Visibility\Insecure Temporary File Version:0

Categories

NIST SP 800-53: SC-4 Information in Shared Resources (P1)
OWASP Top 10 2017: A3-Sensitive Data Exposure

Description
Insecure Temporary File\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050080&projectid=50070&pathid=567
Status	New

	Source	Destination
File	cosmopolitan/quickjs-libc.c	cosmopolitan/quickjs-libc.c
Line	947	947
Object	tmpfile	tmpfile

Code Snippet
File Name cosmopolitan/quickjs-libc.c
Method static JSValue js_std_tmpfile(JSContext *ctx, JSValueConst this_val,

```
....
947.       f = tmpfile();
```

Buffer Overflow LongString

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.
Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

Buffer Overflow Indexes

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.
Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

Buffer Overflow cpycat

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.
Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

Buffer Overflow IndexFromInput

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.
Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

Buffer Overflow StrcpyStrcat

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.
Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

Buffer Overflow OutOfBound

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.
Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

Buffer Overflow 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

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);
```

Char 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;
}
```

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

Short Overflow

Risk

What might happen

Assigning large data types into smaller data types, without proper checks and explicit casting, will lead to undefined behavior and unintentional effects, such as data corruption (e.g. value wraparound, wherein maximum values become minimum values); system crashes; infinite loops; logic errors, such as bypassing of security mechanisms; or even buffer overflows leading to arbitrary code execution.

Cause

How does it happen

This flaw can occur when implicitly casting numerical data types of a larger size, into a variable with a data type of a smaller size. This forces the program to discard some bits of information from the number. Depending on how the numerical data types are stored in memory, this is often the bits with the highest value, causing substantial corruption of the stored number. Alternatively, the sign bit of a signed integer could be lost, completely reversing the intention of the number.

General Recommendations

How to avoid it

- Avoid casting larger data types to smaller types.
- Prefer promoting the target variable to a large enough data type.
- If downcasting is necessary, always check that values are valid and in range of the target type, before casting

---

Source Code Examples

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

CPP

Use of Variable after It was Freed

```
free(input);
printf("%s", input);
```

Use of Pointer to Local Variable That Was Freed On Return

```
int* func1()
{
	int i;
	i = 1;
	return &i;
}

void func2()

{

	int j;
	j = 5;
```

```
}

//..
	int * i = func1();
	printf("%d\r\n", *i); // Output could be 1 or Segmentation Fault
	func2();
	printf("%d\r\n", *i); // Output is 5, which is j's value, as func2() overwrote data in
the stack
//..
```

Use of Uninitialized Variable

Weakness ID: 457 *(Weakness Variant)* **Status:** Draft

Description

Description Summary

The code uses a variable that has not been initialized, leading to unpredictable or unintended results.

Extended Description

In some languages, such as C, an uninitialized variable contains contents of previously-used memory. An attacker can sometimes control or read these contents.

Time of Introduction

- Implementation

Applicable Platforms

Languages

C: *(Sometimes)*

C++: *(Sometimes)*

Perl: *(Often)*

All

Common Consequences

Scope	Effect
Availability Integrity	Initial variables usually contain junk, which can not be trusted for consistency. This can lead to denial of service conditions, or modify control flow in unexpected ways. In some cases, an attacker can "pre-initialize" the variable using previous actions, which might enable code execution. This can cause a race condition if a lock variable check passes when it should not.
Authorization	Strings that are not initialized are especially dangerous, since many functions expect a null at the end -- and only at the end -- of a string.

Likelihood of Exploit

High

Demonstrative Examples

Example 1

The following switch statement is intended to set the values of the variables aN and bN, but in the default case, the programmer has accidentally set the value of aN twice. As a result, bN will have an undefined value.

(Bad Code)

Example Language: **C**

```
switch (ctl) {
case -1:
aN = 0;
bN = 0;
break;
case 0:
aN = i;
bN = -i;
break;
case 1:
aN = i + NEXT_SZ;
bN = i - NEXT_SZ;
break;
default:
```

```
aN = -1;
aN = -1;
break;
}
repaint(aN, bN);
```

Most uninitialized variable issues result in general software reliability problems, but if attackers can intentionally trigger the use of an uninitialized variable, they might be able to launch a denial of service attack by crashing the program. Under the right circumstances, an attacker may be able to control the value of an uninitialized variable by affecting the values on the stack prior to the invocation of the function.

Example 2

Example Languages: **C++ and Java**

```
int foo;
void bar() {
if (foo==0)
/.../
/../
}
```

Observed Examples

Reference	Description
CVE-2008-0081	Uninitialized variable leads to code execution in popular desktop application.
CVE-2007-4682	Crafted input triggers dereference of an uninitialized object pointer.
CVE-2007-3468	Crafted audio file triggers crash when an uninitialized variable is used.
CVE-2007-2728	Uninitialized random seed variable used.

Potential Mitigations

Phase: Implementation

Assign all variables to an initial value.

Phase: Build and Compilation

Most compilers will complain about the use of uninitialized variables if warnings are turned on.

Phase: Requirements

The choice could be made to use a language that is not susceptible to these issues.

Phase: Architecture and Design

Mitigating technologies such as safe string libraries and container abstractions could be introduced.

Other Notes

Before variables are initialized, they generally contain junk data of what was left in the memory that the variable takes up. This data is very rarely useful, and it is generally advised to pre-initialize variables or set them to their first values early. If one forgets -- in the C language -- to initialize, for example a char *, many of the simple string libraries may often return incorrect results as they expect the null termination to be at the end of a string.

Stack variables in C and C++ are not initialized by default. Their initial values are determined by whatever happens to be in their location on the stack at the time the function is invoked. Programs should never use the value of an uninitialized variable. It is not uncommon for programmers to use an uninitialized variable in code that handles errors or other rare and exceptional circumstances. Uninitialized variable warnings can sometimes indicate the presence of a typographic error in the code.

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	Weakness Base	456	Missing Initialization	**Development Concepts (primary)699 Research Concepts**

				(primary)1000
MemberOf	View	630	Weaknesses Examined by SAMATE	**Weaknesses Examined by SAMATE (primary)630**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Uninitialized variable
7 Pernicious Kingdoms			Uninitialized Variable

White Box Definitions

A weakness where the code path has:

1. start statement that defines variable
2. end statement that accesses the variable
3. the code path does not contain a statement that assigns value to the variable

References

mercy. "Exploiting Uninitialized Data". Jan 2006. < http://www.felinemenace.org/~mercy/papers/UBehavior/UBehavior.zip>.

Microsoft Security Vulnerability Research & Defense. "MS08-014 : The Case of the Uninitialized Stack Variable Vulnerability". 2008-03-11. <http://blogs.technet.com/swi/archive/2008/03/11/the-case-of-the-uninitialized-stack-variable-vulnerability.aspx>.

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, Description, Relationships, Observed Example, Other Notes, References, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences, Demonstrative Examples, Potential Mitigations		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		

Previous Entry Names

Change Date	Previous Entry Name
2008-04-11	Uninitialized Variable

BACK TO TOP

Use of Zero Initialized Pointer

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
- Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
- Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.

Source Code Examples

CPP

Explicit NULL Dereference

```
char * input = NULL;
printf("%s", input);
```

Implicit NULL Dereference

```
char * input;
printf("%s", input);
```

Java

Explicit Null Dereference

```
Object o = null;
out.println(o.getClass());
```

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

Potential Path Traversal

Risk

What might happen

An attacker could define any arbitrary file path for the application to use, potentially leading to:

- Stealing sensitive files, such as configuration or system files
- Overwriting files such as program binaries, configuration files, or system files
- Deleting critical files, causing a denial of service (DoS).

Cause

How does it happen

The application uses user input in the file path for accessing files on the application server's local disk. This enables an attacker to arbitrarily determine the file path.

General Recommendations

How to avoid it

1. Ideally, avoid depending on user input for file selection.
2. Validate all input, regardless of source. Validation should be based on a whitelist: accept only data fitting a specified structure, rather than reject bad patterns. Check for:
   - Data type
   - Size
   - Range
   - Format
   - Expected values
3. Accept user input only for the filename, not for the path and folders.
4. Ensure that file path is fully canonicalized.
5. Explicitly limit the application to using a designated folder that separate from the applications binary folder.
6. Restrict the privileges of the application's OS user to necessary files and folders. The application should not be able to write to the application binary folder, and should not read anything outside of the application folder and data folder.

Source Code Examples

CSharp

Using unvalidated user input as the file name may enable the user to access arbitrary files on the server local disk

```
public class PathTraversal
{
        private void foo(TextBox textbox1)
    {
                string fileNum = textbox1.Text;
                string path = "c:\files\file" + fileNum;
                FileStream f = new FileStream(path, FileMode.Open);
                byte[] output = new byte[10];
                f.Read(output,0, 10);
```

```
        }
}
```

Potentially hazardous characters are removed from the user input before use

```
public class PathTraversalFixed
{
        private void foo(TextBox textbox1)
    {
                string fileNum = textbox1.Text.Replace("\", "").Replace("..", "");
         string path = "c:\files\file" + fileNum;
                FileStream f = new FileStream(path, FileMode.Open);
                byte[] output = new byte[10];
                f.Read(output,0, 10);
        }
}
```

Java

Using unvalidated user input as the file name may enable the user to access arbitrary files on the server local disk

```
public class Absolute_Path_Traversal {
    public static void main(String[] args) {
        Scanner userInputScanner = new Scanner(System.in);
        System.out.print("\nEnter file name: ");
        String name = userInputScanner.nextLine();
        String path = "c:\files\file" + name;
        try {
            BufferedReader reader = new BufferedReader(new FileReader(path));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Potentially hazardous characters are removed from the user input before use

```
public class Absolute_Path_Traversal_Fixed {
    public static void main(String[] args) {
        Scanner userInputScanner = new Scanner(System.in);
        System.out.print("\nEnter file name: ");
        String name = userInputScanner.nextLine();
        name = name.replace("/", "").replace("..", "");
        String path = "c:\files\file" + name;
        try {
            BufferedReader reader = new BufferedReader(new FileReader(path));
        } catch (Exception e) {
            e.printStackTrace();
        }
    }
}
```

Unchecked Return Value

Risk

What might happen

A program that does not check function return values could cause the application to enter an undefined state. This could lead to unexpected behavior and unintended consequences, including inconsistent data, system crashes or other error-based exploits.

Cause

How does it happen

The application calls a system function, but does not receive or check the result of this funciton. These functions often return error codes in the result, or share other status codes with it's caller. The application simply ignores this result value, losing this vital information.

General Recommendations

How to avoid it

- Always check the result of any called function that returns a value, and verify the result is an expected value.

- Ensure the calling function responds to all possible return values.

- Expect runtime errors and handle them gracefully. Explicitly define a mechanism for handling unexpected errors.

Source Code Examples

CPP

Unchecked Memory Allocation

```
buff = (char*) malloc(size);
strncpy(buff, source, size);
```

Safer Memory Allocation

```
buff = (char*) malloc(size+1);
if (buff==NULL) exit(1);

strncpy(buff, source, size);
buff[size] = '\0';
```

Use of sizeof() on a Pointer Type

Weakness ID: 467 *(Weakness Variant)* **Status:** Draft

Description

Description Summary

The code calls sizeof() on a malloced pointer type, which always returns the wordsize/8. This can produce an unexpected result if the programmer intended to determine how much memory has been allocated.

Time of Introduction

- Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

Scope	Effect
Integrity	This error can often cause one to allocate a buffer that is much smaller than what is needed, leading to resultant weaknesses such as buffer overflows.

Likelihood of Exploit

High

Demonstrative Examples

Example 1

Care should be taken to ensure sizeof returns the size of the data structure itself, and not the size of the pointer to the data structure.

In this example, sizeof(foo) returns the size of the pointer.

(Bad Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(foo));
```

In this example, sizeof(*foo) returns the size of the data structure and not the size of the pointer.

(Good Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(*foo));
```

Example 2

This example defines a fixed username and password. The AuthenticateUser() function is intended to accept a username and a password from an untrusted user, and check to ensure that it matches the username and password. If the username and password match, AuthenticateUser() is intended to indicate that authentication succeeded.

(Bad Code)

```
/* Ignore CWE-259 (hard-coded password) and CWE-309 (use of password system for authentication) for this example. */

char *username = "admin";
char *pass = "password";

int AuthenticateUser(char *inUser, char *inPass) {
```

```
printf("Sizeof username = %d\n", sizeof(username));
printf("Sizeof pass = %d\n", sizeof(pass));

if (strncmp(username, inUser, sizeof(username))) {
printf("Auth failure of username using sizeof\n");
return(AUTH_FAIL);
}
/* Because of CWE-467, the sizeof returns 4 on many platforms and architectures. */
if (! strncmp(pass, inPass, sizeof(pass))) {
printf("Auth success of password using sizeof\n");
return(AUTH_SUCCESS);
}
else {
printf("Auth fail of password using sizeof\n");
return(AUTH_FAIL);
}
}

int main (int argc, char **argv)
{
int authResult;

if (argc < 3) {
ExitError("Usage: Provide a username and password");
}
authResult = AuthenticateUser(argv[1], argv[2]);
if (authResult != AUTH_SUCCESS) {
ExitError("Authentication failed");
}
else {
DoAuthenticatedTask(argv[1]);
}
}
```

In AuthenticateUser(), because sizeof() is applied to a parameter with an array type, the sizeof() call might return 4 on many modern architectures. As a result, the strncmp() call only checks the first four characters of the input password, resulting in a partial comparison (CWE-187), leading to improper authentication (CWE-287).

Because of the partial comparison, any of these passwords would still cause authentication to succeed for the "admin" user:

(Attack)

```
pass5
passABCDEFGH
passWORD
```

Because only 4 characters are checked, this significantly reduces the search space for an attacker, making brute force attacks more feasible.

The same problem also applies to the username, so values such as "adminXYZ" and "administrator" will succeed for the username.

Potential Mitigations

Phase: Implementation

Use expressions such as "sizeof(*pointer)" instead of "sizeof(pointer)", unless you intend to run sizeof() on a pointer type to gain some platform independence or if you are allocating a variable on the stack.

Other Notes

The use of sizeof() on a pointer can sometimes generate useful information. An obvious case is to find out the wordsize on a platform. More often than not, the appearance of sizeof(pointer) indicates a bug.

Weakness Ordinalities

Ordinality	Description
Primary	*(where the weakness exists independent of other weaknesses)*

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	465	Pointer Issues	**Development Concepts (primary)699**
ChildOf	Weakness Class	682	Incorrect Calculation	**Research Concepts (primary)1000**
ChildOf	Category	737	CERT C Secure Coding Section 03 - Expressions (EXP)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
ChildOf	Category	740	CERT C Secure Coding Section 06 - Arrays (ARR)	Weaknesses Addressed by the CERT C Secure Coding Standard734
CanPrecede	Weakness Base	131	Incorrect Calculation of Buffer Size	Research Concepts1000

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Use of sizeof() on a pointer type
CERT C Secure Coding	ARR01-C		Do not apply the sizeof operator to a pointer when taking the size of an array
CERT C Secure Coding	EXP01-C		Do not take the size of a pointer to determine the size of the pointed-to type

White Box Definitions

A weakness where code path has:

1. end statement that passes an identity of a dynamically allocated memory resource to a sizeof operator
2. start statement that allocates the dynamically allocated memory resource

References

Robert Seacord. "EXP01-A. Do not take the sizeof a pointer to determine the size of a type". <https://www.securecoding.cert.org/confluence/display/seccode/EXP01-A.+Do+not+take+the+sizeof+a+pointer+to+determine+the+size+of+a+type>.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	CLASP		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		

BACK TO TOP

Potential Off by One Error in Loops

Risk

What might happen

An off by one error may result in overwriting or over-reading of unintended memory; in most cases, this can result in unexpected behavior and even application crashes. In other cases, where allocation can be controlled by an attacker, a combination of variable assignment and an off by one error can result in execution of malicious code.

Cause

How does it happen

Often when designating variables to memory, a calculation error may occur when determining size or length that is off by one.

For example in loops, when allocating an array of size 2, its cells are counted as 0,1 - therefore, if a For loop iterator on the array is incorrectly set with the start condition i=0 and the continuation condition i<=2, three cells will be accessed instead of 2, and an attempt will be made to write or read cell [2], which was not originally allocated, resulting in potential corruption of memory outside the bounds of the originally assigned array.

Another example occurs when a null-byte terminated string, in the form of a character array, is copied without its terminating null-byte. Without the null-byte, the string representation is unterminated, resulting in certain functions to over-read memory as they expect the missing null terminator.

General Recommendations

How to avoid it

- Always ensure that a given iteration boundary is correct:
  - With array iterations, consider that arrays begin with cell 0 and end with cell n-1, for a size n array.
  - With character arrays and null-byte terminated string representations, consider that the null byte is required and should not be overwritten or ignored; ensure functions in use are not vulnerable to off-by-one, specifically for instances where null-bytes are automatically appended after the buffer, instead of in place of its last character.
- Where possible, use safe functions that manage memory and are not prone to off-by-one errors.

Source Code Examples

CPP

Off-By-One in For Loop

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
for (int i = 0; i <= 5; i++)
{
      ptr[i] = i * 2 + 1; // ptr[5] will be set, but is out of bounds
```

```
}
```

Proper Iteration in For Loop

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
for (int i = 0; i < 5; i++)
{
      ptr[i] = i * 2 + 1; // ptr[0-4] are well defined
}
```

Off-By-One in strncat

```
strncat(buf, input, sizeof(buf) - strlen(buf)); // actual value should be sizeof(buf)-
strlen(buf)-1 - this form will overwrite the terminating nullbyte
```

Resource Locking Problems

Category ID: 411 *(Category)* **Status:** Draft

Description

Description Summary

Weaknesses in this category are related to improper handling of locks that are used to control access to resources.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	399	Resource Management Errors	**Development Concepts (primary)699**
ParentOf	Weakness Base	412	Unrestricted Externally Accessible Lock	Development Concepts699
ParentOf	Weakness Base	413	Insufficient Resource Locking	**Development Concepts (primary)699**
ParentOf	Weakness Base	414	Missing Lock Check	**Development Concepts (primary)699**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
PLOVER			Resource Locking problems

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	PLOVER		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-09-08	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		

BACK TO TOP

NULL Pointer Dereference

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
- Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
- Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.

Source Code Examples

Heuristic 2nd Order Buffer Overflow malloc

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

Heuristic Buffer Overflow malloc

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

Insecure Temporary File

Weakness ID: 377 *(Weakness Base)* **Status:** Incomplete

Description

Description Summary

Creating and using insecure temporary files can leave application and system data vulnerable to attack.

Time of Introduction

- Architecture and Design
- Implementation

Applicable Platforms

Languages

All

Demonstrative Examples

Example 1

The following code uses a temporary file for storing intermediate data gathered from the network before it is processed.

(Bad Code)

Example Language: **C**

```
if (tmpnam_r(filename)) {

FILE* tmp = fopen(filename,"wb+");
while((recv(sock,recvbuf,DATA_SIZE, 0) > 0)&(amt!=0)) amt = fwrite(recvbuf,1,DATA_SIZE,tmp);
}
...
```

This otherwise unremarkable code is vulnerable to a number of different attacks because it relies on an insecure method for creating temporary files. The vulnerabilities introduced by this function and others are described in the following sections. The most egregious security problems related to temporary file creation have occurred on Unix-based operating systems, but Windows applications have parallel risks. This section includes a discussion of temporary file creation on both Unix and Windows systems. Methods and behaviors can vary between systems, but the fundamental risks introduced by each are reasonably constant.

Other Notes

Applications require temporary files so frequently that many different mechanisms exist for creating them in the C Library and Windows(R) API. Most of these functions are vulnerable to various forms of attacks.

The functions designed to aid in the creation of temporary files can be broken into two groups based whether they simply provide a filename or actually open a new file. - Group 1: "Unique" Filenames: The first group of C Library and WinAPI functions designed to help with the process of creating temporary files do so by generating a unique file name for a new temporary file, which the program is then supposed to open. This group includes C Library functions like tmpnam(), tempnam(), mktemp() and their C++ equivalents prefaced with an _ (underscore) as well as the GetTempFileName() function from the Windows API. This group of functions suffers from an underlying race condition on the filename chosen. Although the functions guarantee that the filename is unique at the time it is selected, there is no mechanism to prevent another process or an attacker from creating a file with the same name after it is selected but before the application attempts to open the file. Beyond the risk of a legitimate collision caused by another call to the same function, there is a high probability that an attacker will be able to create a malicious collision because the filenames generated by these functions are not sufficiently randomized to make them difficult to guess. If a file with the selected name is created, then depending on how the file is opened the existing contents or access permissions of the file may remain intact. If the existing contents of the file are malicious in nature, an attacker may be able to inject dangerous data into the application when it reads data back from the temporary file. If an attacker pre-creates the file with relaxed access permissions, then data stored in the temporary file by the application may be accessed, modified or corrupted by an attacker. On Unix based systems an even more insidious attack is possible if the attacker pre-creates the file as a link to another important file. Then, if the application truncates or writes data to the file, it may unwittingly perform damaging operations for the attacker. This is an especially serious threat if the program operates with elevated permissions. Finally, in the best case the file will be opened with the a call to open() using the O_CREAT and O_EXCL flags or to CreateFile() using the CREATE_NEW attribute, which will fail if the file already exists and therefore prevent the types of attacks described above. However, if an attacker is able to accurately predict a sequence of temporary file names, then the application may be prevented from opening necessary temporary storage causing a denial of service (DoS) attack. This type of attack would not be difficult to mount given the small amount of randomness used in

the selection of the filenames generated by these functions. - Group 2: "Unique" Files: The second group of C Library functions attempts to resolve some of the security problems related to temporary files by not only generating a unique file name, but also opening the file. This group includes C Library functions like tmpfile() and its C++ equivalents prefaced with an _ (underscore), as well as the slightly better-behaved C Library function mkstemp(). The tmpfile() style functions construct a unique filename and open it in the same way that fopen() would if passed the flags "wb+", that is, as a binary file in read/write mode. If the file already exists, tmpfile() will truncate it to size zero, possibly in an attempt to assuage the security concerns mentioned earlier regarding the race condition that exists between the selection of a supposedly unique filename and the subsequent opening of the selected file. However, this behavior clearly does not solve the function's security problems. First, an attacker can pre-create the file with relaxed access-permissions that will likely be retained by the file opened by tmpfile(). Furthermore, on Unix based systems if the attacker pre-creates the file as a link to another important file, the application may use its possibly elevated permissions to truncate that file, thereby doing damage on behalf of the attacker. Finally, if tmpfile() does create a new file, the access permissions applied to that file will vary from one operating system to another, which can leave application data vulnerable even if an attacker is unable to predict the filename to be used in advance. Finally, mkstemp() is a reasonably safe way create temporary files. It will attempt to create and open a unique file based on a filename template provided by the user combined with a series of randomly generated characters. If it is unable to create such a file, it will fail and return -1. On modern systems the file is opened using mode 0600, which means the file will be secure from tampering unless the user explicitly changes its access permissions. However, mkstemp() still suffers from the use of predictable file names and can leave an application vulnerable to denial of service attacks if an attacker causes mkstemp() to fail by predicting and pre-creating the filenames to be used.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	361	Time and State	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Category	376	Temporary File Issues	**Development Concepts (primary)699**
ChildOf	Weakness Class	668	Exposure of Resource to Wrong Sphere	**Research Concepts (primary)1000**
ParentOf	Weakness Base	378	Creation of Temporary File With Insecure Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Base	379	Creation of Temporary File in Directory with Incorrect Permissions	**Research Concepts (primary)1000**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
7 Pernicious Kingdoms			Insecure Temporary File

References

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 23, "Creating Temporary Files Securely" Page 682. 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		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Relationships, Other Notes, Taxonomy Mappings		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2010-02-16	CWE Content Team	MITRE	Internal
	updated References		

BACK TO TOP

Use of sizeof() on a Pointer Type

Weakness ID: 467 *(Weakness Variant)* **Status:** Draft

Description

Description Summary

The code calls sizeof() on a malloced pointer type, which always returns the wordsize/8. This can produce an unexpected result if the programmer intended to determine how much memory has been allocated.

Time of Introduction

- Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

Scope	Effect
Integrity	This error can often cause one to allocate a buffer that is much smaller than what is needed, leading to resultant weaknesses such as buffer overflows.

Likelihood of Exploit

High

Demonstrative Examples

Example 1

Care should be taken to ensure sizeof returns the size of the data structure itself, and not the size of the pointer to the data structure.

In this example, sizeof(foo) returns the size of the pointer.

(Bad Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(foo));
```

In this example, sizeof(*foo) returns the size of the data structure and not the size of the pointer.

(Good Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(*foo));
```

Example 2

This example defines a fixed username and password. The AuthenticateUser() function is intended to accept a username and a password from an untrusted user, and check to ensure that it matches the username and password. If the username and password match, AuthenticateUser() is intended to indicate that authentication succeeded.

(Bad Code)

```
/* Ignore CWE-259 (hard-coded password) and CWE-309 (use of password system for authentication) for this example. */

char *username = "admin";
char *pass = "password";

int AuthenticateUser(char *inUser, char *inPass) {
```

```
printf("Sizeof username = %d\n", sizeof(username));
printf("Sizeof pass = %d\n", sizeof(pass));

if (strncmp(username, inUser, sizeof(username))) {
printf("Auth failure of username using sizeof\n");
return(AUTH_FAIL);
}
/* Because of CWE-467, the sizeof returns 4 on many platforms and architectures. */
if (! strncmp(pass, inPass, sizeof(pass))) {
printf("Auth success of password using sizeof\n");
return(AUTH_SUCCESS);
}
else {
printf("Auth fail of password using sizeof\n");
return(AUTH_FAIL);
}
}

int main (int argc, char **argv)
{
int authResult;

if (argc < 3) {
ExitError("Usage: Provide a username and password");
}
authResult = AuthenticateUser(argv[1], argv[2]);
if (authResult != AUTH_SUCCESS) {
ExitError("Authentication failed");
}
else {
DoAuthenticatedTask(argv[1]);
}
}
```

In AuthenticateUser(), because sizeof() is applied to a parameter with an array type, the sizeof() call might return 4 on many modern architectures. As a result, the strncmp() call only checks the first four characters of the input password, resulting in a partial comparison (CWE-187), leading to improper authentication (CWE-287).

Because of the partial comparison, any of these passwords would still cause authentication to succeed for the "admin" user:

(Attack)

```
pass5
passABCDEFGH
passWORD
```

Because only 4 characters are checked, this significantly reduces the search space for an attacker, making brute force attacks more feasible.

The same problem also applies to the username, so values such as "adminXYZ" and "administrator" will succeed for the username.

Potential Mitigations

Phase: Implementation

Use expressions such as "sizeof(*pointer)" instead of "sizeof(pointer)", unless you intend to run sizeof() on a pointer type to gain some platform independence or if you are allocating a variable on the stack.

Other Notes

The use of sizeof() on a pointer can sometimes generate useful information. An obvious case is to find out the wordsize on a platform. More often than not, the appearance of sizeof(pointer) indicates a bug.

Weakness Ordinalities

Ordinality	Description
Primary	*(where the weakness exists independent of other weaknesses)*

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	465	Pointer Issues	**Development Concepts (primary)699**
ChildOf	Weakness Class	682	Incorrect Calculation	**Research Concepts (primary)1000**
ChildOf	Category	737	CERT C Secure Coding Section 03 - Expressions (EXP)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
ChildOf	Category	740	CERT C Secure Coding Section 06 - Arrays (ARR)	Weaknesses Addressed by the CERT C Secure Coding Standard734
CanPrecede	Weakness Base	131	Incorrect Calculation of Buffer Size	Research Concepts1000

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Use of sizeof() on a pointer type
CERT C Secure Coding	ARR01-C		Do not apply the sizeof operator to a pointer when taking the size of an array
CERT C Secure Coding	EXP01-C		Do not take the size of a pointer to determine the size of the pointed-to type

White Box Definitions

A weakness where code path has:

1. end statement that passes an identity of a dynamically allocated memory resource to a sizeof operator
2. start statement that allocates the dynamically allocated memory resource

References

Robert Seacord. "EXP01-A. Do not take the sizeof a pointer to determine the size of a type". <https://www.securecoding.cert.org/confluence/display/seccode/EXP01-A.+Do+not+take+the+sizeof+a+pointer+to+determine+the+size+of+a+type>.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	CLASP		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		

BACK TO TOP

Improper Validation of Array Index

Weakness ID: 129 *(Weakness Base)* **Status:** Draft

Description

Description Summary

The product uses untrusted input when calculating or using an array index, but the product does not validate or incorrectly validates the index to ensure the index references a valid position within the array.

Alternate Terms

out-of-bounds array index

index-out-of-range

array index underflow

Time of Introduction

- Implementation

Applicable Platforms

Languages

C: *(Often)*

C++: *(Often)*

Language-independent

Common Consequences

Scope	Effect
Integrity Availability	Unchecked array indexing will very likely result in the corruption of relevant memory and perhaps instructions, leading to a crash, if the values are outside of the valid memory area.
Integrity	If the memory corrupted is data, rather than instructions, the system will continue to function with improper values.
Confidentiality Integrity	Unchecked array indexing can also trigger out-of-bounds read or write operations, or operations on the wrong objects; i.e., "buffer overflows" are not always the result. This may result in the exposure or modification of sensitive data.
Integrity	If the memory accessible by the attacker can be effectively controlled, it may be possible to execute arbitrary code, as with a standard buffer overflow and possibly without the use of large inputs if a precise index can be controlled.
Integrity Availability Confidentiality	A single fault could allow either an overflow (CWE-788) or underflow (CWE-786) of the array index. What happens next will depend on the type of operation being performed out of bounds, but can expose sensitive information, cause a system crash, or possibly lead to arbitrary code execution.

Likelihood of Exploit

High

Detection Methods

Automated Static Analysis

This weakness can often be detected using automated static analysis tools. Many modern tools use data flow analysis or constraint-based techniques to minimize the number of false positives.

Automated static analysis generally does not account for environmental considerations when reporting out-of-bounds memory operations. This can make it difficult for users to determine which warnings should be investigated first. For example, an analysis tool might report array index errors that originate from command line arguments in a program that is not expected to run with setuid or other special privileges.

Effectiveness: High

This is not a perfect solution, since 100% accuracy and coverage are not feasible.

Automated Dynamic Analysis

This weakness can be detected using dynamic tools and techniques that interact with the software using large test suites with many diverse inputs, such as fuzz testing (fuzzing), robustness testing, and fault injection. The software's operation may slow down, but it should not become unstable, crash, or generate incorrect results.

Black Box

Black box methods might not get the needed code coverage within limited time constraints, and a dynamic test might not produce any noticeable side effects even if it is successful.

Demonstrative Examples

Example 1

The following C/C++ example retrieves the sizes of messages for a pop3 mail server. The message sizes are retrieved from a socket that returns in a buffer the message number and the message size, the message number (num) and size (size) are extracted from the buffer and the message size is placed into an array using the message number for the array index.

(Bad Code)

Example Language: **C**

```
/* capture the sizes of all messages */
int getsizes(int sock, int count, int *sizes) {
...
char buf[BUFFER_SIZE];
int ok;
int num, size;

// read values from socket and added to sizes array
while ((ok = gen_recv(sock, buf, sizeof(buf))) == 0)
{

// continue read from socket until buf only contains '.'
if (DOTLINE(buf))
break;
else if (sscanf(buf, "%d %d", &num, &size) == 2)
sizes[num - 1] = size;
}
...
}
```

In this example the message number retrieved from the buffer could be a value that is outside the allowable range of indices for the array and could possibly be a negative number. Without proper validation of the value to be used for the array index an array overflow could occur and could potentially lead to unauthorized access to memory addresses and system crashes. The value of the array index should be validated to ensure that it is within the allowable range of indices for the array as in the following code.

(Good Code)

Example Language: **C**

```
/* capture the sizes of all messages */
int getsizes(int sock, int count, int *sizes) {
...
char buf[BUFFER_SIZE];
int ok;
int num, size;

// read values from socket and added to sizes array
while ((ok = gen_recv(sock, buf, sizeof(buf))) == 0)
{

// continue read from socket until buf only contains '.'
if (DOTLINE(buf))
```

```
break;
else if (sscanf(buf, "%d %d", &num, &size) == 2) {
if (num > 0 && num <= (unsigned)count)
sizes[num - 1] = size;
else
/* warn about possible attempt to induce buffer overflow */
report(stderr, "Warning: ignoring bogus data for message sizes returned by server.\n");
}
}
...
}
```

Example 2

In the code snippet below, an unchecked integer value is used to reference an object in an array.

(Bad Code)

Example Language: **Java**

```
public String getValue(int index) {
return array[index];
}
```

If index is outside of the range of the array, this may result in an ArrayIndexOutOfBounds Exception being raised.

Example 3

In the following Java example the method displayProductSummary is called from a Web service servlet to retrieve product summary information for display to the user. The servlet obtains the integer value of the product number from the user and passes it to the displayProductSummary method. The displayProductSummary method passes the integer value of the product number to the getProductSummary method which obtains the product summary from the array object containing the project summaries using the integer value of the product number as the array index.

(Bad Code)

Example Language: **Java**

```
// Method called from servlet to obtain product information
public String displayProductSummary(int index) {

String productSummary = new String("");

try {
String productSummary = getProductSummary(index);

} catch (Exception ex) {...}

return productSummary;
}

public String getProductSummary(int index) {
return products[index];
}
```

In this example the integer value used as the array index that is provided by the user may be outside the allowable range of indices for the array which may provide unexpected results or may comes the application to fail. The integer value used for the array index should be validated to ensure that it is within the allowable range of indices for the array as in the following code.

(Good Code)

Example Language: **Java**

```
// Method called from servlet to obtain product information
public String displayProductSummary(int index) {

String productSummary = new String("");
```

```
try {
String productSummary = getProductSummary(index);

} catch (Exception ex) {...}

return productSummary;
}

public String getProductSummary(int index) {
String productSummary = "";

if ((index >= 0) && (index < MAX_PRODUCTS)) {
productSummary = products[index];
}
else {
System.err.println("index is out of bounds");
throw new IndexOutOfBoundsException();
}

return productSummary;
}
```

An alternative in Java would be to use one of the collection objects such as ArrayList that will automatically generate an exception if an attempt is made to access an array index that is out of bounds.

(Good Code)

Example Language: **Java**

```
ArrayList productArray = new ArrayList(MAX_PRODUCTS);
...
try {
productSummary = (String) productArray.get(index);
} catch (IndexOutOfBoundsException ex) {...}
```

Observed Examples

Reference	Description
CVE-2005-0369	large ID in packet used as array index
CVE-2001-1009	negative array index as argument to POP LIST command
CVE-2003-0721	Integer signedness error leads to negative array index
CVE-2004-1189	product does not properly track a count and a maximum number, which can lead to resultant array index overflow.
CVE-2007-5756	chain: device driver for packet-capturing software allows access to an unintended IOCTL with resultant array index error.

Potential Mitigations

Phase: Architecture and Design

Strategies: Input Validation; Libraries or Frameworks

Use an input validation framework such as Struts or the OWASP ESAPI Validation API. If you use Struts, be mindful of weaknesses covered by the CWE-101 category.

Phase: Architecture and Design

For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side, in order to avoid CWE-602. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the server.

Even though client-side checks provide minimal benefits with respect to server-side security, they are still useful. First, they can support intrusion detection. If the server receives input that should have been rejected by the client, then it may be an indication of an attack. Second, client-side error-checking can provide helpful feedback to the user about the expectations for valid input. Third, there may be a reduction in server-side processing time for accidental input errors, although this is typically a small savings.

Phase: Requirements

Strategy: Language Selection

Use a language with features that can automatically mitigate or eliminate out-of-bounds indexing errors.

For example, Ada allows the programmer to constrain the values of a variable and languages such as Java and Ruby will allow the programmer to handle exceptions when an out-of-bounds index is accessed.

Phase: Implementation

Strategy: Input Validation

Assume all input is malicious. Use an "accept known good" input validation strategy (i.e., use a whitelist). Reject any input that does not strictly conform to specifications, or transform it into something that does. Use a blacklist to reject any unexpected inputs and detect potential attacks.

When accessing a user-controlled array index, use a stringent range of values that are within the target array. Make sure that you do not allow negative values to be used. That is, verify the minimum as well as the maximum of the range of acceptable values.

Phase: Implementation

Be especially careful to validate your input when you invoke code that crosses language boundaries, such as from an interpreted language to native code. This could create an unexpected interaction between the language boundaries. Ensure that you are not violating any of the expectations of the language with which you are interfacing. For example, even though Java may not be susceptible to buffer overflows, providing a large argument in a call to native code might trigger an overflow.

Weakness Ordinalities

Ordinality	Description
Resultant	The most common condition situation leading to unchecked array indexing is the use of loop index variables as buffer indexes. If the end condition for the loop is subject to a flaw, the index can grow or shrink unbounded, therefore causing a buffer overflow or underflow. Another common situation leading to this condition is the use of a function's return value, or the resulting value of a calculation directly as an index in to a buffer.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	20	Improper Input Validation	**Development Concepts (primary)699 Research Concepts (primary)1000**
ChildOf	Category	189	Numeric Errors	Development Concepts699
ChildOf	Category	633	Weaknesses that Affect Memory	**Resource-specific Weaknesses (primary)631**
ChildOf	Category	738	CERT C Secure Coding Section 04 - Integers (INT)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
ChildOf	Category	740	CERT C Secure Coding Section 06 - Arrays (ARR)	Weaknesses Addressed by the CERT C Secure Coding Standard734
ChildOf	Category	802	2010 Top 25 - Risky Resource Management	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
CanPrecede	Weakness Class	119	Failure to Constrain Operations within the Bounds of a Memory Buffer	Research Concepts1000
CanPrecede	Weakness Variant	789	Uncontrolled Memory Allocation	Research Concepts1000
PeerOf	Weakness Base	124	Buffer Underwrite ('Buffer Underflow')	Research Concepts1000

Theoretical Notes

An improperly validated array index might lead directly to the always-incorrect behavior of "access of array using out-of-bounds index."

Affected Resources

- Memory

f Causal Nature

Explicit

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Unchecked array indexing
PLOVER			INDEX - Array index overflow
CERT C Secure Coding	ARR00-C		Understand how arrays work
CERT C Secure Coding	ARR30-C		Guarantee that array indices are within the valid range
CERT C Secure Coding	ARR38-C		Do not add or subtract an integer to a pointer if the resulting value does not refer to a valid array element
CERT C Secure Coding	INT32-C		Ensure that operations on signed integers do not result in overflow

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
100	Overflow Buffers	

References

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 5, "Array Indexing Errors" Page 144. 2nd Edition. Microsoft. 2002.

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	CLASP		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Sean Eidemiller	Cigital	External
	added/updated demonstrative examples		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Alternate Terms, Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Description, Name, Relationships		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Observed Examples, Other Notes, Potential Mitigations, Theoretical Notes, Weakness Ordinalities		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Demonstrative Examples, Detection Factors, Likelihood of Exploit, Potential Mitigations, References, Related Attack Patterns, Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Related Attack Patterns		
Previous Entry Names			
Change Date	**Previous Entry Name**		
2009-10-29	Unchecked Array Indexing		

BACK TO TOP

Improper Access Control (Authorization)

Weakness ID: 285 *(Weakness Class)* **Status:** Draft

Description

Description Summary

The software does not perform or incorrectly performs access control checks across all potential execution paths.

Extended Description

When access control checks are not applied consistently - or not at all - users are able to access data or perform actions that they should not be allowed to perform. This can lead to a wide range of problems, including information leaks, denial of service, and arbitrary code execution.

Alternate Terms

AuthZ:	"AuthZ" is typically used as an abbreviation of "authorization" within the web application security community. It is also distinct from "AuthC," which is an abbreviation of "authentication." The use of "Auth" as an abbreviation is discouraged, since it could be used for either authentication or authorization.

Time of Introduction

- Architecture and Design
- Implementation
- Operation

Applicable Platforms

Languages

Language-independent

Technology Classes

Web-Server: *(Often)*

Database-Server: *(Often)*

Modes of Introduction

A developer may introduce authorization weaknesses because of a lack of understanding about the underlying technologies. For example, a developer may assume that attackers cannot modify certain inputs such as headers or cookies.

Authorization weaknesses may arise when a single-user application is ported to a multi-user environment.

Common Consequences

Scope	Effect
Confidentiality	An attacker could read sensitive data, either by reading the data directly from a data store that is not properly restricted, or by accessing insufficiently-protected, privileged functionality to read the data.
Integrity	An attacker could modify sensitive data, either by writing the data directly to a data store that is not properly restricted, or by accessing insufficiently-protected, privileged functionality to write the data.
Integrity	An attacker could gain privileges by modifying or reading critical data directly, or by accessing insufficiently-protected, privileged functionality.

Likelihood of Exploit

High

Detection Methods

Automated Static Analysis

Automated static analysis is useful for detecting commonly-used idioms for authorization. A tool may be able to analyze related configuration files, such as .htaccess in Apache web servers, or detect the usage of commonly-used authorization libraries.

Generally, automated static analysis tools have difficulty detecting custom authorization schemes. In addition, the software's design may include some functionality that is accessible to any user and does not require an authorization check; an automated technique that detects the absence of authorization may report false positives.

Effectiveness: Limited

Automated Dynamic Analysis

Automated dynamic analysis may find many or all possible interfaces that do not require authorization, but manual analysis is required to determine if the lack of authorization violates business logic

Manual Analysis

This weakness can be detected using tools and techniques that require manual (human) analysis, such as penetration testing, threat modeling, and interactive tools that allow the tester to record and modify an active session.

Specifically, manual static analysis is useful for evaluating the correctness of custom authorization mechanisms.

Effectiveness: Moderate

These may be more effective than strictly automated techniques. This is especially the case with weaknesses that are related to design and business rules. However, manual efforts might not achieve desired code coverage within limited time constraints.

Demonstrative Examples

Example 1

The following program could be part of a bulletin board system that allows users to send private messages to each other. This program intends to authenticate the user before deciding whether a private message should be displayed. Assume that LookupMessageObject() ensures that the $id argument is numeric, constructs a filename based on that id, and reads the message details from that file. Also assume that the program stores all private messages for all users in the same directory.

(Bad Code)

Example Language: **Perl**

```
sub DisplayPrivateMessage {
my($id) = @_;
my $Message = LookupMessageObject($id);
print "From: " . encodeHTML($Message->{from}) . "<br>\n";
print "Subject: " . encodeHTML($Message->{subject}) . "\n";
print "<hr>\n";
print "Body: " . encodeHTML($Message->{body}) . "\n";
}

my $q = new CGI;
# For purposes of this example, assume that CWE-309 and
# CWE-523 do not apply.
if (! AuthenticateUser($q->param('username'), $q->param('password'))) {
ExitError("invalid username or password");
}

my $id = $q->param('id');
DisplayPrivateMessage($id);
```

While the program properly exits if authentication fails, it does not ensure that the message is addressed to the user. As a result, an authenticated attacker could provide any arbitrary identifier and read private messages that were intended for other users.

One way to avoid this problem would be to ensure that the "to" field in the message object matches the username of the authenticated user.

Observed Examples

Reference	Description
CVE-2009-3168	Web application does not restrict access to admin scripts, allowing authenticated users to reset administrative passwords.

CVE-2009-2960	Web application does not restrict access to admin scripts, allowing authenticated users to modify passwords of other users.
CVE-2009-3597	Web application stores database file under the web root with insufficient access control (CWE-219), allowing direct request.
CVE-2009-2282	Terminal server does not check authorization for guest access.
CVE-2009-3230	Database server does not use appropriate privileges for certain sensitive operations.
CVE-2009-2213	Gateway uses default "Allow" configuration for its authorization settings.
CVE-2009-0034	Chain: product does not properly interpret a configuration option for a system group, allowing users to gain privileges.
CVE-2008-6123	Chain: SNMP product does not properly parse a configuration option for which hosts are allowed to connect, allowing unauthorized IP addresses to connect.
CVE-2008-5027	System monitoring software allows users to bypass authorization by creating custom forms.
CVE-2008-7109	Chain: reliance on client-side security (CWE-602) allows attackers to bypass authorization using a custom client.
CVE-2008-3424	Chain: product does not properly handle wildcards in an authorization policy list, allowing unintended access.
CVE-2009-3781	Content management system does not check access permissions for private files, allowing others to view those files.
CVE-2008-4577	ACL-based protection mechanism treats negative access rights as if they are positive, allowing bypass of intended restrictions.
CVE-2008-6548	Product does not check the ACL of a page accessed using an "include" directive, allowing attackers to read unauthorized files.
CVE-2007-2925	Default ACL list for a DNS server does not set certain ACLs, allowing unauthorized DNS queries.
CVE-2006-6679	Product relies on the X-Forwarded-For HTTP header for authorization, allowing unintended access by spoofing the header.
CVE-2005-3623	OS kernel does not check for a certain privilege before setting ACLs for files.
CVE-2005-2801	Chain: file-system code performs an incorrect comparison (CWE-697), preventing defauls ACLs from being properly applied.
CVE-2001-1155	Chain: product does not properly check the result of a reverse DNS lookup because of operator precedence (CWE-783), allowing bypass of DNS-based access restrictions.

Potential Mitigations

Phase: Architecture and Design

Divide your application into anonymous, normal, privileged, and administrative areas. Reduce the attack surface by carefully mapping roles with data and functionality. Use role-based access control (RBAC) to enforce the roles at the appropriate boundaries.

Note that this approach may not protect against horizontal authorization, i.e., it will not protect a user from attacking others with the same role.

Phase: Architecture and Design

Ensure that you perform access control checks related to your business logic. These checks may be different than the access control checks that you apply to more generic resources such as files, connections, processes, memory, and database records. For example, a database may restrict access for medical records to a specific database user, but each record might only be intended to be accessible to the patient and the patient's doctor.

Phase: Architecture and Design

Strategy: Libraries or Frameworks

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness

easier to avoid.

For example, consider using authorization frameworks such as the JAAS Authorization Framework and the OWASP ESAPI Access Control feature.

Phase: Architecture and Design

For web applications, make sure that the access control mechanism is enforced correctly at the server side on every page. Users should not be able to access any unauthorized functionality or information by simply requesting direct access to that page.

One way to do this is to ensure that all pages containing sensitive information are not cached, and that all such pages restrict access to requests that are accompanied by an active and authenticated session token associated with a user who has the required permissions to access that page.

Phases: System Configuration; Installation

Use the access control capabilities of your operating system and server environment and define your access control lists accordingly. Use a "default deny" policy when defining these ACLs.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	254	Security Features	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Weakness Class	284	Access Control (Authorization) Issues	**Development Concepts (primary)699 Research Concepts (primary)1000**
ChildOf	Category	721	OWASP Top Ten 2007 Category A10 - Failure to Restrict URL Access	**Weaknesses in OWASP Top Ten (2007) (primary)629**
ChildOf	Category	723	OWASP Top Ten 2004 Category A2 - Broken Access Control	**Weaknesses in OWASP Top Ten (2004) (primary)711**
ChildOf	Category	753	2009 Top 25 - Porous Defenses	**Weaknesses in the 2009 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)750**
ChildOf	Category	803	2010 Top 25 - Porous Defenses	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
ParentOf	Weakness Variant	219	Sensitive Data Under Web Root	**Research Concepts (primary)1000**
ParentOf	Weakness Base	551	Incorrect Behavior Order: Authorization Before Parsing and Canonicalization	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Class	638	Failure to Use Complete Mediation	Research Concepts1000
ParentOf	Weakness Base	804	Guessable CAPTCHA	**Development Concepts (primary)699 Research Concepts (primary)1000**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
7 Pernicious Kingdoms			Missing Access Control
OWASP Top Ten 2007	A10	CWE More Specific	Failure to Restrict URL Access
OWASP Top Ten 2004	A2	CWE More Specific	Broken Access Control

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
1	Accessing Functionality Not Properly Constrained by ACLs	
13	Subverting Environment Variable Values	

17	Accessing, Modifying or Executing Executable Files
87	Forceful Browsing
39	Manipulating Opaque Client-based Data Tokens
45	Buffer Overflow via Symbolic Links
51	Poison Web Service Registry
59	Session Credential Falsification through Prediction
60	Reusing Session IDs (aka Session Replay)
77	Manipulating User-Controlled Variables
76	Manipulating Input to File System Calls
104	Cross Zone Scripting

References

NIST. "Role Based Access Control and Role Based Security". <http://csrc.nist.gov/groups/SNS/rbac/>.

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 4, "Authorization" Page 114; Chapter 6, "Determining Appropriate Access Control" Page 171. 2nd Edition. Microsoft. 2002.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	7 Pernicious Kingdoms		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-15		Veracode	External
	Suggested OWASP Top Ten 2004 mapping		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Relationships, Other Notes, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences, Description, Likelihood of Exploit, Name, Other Notes, Potential Mitigations, References, Relationships		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Potential Mitigations		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Description, Related Attack Patterns		
2009-07-27	CWE Content Team	MITRE	Internal
	updated Relationships		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Type		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Demonstrative Examples, Detection Factors, Modes of Introduction, Observed Examples, Relationships		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Alternate Terms, Detection Factors, Potential Mitigations, References, Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Potential Mitigations		

Previous Entry Names

Change Date	Previous Entry Name
2009-01-12	Missing or Inconsistent Access Control

BACK TO TOP

Incorrect Permission Assignment for Critical Resource

Weakness ID: 732 *(Weakness Class)* **Status:** Draft

Description

Description Summary

The software specifies permissions for a security-critical resource in a way that allows that resource to be read or modified by unintended actors.

Extended Description

When a resource is given a permissions setting that provides access to a wider range of actors than required, it could lead to the disclosure of sensitive information, or the modification of that resource by unintended parties. This is especially dangerous when the resource is related to program configuration, execution or sensitive user data.

Time of Introduction

- Architecture and Design
- Implementation
- Installation
- Operation

Applicable Platforms

Languages

Language-independent

Modes of Introduction

The developer may set loose permissions in order to minimize problems when the user first runs the program, then create documentation stating that permissions should be tightened. Since system administrators and users do not always read the documentation, this can result in insecure permissions being left unchanged.

The developer might make certain assumptions about the environment in which the software runs - e.g., that the software is running on a single-user system, or the software is only accessible to trusted administrators. When the software is running in a different environment, the permissions become a problem.

Common Consequences

Scope	Effect
Confidentiality	An attacker may be able to read sensitive information from the associated resource, such as credentials or configuration information stored in a file.
Integrity	An attacker may be able to modify critical properties of the associated resource to gain privileges, such as replacing a world-writable executable with a Trojan horse.
Availability	An attacker may be able to destroy or corrupt critical data in the associated resource, such as deletion of records from a database.

Likelihood of Exploit

Medium to High

Detection Methods

Automated Static Analysis

Automated static analysis may be effective in detecting permission problems for system resources such as files, directories, shared memory, device interfaces, etc. Automated techniques may be able to detect the use of library functions that modify permissions, then analyze function calls for arguments that contain potentially insecure values.

However, since the software's intended security policy might allow loose permissions for certain operations (such as publishing a file on a web server), automated static analysis may produce some false positives - i.e., warnings that do not have any security consequences or require any code changes.

When custom permissions models are used - such as defining who can read messages in a particular forum in a bulletin board system - these can be difficult to detect using automated static analysis. It may be possible to define custom signatures that

identify any custom functions that implement the permission checks and assignments.

Automated Dynamic Analysis

Automated dynamic analysis may be effective in detecting permission problems for system resources such as files, directories, shared memory, device interfaces, etc.

However, since the software's intended security policy might allow loose permissions for certain operations (such as publishing a file on a web server), automated dynamic analysis may produce some false positives - i.e., warnings that do not have any security consequences or require any code changes.

When custom permissions models are used - such as defining who can read messages in a particular forum in a bulletin board system - these can be difficult to detect using automated dynamic analysis. It may be possible to define custom signatures that identify any custom functions that implement the permission checks and assignments.

Manual Static Analysis

Manual static analysis may be effective in detecting the use of custom permissions models and functions. The code could then be examined to identifying usage of the related functions. Then the human analyst could evaluate permission assignments in the context of the intended security model of the software.

Manual Dynamic Analysis

Manual dynamic analysis may be effective in detecting the use of custom permissions models and functions. The program could then be executed with a focus on exercising code paths that are related to the custom permissions. Then the human analyst could evaluate permission assignments in the context of the intended security model of the software.

Fuzzing

Fuzzing is not effective in detecting this weakness.

Demonstrative Examples

Example 1

The following code sets the umask of the process to 0 before creating a file and writing "Hello world" into the file.

(Bad Code)

Example Language: **C**

```
#define OUTFILE "hello.out"

umask(0);
FILE *out;
/* Ignore CWE-59 (link following) for brevity */
out = fopen(OUTFILE, "w");
if (out) {
fprintf(out, "hello world!\n");
fclose(out);
}
```

After running this program on a UNIX system, running the "ls -l" command might return the following output:

(Result)

```
-rw-rw-rw- 1 username 13 Nov 24 17:58 hello.out
```

The "rw-rw-rw-" string indicates that the owner, group, and world (all users) can read the file and write to it.

Example 2

The following code snippet might be used as a monitor to periodically record whether a web site is alive. To ensure that the file can always be modified, the code uses chmod() to make the file world-writable.

(Bad Code)

Example Language: **Perl**

```
$fileName = "secretFile.out";

if (-e $fileName) {
chmod 0777, $fileName;
}
```

```
my $outFH;
if (! open($outFH, ">>$fileName")) {
ExitError("Couldn't append to $fileName: $!");
}
my $dateString = FormatCurrentTime();
my $status = IsHostAlive("cwe.mitre.org");
print $outFH "$dateString cwe status: $status!\n";
close($outFH);
```

The first time the program runs, it might create a new file that inherits the permissions from its environment. A file listing might look like:

(Result)

```
-rw-r--r-- 1 username 13 Nov 24 17:58 secretFile.out
```

This listing might occur when the user has a default umask of 022, which is a common setting. Depending on the nature of the file, the user might not have intended to make it readable by everyone on the system.

The next time the program runs, however - and all subsequent executions - the chmod will set the file's permissions so that the owner, group, and world (all users) can read the file and write to it:

(Result)

```
-rw-rw-rw- 1 username 13 Nov 24 17:58 secretFile.out
```

Perhaps the programmer tried to do this because a different process uses different permissions that might prevent the file from being updated.

Example 3

The following command recursively sets world-readable permissions for a directory and all of its children:

(Bad Code)

Example Language: **Shell**

```
chmod -R ugo+r DIRNAME
```

If this command is run from a program, the person calling the program might not expect that all the files under the directory will be world-readable. If the directory is expected to contain private data, this could become a security problem.

Observed Examples

Reference	Description
CVE-2009-3482	Anti-virus product sets insecure "Everyone: Full Control" permissions for files under the "Program Files" folder, allowing attackers to replace executables with Trojan horses.
CVE-2009-3897	Product creates directories with 0777 permissions at installation, allowing users to gain privileges and access a socket used for authentication.
CVE-2009-3489	Photo editor installs a service with an insecure security descriptor, allowing users to stop or start the service, or execute commands as SYSTEM.
CVE-2009-3289	Library function copies a file to a new target and uses the source file's permissions for the target, which is incorrect when the source file is a symbolic link, which typically has 0777 permissions.
CVE-2009-0115	Device driver uses world-writable permissions for a socket file, allowing attackers to inject arbitrary commands.
CVE-2009-1073	LDAP server stores a cleartext password in a world-readable file.
CVE-2009-0141	Terminal emulator creates TTY devices with world-writable permissions, allowing an attacker to write to the terminals of other users.

CVE-2008-0662	VPN product stores user credentials in a registry key with "Everyone: Full Control" permissions, allowing attackers to steal the credentials.
CVE-2008-0322	Driver installs its device interface with "Everyone: Write" permissions.
CVE-2009-3939	Driver installs a file with world-writable permissions.
CVE-2009-3611	Product changes permissions to 0777 before deleting a backup; the permissions stay insecure for subsequent backups.
CVE-2007-6033	Product creates a share with "Everyone: Full Control" permissions, allowing arbitrary program execution.
CVE-2007-5544	Product uses "Everyone: Full Control" permissions for memory-mapped files (shared memory) in inter-process communication, allowing attackers to tamper with a session.
CVE-2005-4868	Database product uses read/write permissions for everyone for its shared memory, allowing theft of credentials.
CVE-2004-1714	Security product uses "Everyone: Full Control" permissions for its configuration files.
CVE-2001-0006	"Everyone: Full Control" permissions assigned to a mutex allows users to disable network connectivity.
CVE-2002-0969	Chain: database product contains buffer overflow that is only reachable through a .ini configuration file - which has "Everyone: Full Control" permissions.

Potential Mitigations

Phase: Implementation

When using a critical resource such as a configuration file, check to see if the resource has insecure permissions (such as being modifiable by any regular user), and generate an error or even exit the software if there is a possibility that the resource could have been modified by an unauthorized party.

Phase: Architecture and Design

Divide your application into anonymous, normal, privileged, and administrative areas. Reduce the attack surface by carefully defining distinct user groups, privileges, and/or roles. Map these against data, functionality, and the related resources. Then set the permissions accordingly. This will allow you to maintain more fine-grained control over your resources.

Phases: Implementation; Installation

During program startup, explicitly set the default permissions or umask to the most restrictive setting possible. Also set the appropriate permissions during program installation. This will prevent you from inheriting insecure permissions from any user who installs or runs the program.

Phase: System Configuration

For all configuration files, executables, and libraries, make sure that they are only readable and writable by the software's administrator.

Phase: Documentation

Do not suggest insecure configuration changes in your documentation, especially if those configurations can extend to resources and other software that are outside the scope of your own software.

Phase: Installation

Do not assume that the system administrator will manually change the configuration to the settings that you recommend in the manual.

Phase: Testing

Use tools and techniques that require manual (human) analysis, such as penetration testing, threat modeling, and interactive tools that allow the tester to record and modify an active session. These may be more effective than strictly automated techniques. This is especially the case with weaknesses that are related to design and business rules.

Phase: Testing

Use monitoring tools that examine the software's process as it interacts with the operating system and the network. This technique is useful in cases when source code is unavailable, if the software was not developed by you, or if you want to verify that the build phase did not introduce any new weaknesses. Examples include debuggers that directly attach to the running process; system-call tracing utilities such as truss (Solaris) and strace (Linux); system activity monitors such as FileMon, RegMon, Process Monitor, and other Sysinternals utilities (Windows); and sniffers and protocol analyzers that monitor network traffic.

Attach the monitor to the process and watch for library functions or system calls on OS resources such as files, directories, and shared memory. Examine the arguments to these calls to infer which permissions are being used.

Note that this technique is only useful for permissions issues related to system resources. It is not likely to detect application-level business rules that are related to permissions, such as if a user of a blog system marks a post as "private," but the blog system inadvertently marks it as "public."

Phases: Testing; System Configuration

Ensure that your software runs properly under the Federal Desktop Core Configuration (FDCC) or an equivalent hardening configuration guide, which many organizations use to limit the attack surface and potential risk of deployed software.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	275	Permission Issues	**Development Concepts (primary)699**
ChildOf	Weakness Class	668	Exposure of Resource to Wrong Sphere	**Research Concepts (primary)1000**
ChildOf	Category	753	2009 Top 25 - Porous Defenses	**Weaknesses in the 2009 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)750**
ChildOf	Category	803	2010 Top 25 - Porous Defenses	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
RequiredBy	Compound Element: Composite	689	Permission Race Condition During Resource Copy	Research Concepts1000
ParentOf	Weakness Variant	276	Incorrect Default Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	277	Insecure Inherited Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	278	Insecure Preserved Inherited Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	279	Incorrect Execution-Assigned Permissions	**Research Concepts (primary)1000**
ParentOf	Weakness Base	281	Improper Preservation of Permissions	**Research Concepts (primary)1000**

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
232	Exploitation of Privilege/Trust	
1	Accessing Functionality Not Properly Constrained by ACLs	
17	Accessing, Modifying or Executing Executable Files	
60	Reusing Session IDs (aka Session Replay)	
61	Session Fixation	
62	Cross Site Request Forgery (aka Session Riding)	
122	Exploitation of Authorization	
180	Exploiting Incorrectly Configured Access Control Security Levels	
234	Hijacking a privileged process	

References

Mark Dowd, John McDonald and Justin Schuh. "The Art of Software Security Assessment". Chapter 9, "File Permissions." Page 495.. 1st Edition. Addison Wesley. 2006.

John Viega and Gary McGraw. "Building Secure Software". Chapter 8, "Access Control." Page 194.. 1st Edition. Addison-Wesley. 2002.

Maintenance Notes

The relationships between privileges, permissions, and actors (e.g. users and groups) need further refinement within the Research view. One complication is that these concepts apply to two different pillars, related to control of resources (CWE-664) and protection mechanism failures (CWE-396).

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
2008-09-08			Internal CWE Team
	new weakness-focused entry for Research view.		
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2009-01-12	CWE Content Team	MITRE	Internal
	updated Description, Likelihood of Exploit, Name, Potential Mitigations, Relationships		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Potential Mitigations, Related Attack Patterns		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Name		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Demonstrative Examples, Detection Factors, Modes of Introduction, Observed Examples, Potential Mitigations, References		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Potential Mitigations, Related Attack Patterns		
Previous Entry Names			
Change Date	**Previous Entry Name**		
2009-01-12	Insecure Permission Assignment for Resource		
2009-05-27	Insecure Permission Assignment for Critical Resource		

BACK TO TOP

Exposure of System Data to Unauthorized Control Sphere

Risk

What might happen

System data can provide attackers with valuable insights on systems and services they are targeting - any type of system data, from service version to operating system fingerprints, can assist attackers to hone their attack, correlate data with known vulnerabilities or focus efforts on developing new attacks against specific technologies.

Cause

How does it happen

System data is read and subsequently exposed where it might be read by untrusted entities.

General Recommendations

How to avoid it

Consider the implications of exposure of the specified input, and expected level of access to the specified output. If not required, consider removing this code, or modifying exposed information to exclude potentially sensitive system data.

Source Code Examples

Java

Leaking Environment Variables in JSP Web-Page

```
String envVarValue = System.getenv(envVar);
if (envVarValue == null) {
    out.println("Environment variable is not defined:");
    out.println(System.getenv());
} else {
    //[..]
};
```

TOCTOU

Risk

What might happen

At best, a Race Condition may cause errors in accuracy, overidden values or unexpected behavior that may result in denial-of-service. At worst, it may allow attackers to retrieve data or bypass security processes by replaying a controllable Race Condition until it plays out in their favor.

Cause

How does it happen

Race Conditions occur when a public, single instance of a resource is used by multiple concurrent logical processes. If the these logical processes attempt to retrieve and update the resource without a timely management system, such as a lock, a Race Condition will occur.

An example for when a Race Condition occurs is a resource that may return a certain value to a process for further editing, and then updated by a second process, resulting in the original process' data no longer being valid. Once the original process edits and updates the incorrect value back into the resource, the second process' update has been overwritten and lost.

General Recommendations

How to avoid it

When sharing resources between concurrent processes across the application ensure that these resources are either thread-safe, or implement a locking mechanism to ensure expected concurrent activity.

Source Code Examples

Java

Different Threads Increment and Decrement The Same Counter Repeatedly, Resulting in a Race Condition

```
	public static int counter = 0;
	public static void start() throws InterruptedException {
		incrementCounter ic;
		decrementCounter dc;
		while(counter == 0) {
			counter = 0;
			ic = new incrementCounter();
			dc = new decrementCounter();
			ic.start();
			dc.start();
			ic.join();
			dc.join();
		}
		System.out.println(counter); //Will stop and return either -1 or 1 due to race condition over counter
	}

	public static class incrementCounter extends Thread {
	    public void run() {
		counter++;
	    }
```

```
    }

    public static class decrementCounter extends Thread {
        public void run() {
           counter--;
        }
    }
```

Different Threads Increment and Decrement The Same Thread-Safe Counter Repeatedly, Never Resulting in a Race Condition

```
    public static int counter = 0;
    public static Object lock = new Object();

    public static void start() throws InterruptedException {
           incrementCounter ic;
           decrementCounter dc;
           while(counter == 0) { // because of proper locking, this condition is never false
                  counter = 0;
                  ic = new incrementCounter();
                  dc = new decrementCounter();
                  ic.start();
                  dc.start();
                  ic.join();
                  dc.join();
           }
           System.out.println(counter); // Never reached
    }

    public static class incrementCounter extends Thread {
        public void run() {
           synchronized (lock) {
                  counter++;
           }
        }
    }

    public static class decrementCounter extends Thread {
        public void run() {
           synchronized (lock) {
                  counter--;
           }
        }
    }
```

Scanned Languages

Language	Hash Number	Change Date
CPP	4541647240435660	6/19/2024
Common	0105849645654507	6/19/2024