freebsd-src-4 Scan Report

Project Name	freebsd-src-4
Scan Start	Saturday, June 22, 2024 9:07:04 AM
Preset	Checkmarx Default
Scan Time	00h:05m:32s
Lines Of Code Scanned	81439
Files Scanned	31
Report Creation Time	Saturday, June 22, 2024 9:42:02 AM
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084
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

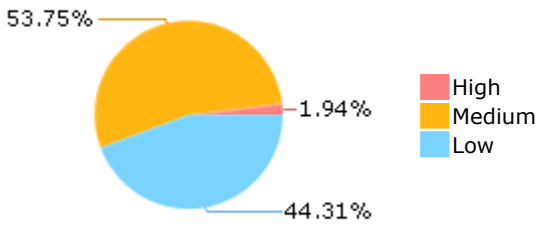
Results Limit

Results limit per query was set to 50

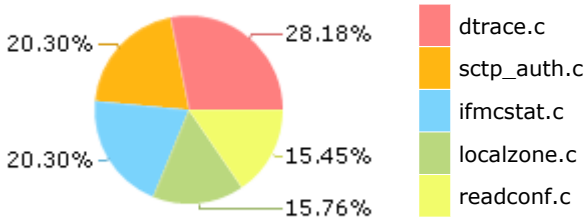
Selected Queries

Selected queries are listed in Result Summary

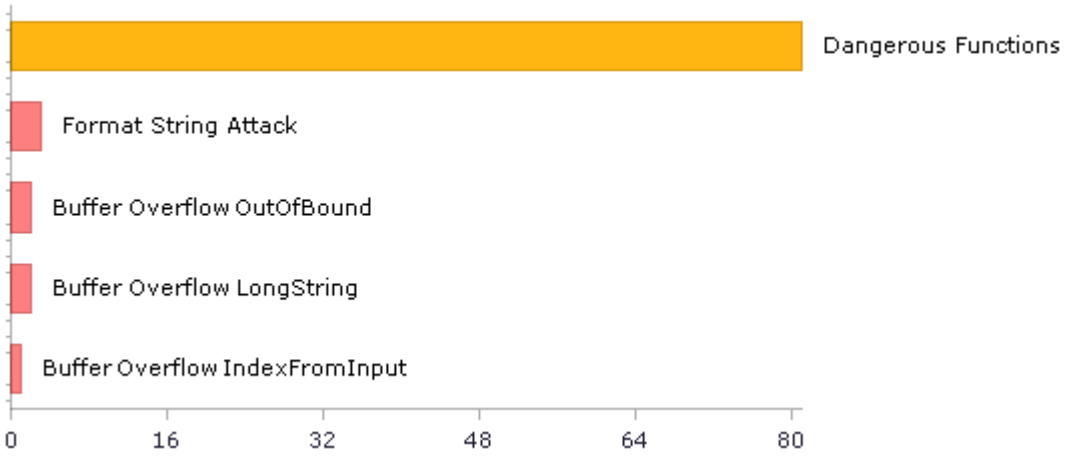
Result Summary



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	93	60
A2-Broken Authentication	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	44	44
A3-Sensitive Data Exposure	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	App. Specific	5	4
A4-XML External Entities (XXE)	App. Specific	AVERAGE	COMMON	EASY	SEVERE	App. Specific	0	0
A5-Broken Access Control*	App. Specific	AVERAGE	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A6-Security Misconfiguration	App. Specific	EASY	WIDESPREAD	EASY	MODERATE	App. Specific	0	0
A7-Cross-Site Scripting (XSS)	App. Specific	EASY	WIDESPREAD	EASY	MODERATE	App. Specific	0	0
A8-Insecure Deserialization	App. Specific	DIFFICULT	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A9-Using Components with Known Vulnerabilities*	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	MODERATE	App. Specific	81	81
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	2	2
A3-Cross-Site Scripting (XSS)	EXTERNAL, INTERNAL, ADMIN USERS	AVERAGE	VERY WIDESPREAD	EASY	MODERATE	AFFECTED DATA AND SYSTEM	0	0
A4-Insecure Direct Object References	SYSTEM USERS	EASY	COMMON	EASY	MODERATE	EXPOSED DATA	0	0
A5-Security Misconfiguration	EXTERNAL, INTERNAL, ADMIN USERS	EASY	COMMON	EASY	MODERATE	ALL DATA AND SYSTEM	0	0
A6-Sensitive Data Exposure	EXTERNAL, INTERNAL, ADMIN USERS, USERS BROWSERS	DIFFICULT	UNCOMMON	AVERAGE	SEVERE	EXPOSED DATA	2	2
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	81	81
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	0	0
PCI DSS (3.2) - 6.5.2 - Buffer overflows	61	55
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.	7	7
Audit And Accountability*	Organizations must: (i) create, protect, and retain information system audit records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful, unauthorized, or inappropriate information system activity; and (ii) ensure that the actions of individual information system users can be uniquely traced to those users so they can be held accountable for their actions.	0	0
Configuration Management	Organizations must: (i) establish and maintain baseline configurations and inventories of organizational information systems (including hardware, software, firmware, and documentation) throughout the respective system development life cycles; and (ii) establish and enforce security configuration settings for information technology products employed in organizational information systems.	23	22
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.	46	41
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.	2	2
System And Communications Protection	Organizations must: (i) monitor, control, and protect organizational communications (i.e., information transmitted or received by organizational information systems) at the external boundaries and key internal boundaries of the information systems; and (ii) employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational information systems.	0	0
System And Information Integrity	Organizations must: (i) identify, report, and correct information and information system flaws in a timely manner; (ii) provide protection from malicious code at appropriate locations within organizational information systems; and (iii) monitor information system security alerts and advisories and take appropriate actions in response.	1	1

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

Scan Summary - NIST SP 800-53

Category	Issues Found	Best Fix Locations
AC-12 Session Termination (P2)	0	0
AC-3 Access Enforcement (P1)	62	62
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)	3	2
SC-17 Public Key Infrastructure Certificates (P1)	0	0
SC-18 Mobile Code (P2)	0	0
SC-23 Session Authenticity (P1)*	9	4
SC-28 Protection of Information at Rest (P1)	2	2
SC-4 Information in Shared Resources (P1)	2	2
SC-5 Denial of Service Protection (P1)*	72	28
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*	15	9
SI-11 Error Handling (P2)*	22	22
SI-15 Information Output Filtering (P0)	0	0
SI-16 Memory Protection (P1)	7	7

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

Scan Summary - 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

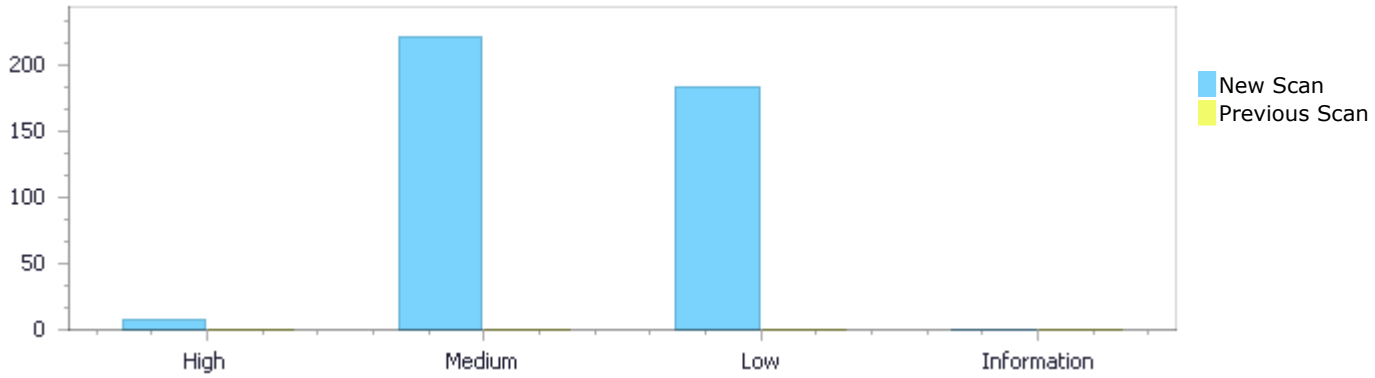
Scan Summary - Custom

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	8	222	183	0	413
Recurrent Issues	0	0	0	0	0
Total	8	222	183	0	413
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	8	222	183	0	413
Urgent	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0
Total	8	222	183	0	413

Result Summary

Vulnerability Type	Occurrences	Severity
Format String Attack	3	High
Buffer Overflow LongString	2	High
Buffer Overflow OutOfBound	2	High
Buffer Overflow IndexFromInput	1	High
Dangerous Functions	81	Medium

Buffer Overflow boundcpy WrongSizeParam	47	Medium
MemoryFree on StackVariable	37	Medium
Use of Uninitialized Pointer	20	Medium
Use of Zero Initialized Pointer	10	Medium
Memory Leak	8	Medium
Double Free	7	Medium
Wrong Size t Allocation	4	Medium
Inadequate Encryption Strength	3	Medium
Heap Inspection	2	Medium
Use of Uninitialized Variable	2	Medium
Integer Overflow	1	Medium
Improper Resource Access Authorization	35	Low
NULL Pointer Dereference	32	Low
Unchecked Return Value	22	Low
Exposure of System Data to Unauthorized Control Sphere	20	Low
Sizeof Pointer Argument	20	Low
Use of Sizeof On a Pointer Type	16	Low
Reliance on DNS Lookups in a Decision	9	Low
TOCTOU	9	Low
Incorrect Permission Assignment For Critical Resources	7	Low
Heuristic Buffer Overflow malloc	6	Low
Inconsistent Implementations	4	Low
Use Of Hardcoded Password	2	Low
Unchecked Array Index	1	Low

10 Most Vulnerable Files

High and Medium Vulnerabilities

File Name	Issues Found
freebsd-src-4/sctp_auth.c	66
freebsd-src-4/readconf.c	45
freebsd-src-4/localzone.c	43
freebsd-src-4/dtrace.c	32
freebsd-src-4/print-802_11.c	13
freebsd-src-4/trans_udp.c	9
freebsd-src-4/ifmcstat.c	6
freebsd-src-4/srp_vfy.c	5
freebsd-src-4/rand_lib.c	4
freebsd-src-4/ed25519_ref10.c	3

Scan Results Details

Format String Attack

Query Path:
CPP\Cx\CPP Buffer Overflow\Format String Attack 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

Format String Attack\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=3
Status	New

Method udp_init_port at line 106 of freebsd-src-4/trans_udp.c receives the "creating socket: %m" value from user input. This value is then used to construct a "format string" "creating socket: %m", which is provided as an argument to a string formatting function in udp_init_port method of freebsd-src-4/trans_udp.c at line 106.

	Source	Destination
File	freebsd-src-4/trans_udp.c	freebsd-src-4/trans_udp.c
Line	114	114
Object	"creating socket: %m"	"creating socket: %m"

Code Snippet
File Name freebsd-src-4/trans_udp.c
Method udp_init_port(struct tport *tp)

```
....
114.              syslog(LOG_ERR, "creating UDP socket: %m");
```

Format String Attack\Path 2:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=4
Status	New

Method udp_init_port at line 106 of freebsd-src-4/trans_udp.c receives the "setsockopt(IP_RECVDSTADDR): %m" value from user input. This value is then used to construct a "format string" "setsockopt(IP_RECVDSTADDR): %m", which is provided as an argument to a string formatting function in udp_init_port method of freebsd-src-4/trans_udp.c at line 106.

	Source	Destination
File	freebsd-src-4/trans_udp.c	freebsd-src-4/trans_udp.c

Line	127	127
Object	"setsockopt(IP_RECVDSTADDR): %m"	"setsockopt(IP_RECVDSTADDR): %m"

Code Snippet
File Name freebsd-src-4/trans_udp.c
Method udp_init_port(struct tport *tp)

```
....
127.                    syslog(LOG_ERR, "setsockopt(IP_RECVDSTADDR):
%m");
```

Format String Attack\Path 3:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=5
Status	New

Method udp_init_port at line 106 of freebsd-src-4/trans_udp.c receives the "bind: %s:%u %m" value from user input. This value is then used to construct a "format string" "bind: %s:%u %m", which is provided as an argument to a string formatting function in udp_init_port method of freebsd-src-4/trans_udp.c at line 106.

	Source	Destination
File	freebsd-src-4/trans_udp.c	freebsd-src-4/trans_udp.c
Line	140	140
Object	"bind: %s:%u %m"	"bind: %s:%u %m"

Code Snippet
File Name freebsd-src-4/trans_udp.c
Method udp_init_port(struct tport *tp)

```
....
140.                syslog(LOG_ERR, "bind: %s:%u %m",
inet_ntoa(addr.sin_addr),
```

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=1070094&projectid=700

	84&pathid=1
Status	New

The size of the buffer used by execute_in_shell in argv, at line 535 of freebsd-src-4/readconf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that match_cfg_line passes to "%llu", at line 591 of freebsd-src-4/readconf.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	699	561
Object	"%llu"	argv

Code Snippet

File Name freebsd-src-4/readconf.c

Method match_cfg_line(Options *options, char **condition, struct passwd *pw,

```
....
699.                    snprintf(uidstr, sizeof(uidstr), "%llu",
```

File Name freebsd-src-4/readconf.c

Method execute_in_shell(const char *cmd)

```
....
561.                argv[2] = xstrdup(cmd);
```

Buffer Overflow LongString\Path 2:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=2
Status	New

The size of the buffer used by parse_forward in Address, at line 2891 of freebsd-src-4/readconf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that process_config_line_depth passes to "%s:%s", at line 948 of freebsd-src-4/readconf.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	1553	2914
Object	"%s:%s"	Address

Code Snippet

File Name freebsd-src-4/readconf.c

Method process_config_line_depth(Options *options, struct passwd *pw, const char *host,

```
....
1553.                          snprintf(fwdarg, sizeof(fwdarg), "%s:%s",
arg,
```

File Name	freebsd-src-4/readconf.c
Method	parse_forward(struct Forward *fwd, const char *fwdspec, int dynamicfwd, int remotefwd)

```
....
2914.             if (parse_fwd_field(&cp, &fwdargs[i]) != 0)
```

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=1070094&projectid=70084&pathid=7
Status	New

The size of the buffer used by sc25519_is_canonical in i, at line 1881 of freebsd-src-4/ed25519_ref10.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sc25519_is_canonical passes to L, at line 1881 of freebsd-src-4/ed25519_ref10.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1884	1896
Object	L	i

Code Snippet

File Name	freebsd-src-4/ed25519_ref10.c
Method	sc25519_is_canonical(const unsigned char *s)

```
....
1884.      static const unsigned char L[32] = {
....
1896.          n &= ((s[i] ^ L[i]) - 1) >> 8;
```

Buffer Overflow OutOfBound\Path 2:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=8
Status	New

The size of the buffer used by sc25519_is_canonical in i, at line 1881 of freebsd-src-4/ed25519_ref10.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sc25519_is_canonical passes to L, at line 1881 of freebsd-src-4/ed25519_ref10.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1884	1895
Object	L	i

Code Snippet
File Name freebsd-src-4/ed25519_ref10.c
Method sc25519_is_canonical(const unsigned char *s)

```
....
1884.      static const unsigned char L[32] = {
....
1895.          c |= ((s[i] - L[i]) >> 8) & n;
```

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=1070094&projectid=70084&pathid=6
Status	New

The size of the buffer used by main in i, at line 1307 of freebsd-src-4/dtrace.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 1307 of freebsd-src-4/dtrace.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1307	1816
Object	argc	i

Code Snippet

File Name	freebsd-src-4/dtrace.c
Method	main(int argc, char *argv[])

```
....
1307.  main(int argc, char *argv[])
....
1816.                        dtrace_dof_create(g_dtp, g_cmdv[i].dc_prog,
0), i);
```

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=1070094&projectid=70084&pathid=210
Status	New

The dangerous function, alloca, was found in use at line 1307 in freebsd-src-4/dtrace.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1858	1858
Object	alloca	alloca

Code Snippet

File Name	freebsd-src-4/dtrace.c
Method	main(int argc, char *argv[])

```
....
1858.                    char **objv = alloca(g_cmdc * sizeof (char *));
```

Dangerous Functions\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=211
Status	New

The dangerous function, alloca, was found in use at line 1105 in freebsd-src-4/dtrace.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1144	1144
Object	alloca	alloca

Code Snippet
File Name freebsd-src-4/dtrace.c
Method chew(const dtrace_probedata_t *data, void *arg)

```
....
1144.                    name = alloca(len);
```

Dangerous Functions\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=212
Status	New

The dangerous function, alloca, was found in use at line 1105 in freebsd-src-4/dtrace.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1150	1150
Object	alloca	alloca

Code Snippet
File Name freebsd-src-4/dtrace.c
Method chew(const dtrace_probedata_t *data, void *arg)

```
....
1150.                    name = alloca(len);
```

Dangerous Functions\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=213
Status	New

The dangerous function, memcpy, was found in use at line 1959 in freebsd-src-4/ed25519_ref10.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c

Line	1971	1971
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/ed25519_ref10.c
Method ge25519_from_uniform(unsigned char s[32], const unsigned char r[32])

```
....
1971.       memcpy(s, r, 32);
```

Dangerous Functions\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=214
Status	New

The dangerous function, memcpy, was found in use at line 396 in freebsd-src-4/localzone.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	418	418
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/localzone.c
Method rrset_insert_rr(struct regional* region, struct packed_rrset_data* pd,

```
....
418.              memcpy(pd->rr_len+1, oldlen,
```

Dangerous Functions\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=215
Status	New

The dangerous function, memcpy, was found in use at line 396 in freebsd-src-4/localzone.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	420	420
Object	memcpy	memcpy

Code Snippet	
File Name	freebsd-src-4/localzone.c
Method	rrset_insert_rr(struct regional* region, struct packed_rrset_data* pd,

```
....
420.                memcpy(pd->rr_ttl+1, oldttl,
```

Dangerous Functions\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=216
Status	New

The dangerous function, memcpy, was found in use at line 396 in freebsd-src-4/localzone.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	422	422
Object	memcpy	memcpy

Code Snippet	
File Name	freebsd-src-4/localzone.c
Method	rrset_insert_rr(struct regional* region, struct packed_rrset_data* pd,

```
....
422.                memcpy(pd->rr_data+1, olddata,
```

Dangerous Functions\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=217
Status	New

The dangerous function, memcpy, was found in use at line 1137 in freebsd-src-4/print-802_11.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/print-802_11.c	freebsd-src-4/print-802_11.c
Line	1173	1173
Object	memcpy	memcpy

Code Snippet	
File Name	freebsd-src-4/print-802_11.c

Method	parse_elements(netdissect_options *ndo,

```
....
1173.                    memcpy(&ssid, p + offset, 2);
```

Dangerous Functions\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=218
Status	New

The dangerous function, memcpy, was found in use at line 1137 in freebsd-src-4/print-802_11.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/print-802_11.c	freebsd-src-4/print-802_11.c
Line	1179	1179
Object	memcpy	memcpy

Code Snippet	
File Name	freebsd-src-4/print-802_11.c
Method	parse_elements(netdissect_options *ndo,

```
....
1179.                          memcpy(&ssid.ssid, p + offset,
ssid.length);
```

Dangerous Functions\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=219
Status	New

The dangerous function, memcpy, was found in use at line 1137 in freebsd-src-4/print-802_11.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/print-802_11.c	freebsd-src-4/print-802_11.c
Line	1197	1197
Object	memcpy	memcpy

Code Snippet	
File Name	freebsd-src-4/print-802_11.c
Method	parse_elements(netdissect_options *ndo,

```
....
1197.                    memcpy(&challenge, p + offset, 2);
```

Dangerous Functions\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=220
Status	New

The dangerous function, memcpy, was found in use at line 1137 in freebsd-src-4/print-802_11.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/print-802_11.c	freebsd-src-4/print-802_11.c
Line	1204	1204
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/print-802_11.c
Method parse_elements(netdissect_options *ndo,

```
....
1204.                          memcpy(&challenge.text, p + offset,
```

Dangerous Functions\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=221
Status	New

The dangerous function, memcpy, was found in use at line 1137 in freebsd-src-4/print-802_11.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/print-802_11.c	freebsd-src-4/print-802_11.c
Line	1223	1223
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/print-802_11.c
Method parse_elements(netdissect_options *ndo,

```
....
1223.                    memcpy(&rates, p + offset, 2);
```

Dangerous Functions\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=222
Status	New

The dangerous function, memcpy, was found in use at line 1137 in freebsd-src-4/print-802_11.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/print-802_11.c	freebsd-src-4/print-802_11.c
Line	1229	1229
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/print-802_11.c
Method parse_elements(netdissect_options *ndo,

```
....
1229.                          memcpy(&rates.rate, p + offset,
rates.length);
```

Dangerous Functions\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=223
Status	New

The dangerous function, memcpy, was found in use at line 1137 in freebsd-src-4/print-802_11.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/print-802_11.c	freebsd-src-4/print-802_11.c
Line	1255	1255
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/print-802_11.c
Method parse_elements(netdissect_options *ndo,

```
....
1255.                   memcpy(&ds, p + offset, 2);
```

Dangerous Functions\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=224
Status	New

The dangerous function, memcpy, was found in use at line 1137 in freebsd-src-4/print-802_11.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/print-802_11.c	freebsd-src-4/print-802_11.c
Line	1279	1279
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/print-802_11.c
Method parse_elements(netdissect_options *ndo,

```
....
1279.                   memcpy(&cf, p + offset, 2);
```

Dangerous Functions\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=225
Status	New

The dangerous function, memcpy, was found in use at line 1137 in freebsd-src-4/print-802_11.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/print-802_11.c	freebsd-src-4/print-802_11.c
Line	1312	1312
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/print-802_11.c
Method parse_elements(netdissect_options *ndo,

```
....
1312.                    memcpy(&tim, p + offset, 2);
```

Dangerous Functions\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=226
Status	New

The dangerous function, memcpy, was found in use at line 1137 in freebsd-src-4/print-802_11.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/print-802_11.c	freebsd-src-4/print-802_11.c
Line	1331	1331
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/print-802_11.c
Method parse_elements(netdissect_options *ndo,

```
....
1331.                    memcpy(tim.bitmap, p + offset, tim.length - 3);
```

Dangerous Functions\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=227
Status	New

The dangerous function, memcpy, was found in use at line 1368 in freebsd-src-4/print-802_11.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/print-802_11.c	freebsd-src-4/print-802_11.c
Line	1382	1382
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/print-802_11.c
Method handle_beacon(netdissect_options *ndo,

```
....
1382.       memcpy(&pbody.timestamp, p, IEEE802_11_TSTAMP_LEN);
```

Dangerous Functions\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=228
Status	New

The dangerous function, memcpy, was found in use at line 1473 in freebsd-src-4/print-802_11.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/print-802_11.c	freebsd-src-4/print-802_11.c
Line	1493	1493
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/print-802_11.c
Method handle_reassoc_request(netdissect_options *ndo,

```
....
1493.       memcpy(&pbody.ap, p+offset, IEEE802_11_AP_LEN);
```

Dangerous Functions\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=229
Status	New

The dangerous function, memcpy, was found in use at line 1534 in freebsd-src-4/print-802_11.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/print-802_11.c	freebsd-src-4/print-802_11.c
Line	1548	1548
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/print-802_11.c
Method handle_probe_response(netdissect_options *ndo,

```
....
1548.       memcpy(&pbody.timestamp, p, IEEE802_11_TSTAMP_LEN);
```

Dangerous Functions\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=230
Status	New

The dangerous function, memcpy, was found in use at line 624 in freebsd-src-4/rand_lib.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/rand_lib.c	freebsd-src-4/rand_lib.c
Line	648	648
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/rand_lib.c
Method static int rand_pool_grow(RAND_POOL *pool, size_t len)

```
....
648.           memcpy(p, pool->buffer, pool->len);
```

Dangerous Functions\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=231
Status	New

The dangerous function, memcpy, was found in use at line 724 in freebsd-src-4/rand_lib.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/rand_lib.c	freebsd-src-4/rand_lib.c
Line	759	759
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/rand_lib.c
Method int rand_pool_add(RAND_POOL *pool,

```
....
759.          memcpy(pool->buffer + pool->len, buffer, len);
```

Dangerous Functions\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=232
Status	New

The dangerous function, memcpy, was found in use at line 84 in freebsd-src-4/sctp_auth.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	96	96
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_copy_chunklist(sctp_auth_chklist_t *list)

```
....
96.   memcpy(new_list, list, sizeof(*new_list));
```

Dangerous Functions\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=233
Status	New

The dangerous function, memcpy, was found in use at line 331 in freebsd-src-4/sctp_auth.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	340	340
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_set_key(uint8_t *key, uint32_t keylen)

```
....
340.        memcpy(new_key->key, key, keylen);
```

Dangerous Functions\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=234
Status	New

The dangerous function, memcpy, was found in use at line 402 in freebsd-src-4/sctp_auth.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	429	429
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_compute_hashkey(sctp_key_t *key1, sctp_key_t *key2, sctp_key_t *shared)

```
....
429.                    memcpy(key_ptr, shared->key, shared->keylen);
```

Dangerous Functions\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=235
Status	New

The dangerous function, memcpy, was found in use at line 402 in freebsd-src-4/sctp_auth.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	433	433
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_compute_hashkey(sctp_key_t *key1, sctp_key_t *key2, sctp_key_t *shared)

```
....
433.                    memcpy(key_ptr, key1->key, key1->keylen);
```

Dangerous Functions\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=236
Status	New

The dangerous function, memcpy, was found in use at line 402 in freebsd-src-4/sctp_auth.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	437	437
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_compute_hashkey(sctp_key_t *key1, sctp_key_t *key2, sctp_key_t *shared)

```
....
437.                    memcpy(key_ptr, key2->key, key2->keylen);
```

Dangerous Functions\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=237
Status	New

The dangerous function, memcpy, was found in use at line 402 in freebsd-src-4/sctp_auth.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	442	442
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_compute_hashkey(sctp_key_t *key1, sctp_key_t *key2, sctp_key_t *shared)

```
....
442.                    memcpy(key_ptr, shared->key, shared->keylen);
```

Dangerous Functions\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=238
Status	New

The dangerous function, memcpy, was found in use at line 402 in freebsd-src-4/sctp_auth.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	446	446
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_compute_hashkey(sctp_key_t *key1, sctp_key_t *key2, sctp_key_t *shared)

```
....
446.                    memcpy(key_ptr, key2->key, key2->keylen);
```

Dangerous Functions\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=239
Status	New

The dangerous function, memcpy, was found in use at line 402 in freebsd-src-4/sctp_auth.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	450	450
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_compute_hashkey(sctp_key_t *key1, sctp_key_t *key2, sctp_key_t *shared)

```
....
450.                    memcpy(key_ptr, key1->key, key1->keylen);
```

Dangerous Functions\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=240
Status	New

The dangerous function, memcpy, was found in use at line 749 in freebsd-src-4/sctp_auth.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	759	759
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_serialize_hmaclist(sctp_hmaclist_t *list, uint8_t *ptr)

```
....
759.                memcpy(ptr, &hmac_id, sizeof(hmac_id));
```

Dangerous Functions\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=241
Status	New

The dangerous function, memcpy, was found in use at line 915 in freebsd-src-4/sctp_auth.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	949	949
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_hmac(uint16_t hmac_algo, uint8_t *key, uint32_t keylen,

```
....
949.        memcpy(ipad, key, keylen);
```

Dangerous Functions\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=242
Status	New

The dangerous function, memcpy, was found in use at line 915 in freebsd-src-4/sctp_auth.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	950	950
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_hmac(uint16_t hmac_algo, uint8_t *key, uint32_t keylen,

```
....
950.        memcpy(opad, key, keylen);
```

Dangerous Functions\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=243
Status	New

The dangerous function, memcpy, was found in use at line 975 in freebsd-src-4/sctp_auth.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1009	1009
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_hmac_m(uint16_t hmac_algo, uint8_t *key, uint32_t keylen,

```
....
1009.       memcpy(ipad, key, keylen);
```

Dangerous Functions\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=244
Status	New

The dangerous function, memcpy, was found in use at line 975 in freebsd-src-4/sctp_auth.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1010	1010
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_hmac_m(uint16_t hmac_algo, uint8_t *key, uint32_t keylen,

```
....
1010.       memcpy(opad, key, keylen);
```

Dangerous Functions\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=245
Status	New

The dangerous function, memcpy, was found in use at line 1057 in freebsd-src-4/sctp_auth.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1084	1084
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_compute_hmac(uint16_t hmac_algo, sctp_key_t *key, uint8_t *text,

```
....
1084.               memcpy(key->key, temp, key->keylen);
```

Dangerous Functions\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=246
Status	New

The dangerous function, memcpy, was found in use at line 1092 in freebsd-src-4/sctp_auth.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1118	1118
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_compute_hmac_m(uint16_t hmac_algo, sctp_key_t *key, struct mbuf *m,

```
....
1118.               memcpy(key->key, temp, key->keylen);
```

Dangerous Functions\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=247
Status	New

The dangerous function, memcpy, was found in use at line 1362 in freebsd-src-4/sctp_auth.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1463	1463
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_auth_get_cookie_params(struct sctp_tcb *stcb, struct mbuf *m,

```
....
1463.                    memcpy(new_key->key, p_random, keylen);
```

Dangerous Functions\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=248
Status	New

The dangerous function, memcpy, was found in use at line 1362 in freebsd-src-4/sctp_auth.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1469	1469
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_auth_get_cookie_params(struct sctp_tcb *stcb, struct mbuf *m,

```
....
1469.                    memcpy(new_key->key + keylen, chunks,
```

Dangerous Functions\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=249
Status	New

The dangerous function, memcpy, was found in use at line 1362 in freebsd-src-4/sctp_auth.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1475	1475
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_auth_get_cookie_params(struct sctp_tcb *stcb, struct mbuf *m,

```
....
1475.                   memcpy(new_key->key + keylen, hmacs,
```

Dangerous Functions\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=250
Status	New

The dangerous function, memcpy, was found in use at line 1590 in freebsd-src-4/sctp_auth.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1691	1691
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_handle_auth(struct sctp_tcb *stcb, struct sctp_auth_chunk *auth,

```
....
1691.       memcpy(digest, auth->hmac, digestlen);
```

Dangerous Functions\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=251
Status	New

The dangerous function, memcpy, was found in use at line 220 in freebsd-src-4/trans_udp.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/trans_udp.c	freebsd-src-4/trans_udp.c
Line	246	246
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/trans_udp.c
Method udp_send(struct tport *tp, const u_char *buf, size_t len,

```
....
246.                memcpy(CMSG_DATA(cmsg), &p->dstaddr, sizeof(struct
in_addr));
```

Dangerous Functions\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=252
Status	New

The dangerous function, memcpy, was found in use at line 271 in freebsd-src-4/trans_udp.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/trans_udp.c	freebsd-src-4/trans_udp.c
Line	328	328
Object	memcpy	memcpy

Code Snippet
File Name freebsd-src-4/trans_udp.c
Method udp_recv(struct tport *tp, struct port_input *pi)

```
....
328.                    memcpy(&p->dstaddr, CMSG_DATA(cmsg),
```

Dangerous Functions\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=253
Status	New

The dangerous function, sprintf, was found in use at line 961 in freebsd-src-4/dtrace.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1036	1036
Object	sprintf	sprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method bufhandler(const dtrace_bufdata_t *bufdata, void *arg)

```
....
1036.                   (void) sprintf(buf, "%d (data: ", rec-
>dtrd_offset);
```

Dangerous Functions\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=254
Status	New

The dangerous function, strlen, was found in use at line 187 in freebsd-src-4/dtrace.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	194	194
Object	strlen	strlen

Code Snippet
File Name freebsd-src-4/dtrace.c
Method verror(const char *fmt, va_list ap)

```
....
194.        if (fmt[strlen(fmt) - 1] != '\n')
```

Dangerous Functions\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=255
Status	New

The dangerous function, strlen, was found in use at line 220 in freebsd-src-4/dtrace.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	236	236
Object	strlen	strlen

Code Snippet
File Name freebsd-src-4/dtrace.c
Method dfatal(const char *fmt, ...)

```
....
236.        if (fmt != NULL && fmt[strlen(fmt) - 1] != '\n') {
```

Dangerous Functions\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=256
Status	New

The dangerous function, strlen, was found in use at line 307 in freebsd-src-4/dtrace.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	310	310
Object	strlen	strlen

Code Snippet
File Name freebsd-src-4/dtrace.c
Method make_argv(char *s)

```
....
310.        char **argv = malloc(sizeof (char *) * (strlen(s) / 2 + 1));
```

Dangerous Functions\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=257
Status	New

The dangerous function, strlen, was found in use at line 328 in freebsd-src-4/dtrace.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	359	359
Object	strlen	strlen

Code Snippet
File Name freebsd-src-4/dtrace.c
Method dof_prune(const char *fname)

```
....
359.        len = strlen("dof-data-");
```

Dangerous Functions\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=258
Status	New

The dangerous function, strlen, was found in use at line 961 in freebsd-src-4/dtrace.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	990	990
Object	strlen	strlen

Code Snippet
File Name freebsd-src-4/dtrace.c
Method bufhandler(const dtrace_bufdata_t *bufdata, void *arg)

```
....
990.        BUFDUMPHDR(">>> Called buffer handler");
```

Dangerous Functions\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=259
Status	New

The dangerous function, strlen, was found in use at line 961 in freebsd-src-4/dtrace.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	991	991
Object	strlen	strlen

Code Snippet
File Name freebsd-src-4/dtrace.c
Method bufhandler(const dtrace_bufdata_t *bufdata, void *arg)

```
....
991.        BUFDUMPHDR("");
```

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=1070094&projectid=70084&pathid=51
Status	New

The size of the buffer used by sctp_serialize_hmaclist in hmac_id, at line 749 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_serialize_hmaclist passes to hmac_id, at line 749 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	759	759
Object	hmac_id	hmac_id

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_serialize_hmaclist(sctp_hmaclist_t *list, uint8_t *ptr)

```
....
759.              memcpy(ptr, &hmac_id, sizeof(hmac_id));
```

Buffer Overflow boundcpy WrongSizeParam\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=52
Status	New

The size of the buffer used by udp_send in in_addr, at line 220 of freebsd-src-4/trans_udp.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that udp_send passes to in_addr, at line 220 of freebsd-src-4/trans_udp.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/trans_udp.c	freebsd-src-4/trans_udp.c

Line	246	246
Object	in_addr	in_addr

Code Snippet
File Name freebsd-src-4/trans_udp.c
Method udp_send(struct tport *tp, const u_char *buf, size_t len,

```
....
246.                memcpy(CMSG_DATA(cmsg), &p->dstaddr, sizeof(struct
in_addr));
```

Buffer Overflow boundcpy WrongSizeParam\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=53
Status	New

The size of the buffer used by udp_recv in in_addr, at line 271 of freebsd-src-4/trans_udp.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that udp_recv passes to in_addr, at line 271 of freebsd-src-4/trans_udp.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/trans_udp.c	freebsd-src-4/trans_udp.c
Line	329	329
Object	in_addr	in_addr

Code Snippet
File Name freebsd-src-4/trans_udp.c
Method udp_recv(struct tport *tp, struct port_input *pi)

```
....
329.                        sizeof(struct in_addr));
```

Buffer Overflow boundcpy WrongSizeParam\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=54
Status	New

The size of the buffer used by local_data_answer in ->, at line 1465 of freebsd-src-4/localzone.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that local_data_answer passes to ->, at line 1465 of freebsd-src-4/localzone.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c

Line	1551	1551
Object	->	->

Code Snippet
File Name freebsd-src-4/localzone.c
Method local_data_answer(struct local_zone* z, struct module_env* env,

```
....
1551.                          sizeof(qinfo->local_alias->rrset->entry));
```

Buffer Overflow boundcpy WrongSizeParam\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=55
Status	New

The size of the buffer used by initialize_options in ->, at line 2332 of freebsd-src-4/readconf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that initialize_options passes to ->, at line 2332 of freebsd-src-4/readconf.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	2372	2372
Object	->	->

Code Snippet
File Name freebsd-src-4/readconf.c
Method initialize_options(Options * options)

```
....
2372.       memset(options->identity_keys, 0, sizeof(options-
>identity_keys));
```

Buffer Overflow boundcpy WrongSizeParam\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=56
Status	New

The size of the buffer used by initialize_options in ->, at line 2332 of freebsd-src-4/readconf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that initialize_options passes to ->, at line 2332 of freebsd-src-4/readconf.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	2374	2374

Object	->	->

Code Snippet
File Name freebsd-src-4/readconf.c
Method initialize_options(Options * options)

```
....
2374.       memset(options->certificates, 0, sizeof(options-
>certificates));
```

Buffer Overflow boundcpy WrongSizeParam\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=57
Status	New

The size of the buffer used by fill_default_options in options, at line 2471 of freebsd-src-4/readconf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that fill_default_options passes to options, at line 2471 of freebsd-src-4/readconf.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	2705	2705
Object	options	options

Code Snippet
File Name freebsd-src-4/readconf.c
Method fill_default_options(Options * options)

```
....
2705.                sizeof(*options->permitted_cnames));
```

Buffer Overflow boundcpy WrongSizeParam\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=58
Status	New

The size of the buffer used by sctp_notify_authentication in sctp_authkey_event, at line 1710 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_notify_authentication passes to sctp_authkey_event, at line 1710 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1738	1738

Object	sctp_authkey_event	sctp_authkey_event

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_notify_authentication(struct sctp_tcb *stcb, uint32_t indication,

```
....
1738.       memset(auth, 0, sizeof(struct sctp_authkey_event));
```

Buffer Overflow boundcpy WrongSizeParam\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=59
Status	New

The size of the buffer used by rrset_insert_rr in pd, at line 396 of freebsd-src-4/localzone.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that rrset_insert_rr passes to pd, at line 396 of freebsd-src-4/localzone.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	419	419
Object	pd	pd

Code Snippet
File Name freebsd-src-4/localzone.c
Method rrset_insert_rr(struct regional* region, struct packed_rrset_data* pd,

```
....
419.                    sizeof(*pd->rr_len)*(pd->count-1));
```

Buffer Overflow boundcpy WrongSizeParam\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=60
Status	New

The size of the buffer used by rrset_insert_rr in pd, at line 396 of freebsd-src-4/localzone.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that rrset_insert_rr passes to pd, at line 396 of freebsd-src-4/localzone.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	421	421
Object	pd	pd

Code Snippet	
File Name	freebsd-src-4/localzone.c
Method	rrset_insert_rr(struct regional* region, struct packed_rrset_data* pd,

```
....
421.                    sizeof(*pd->rr_ttl)*(pd->count-1));
```

Buffer Overflow boundcpy WrongSizeParam\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=61
Status	New

The size of the buffer used by rrset_insert_rr in pd, at line 396 of freebsd-src-4/localzone.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that rrset_insert_rr passes to pd, at line 396 of freebsd-src-4/localzone.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	423	423
Object	pd	pd

Code Snippet	
File Name	freebsd-src-4/localzone.c
Method	rrset_insert_rr(struct regional* region, struct packed_rrset_data* pd,

```
....
423.                    sizeof(*pd->rr_data)*(pd->count-1));
```

Buffer Overflow boundcpy WrongSizeParam\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=62
Status	New

The size of the buffer used by sctp_auth_get_cookie_params in num_chunks, at line 1362 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_auth_get_cookie_params passes to num_chunks, at line 1362 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1470	1470
Object	num_chunks	num_chunks

Code Snippet	
File Name	freebsd-src-4/sctp_auth.c

Method	sctp_auth_get_cookie_params(struct sctp_tcb *stcb, struct mbuf *m,

```
....
1470.                          sizeof(*chunks) + num_chunks);
```

Buffer Overflow boundcpy WrongSizeParam\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=63
Status	New

The size of the buffer used by sctp_auth_get_cookie_params in chunks, at line 1362 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_auth_get_cookie_params passes to chunks, at line 1362 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1470	1470
Object	chunks	chunks

Code Snippet	
File Name	freebsd-src-4/sctp_auth.c
Method	sctp_auth_get_cookie_params(struct sctp_tcb *stcb, struct mbuf *m,

```
....
1470.                          sizeof(*chunks) + num_chunks);
```

Buffer Overflow boundcpy WrongSizeParam\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=64
Status	New

The size of the buffer used by sctp_auth_get_cookie_params in hmacs_len, at line 1362 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_auth_get_cookie_params passes to hmacs_len, at line 1362 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1476	1476
Object	hmacs_len	hmacs_len

Code Snippet	
File Name	freebsd-src-4/sctp_auth.c
Method	sctp_auth_get_cookie_params(struct sctp_tcb *stcb, struct mbuf *m,

```
....
1476.                          sizeof(*hmacs) + hmacs_len);
```

Buffer Overflow boundcpy WrongSizeParam\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=65
Status	New

The size of the buffer used by sctp_auth_get_cookie_params in hmacs, at line 1362 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_auth_get_cookie_params passes to hmacs, at line 1362 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1476	1476
Object	hmacs	hmacs

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_auth_get_cookie_params(struct sctp_tcb *stcb, struct mbuf *m,

```
....
1476.                          sizeof(*hmacs) + hmacs_len);
```

Buffer Overflow boundcpy WrongSizeParam\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=66
Status	New

The size of the buffer used by local_rrset_remove_rr in num, at line 438 of freebsd-src-4/localzone.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that local_rrset_remove_rr passes to num, at line 438 of freebsd-src-4/localzone.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	449	449
Object	num	num

Code Snippet
File Name freebsd-src-4/localzone.c
Method local_rrset_remove_rr(struct packed_rrset_data* pd, size_t index)

```
....
449.                memmove(pd->rr_len+index, pd->rr_len+nexti,
sizeof(*pd->rr_len)*num);
```

Buffer Overflow boundcpy WrongSizeParam\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=67
Status	New

The size of the buffer used by local_rrset_remove_rr in pd, at line 438 of freebsd-src-4/localzone.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that local_rrset_remove_rr passes to pd, at line 438 of freebsd-src-4/localzone.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	449	449
Object	pd	pd

Code Snippet
File Name freebsd-src-4/localzone.c
Method local_rrset_remove_rr(struct packed_rrset_data* pd, size_t index)

```
....
449.                memmove(pd->rr_len+index, pd->rr_len+nexti,
sizeof(*pd->rr_len)*num);
```

Buffer Overflow boundcpy WrongSizeParam\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=68
Status	New

The size of the buffer used by local_rrset_remove_rr in num, at line 438 of freebsd-src-4/localzone.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that local_rrset_remove_rr passes to num, at line 438 of freebsd-src-4/localzone.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	450	450
Object	num	num

Code Snippet
File Name freebsd-src-4/localzone.c

Method	local_rrset_remove_rr(struct packed_rrset_data* pd, size_t index)

```
....
450.              memmove(pd->rr_ttl+index, pd->rr_ttl+nexti,
sizeof(*pd->rr_ttl)*num);
```

Buffer Overflow boundcpy WrongSizeParam\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=69
Status	New

The size of the buffer used by local_rrset_remove_rr in pd, at line 438 of freebsd-src-4/localzone.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that local_rrset_remove_rr passes to pd, at line 438 of freebsd-src-4/localzone.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	450	450
Object	pd	pd

Code Snippet	
File Name	freebsd-src-4/localzone.c
Method	local_rrset_remove_rr(struct packed_rrset_data* pd, size_t index)

```
....
450.              memmove(pd->rr_ttl+index, pd->rr_ttl+nexti,
sizeof(*pd->rr_ttl)*num);
```

Buffer Overflow boundcpy WrongSizeParam\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=70
Status	New

The size of the buffer used by local_rrset_remove_rr in num, at line 438 of freebsd-src-4/localzone.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that local_rrset_remove_rr passes to num, at line 438 of freebsd-src-4/localzone.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	451	451
Object	num	num

Code Snippet

File Name	freebsd-src-4/localzone.c
Method	local_rrset_remove_rr(struct packed_rrset_data* pd, size_t index)

```
....
451.              memmove(pd->rr_data+index, pd->rr_data+nexti,
sizeof(*pd->rr_data)*num);
```

Buffer Overflow boundcpy WrongSizeParam\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=71
Status	New

The size of the buffer used by local_rrset_remove_rr in pd, at line 438 of freebsd-src-4/localzone.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that local_rrset_remove_rr passes to pd, at line 438 of freebsd-src-4/localzone.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	451	451
Object	pd	pd

Code Snippet

File Name	freebsd-src-4/localzone.c
Method	local_rrset_remove_rr(struct packed_rrset_data* pd, size_t index)

```
....
451.              memmove(pd->rr_data+index, pd->rr_data+nexti,
sizeof(*pd->rr_data)*num);
```

Buffer Overflow boundcpy WrongSizeParam\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=72
Status	New

The size of the buffer used by local_data_find_tag_datas in d, at line 1354 of freebsd-src-4/localzone.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that local_data_find_tag_datas passes to d, at line 1354 of freebsd-src-4/localzone.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1422	1422
Object	d	d

Code Snippet	
File Name	freebsd-src-4/localzone.c
Method	local_data_find_tag_datas(const struct query_info* qinfo,

```
....
1422.                    memmove(d->rr_len, oldlen, d-
>count*sizeof(size_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=73
Status	New

The size of the buffer used by local_data_find_tag_datas in long, at line 1354 of freebsd-src-4/localzone.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that local_data_find_tag_datas passes to long, at line 1354 of freebsd-src-4/localzone.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1422	1422
Object	long	long

Code Snippet	
File Name	freebsd-src-4/localzone.c
Method	local_data_find_tag_datas(const struct query_info* qinfo,

```
....
1422.                    memmove(d->rr_len, oldlen, d-
>count*sizeof(size_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=74
Status	New

The size of the buffer used by local_data_find_tag_datas in d, at line 1354 of freebsd-src-4/localzone.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that local_data_find_tag_datas passes to d, at line 1354 of freebsd-src-4/localzone.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1423	1423
Object	d	d

Code Snippet
File Name freebsd-src-4/localzone.c
Method local_data_find_tag_datas(const struct query_info* qinfo,

```
....
1423.                    memmove(d->rr_data, olddata, d-
>count*sizeof(uint8_t*));
```

Buffer Overflow boundcpy WrongSizeParam\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=75
Status	New

The size of the buffer used by local_data_find_tag_datas in uint8_t, at line 1354 of freebsd-src-4/localzone.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that local_data_find_tag_datas passes to uint8_t, at line 1354 of freebsd-src-4/localzone.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1423	1423
Object	uint8_t	uint8_t

Code Snippet
File Name freebsd-src-4/localzone.c
Method local_data_find_tag_datas(const struct query_info* qinfo,

```
....
1423.                    memmove(d->rr_data, olddata, d-
>count*sizeof(uint8_t*));
```

Buffer Overflow boundcpy WrongSizeParam\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=76
Status	New

The size of the buffer used by local_data_find_tag_datas in d, at line 1354 of freebsd-src-4/localzone.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that local_data_find_tag_datas passes to d, at line 1354 of freebsd-src-4/localzone.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1424	1424

Object	d	d

Code Snippet
File Name freebsd-src-4/localzone.c
Method local_data_find_tag_datas(const struct query_info* qinfo,

```
....
1424.                    memmove(d->rr_ttl, oldttl, d-
>count*sizeof(time_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=77
Status	New

The size of the buffer used by local_data_find_tag_datas in time_t, at line 1354 of freebsd-src-4/localzone.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that local_data_find_tag_datas passes to time_t, at line 1354 of freebsd-src-4/localzone.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1424	1424
Object	time_t	time_t

Code Snippet
File Name freebsd-src-4/localzone.c
Method local_data_find_tag_datas(const struct query_info* qinfo,

```
....
1424.                    memmove(d->rr_ttl, oldttl, d-
>count*sizeof(time_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=78
Status	New

The size of the buffer used by parse_fwd_field in cp, at line 2825 of freebsd-src-4/readconf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that parse_fwd_field passes to cp, at line 2825 of freebsd-src-4/readconf.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	2861	2861

Object	cp	cp

Code Snippet
File Name freebsd-src-4/readconf.c
Method parse_fwd_field(char **p, struct fwdarg *fwd)

```
....
2861.                   memmove(cp, cp + 1, strlen(cp + 1) + 1);
```

Buffer Overflow boundcpy WrongSizeParam\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=79
Status	New

The size of the buffer used by rand_pool_grow in pool, at line 624 of freebsd-src-4/rand_lib.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that rand_pool_grow passes to pool, at line 624 of freebsd-src-4/rand_lib.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/rand_lib.c	freebsd-src-4/rand_lib.c
Line	648	648
Object	pool	pool

Code Snippet
File Name freebsd-src-4/rand_lib.c
Method static int rand_pool_grow(RAND_POOL *pool, size_t len)

```
....
648.          memcpy(p, pool->buffer, pool->len);
```

Buffer Overflow boundcpy WrongSizeParam\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=80
Status	New

The size of the buffer used by sctp_set_key in keylen, at line 331 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_set_key passes to keylen, at line 331 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	340	340
Object	keylen	keylen

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_set_key(uint8_t *key, uint32_t keylen)

```
....
340.        memcpy(new_key->key, key, keylen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=81
Status	New

The size of the buffer used by sctp_compute_hashkey in shared, at line 402 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_compute_hashkey passes to shared, at line 402 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	429	429
Object	shared	shared

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_compute_hashkey(sctp_key_t *key1, sctp_key_t *key2, sctp_key_t *shared)

```
....
429.                    memcpy(key_ptr, shared->key, shared->keylen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=82
Status	New

The size of the buffer used by sctp_compute_hashkey in key1, at line 402 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_compute_hashkey passes to key1, at line 402 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	433	433
Object	key1	key1

Code Snippet	
File Name	freebsd-src-4/sctp_auth.c
Method	sctp_compute_hashkey(sctp_key_t *key1, sctp_key_t *key2, sctp_key_t *shared)

```
....
433.                    memcpy(key_ptr, key1->key, key1->keylen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=83
Status	New

The size of the buffer used by sctp_compute_hashkey in key2, at line 402 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_compute_hashkey passes to key2, at line 402 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	437	437
Object	key2	key2

Code Snippet	
File Name	freebsd-src-4/sctp_auth.c
Method	sctp_compute_hashkey(sctp_key_t *key1, sctp_key_t *key2, sctp_key_t *shared)

```
....
437.                    memcpy(key_ptr, key2->key, key2->keylen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=84
Status	New

The size of the buffer used by sctp_compute_hashkey in shared, at line 402 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_compute_hashkey passes to shared, at line 402 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	442	442
Object	shared	shared

Code Snippet	
File Name	freebsd-src-4/sctp_auth.c
Method	sctp_compute_hashkey(sctp_key_t *key1, sctp_key_t *key2, sctp_key_t *shared)

```
....
442.                    memcpy(key_ptr, shared->key, shared->keylen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=85
Status	New

The size of the buffer used by sctp_compute_hashkey in key2, at line 402 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_compute_hashkey passes to key2, at line 402 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	446	446
Object	key2	key2

Code Snippet	
File Name	freebsd-src-4/sctp_auth.c
Method	sctp_compute_hashkey(sctp_key_t *key1, sctp_key_t *key2, sctp_key_t *shared)

```
....
446.                    memcpy(key_ptr, key2->key, key2->keylen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=86
Status	New

The size of the buffer used by sctp_compute_hashkey in key1, at line 402 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_compute_hashkey passes to key1, at line 402 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	450	450

Object	key1	key1

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_compute_hashkey(sctp_key_t *key1, sctp_key_t *key2, sctp_key_t *shared)

```
....
450.                    memcpy(key_ptr, key1->key, key1->keylen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 37:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=87
Status New

The size of the buffer used by sctp_hmac in keylen, at line 915 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_hmac passes to keylen, at line 915 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	949	949
Object	keylen	keylen

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_hmac(uint16_t hmac_algo, uint8_t *key, uint32_t keylen,

```
....
949.        memcpy(ipad, key, keylen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 38:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=88
Status New

The size of the buffer used by sctp_hmac in keylen, at line 915 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_hmac passes to keylen, at line 915 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	950	950
Object	keylen	keylen

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_hmac(uint16_t hmac_algo, uint8_t *key, uint32_t keylen,

```
....
950.        memcpy(opad, key, keylen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=89
Status	New

The size of the buffer used by sctp_hmac_m in keylen, at line 975 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_hmac_m passes to keylen, at line 975 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1009	1009
Object	keylen	keylen

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_hmac_m(uint16_t hmac_algo, uint8_t *key, uint32_t keylen,

```
....
1009.       memcpy(ipad, key, keylen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=90
Status	New

The size of the buffer used by sctp_hmac_m in keylen, at line 975 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_hmac_m passes to keylen, at line 975 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1010	1010
Object	keylen	keylen

Code Snippet
File Name freebsd-src-4/sctp_auth.c

Method	sctp_hmac_m(uint16_t hmac_algo, uint8_t *key, uint32_t keylen,

```
....
1010.       memcpy(opad, key, keylen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=91
Status	New

The size of the buffer used by sctp_compute_hmac in key, at line 1057 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_compute_hmac passes to key, at line 1057 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1084	1084
Object	key	key

Code Snippet	
File Name	freebsd-src-4/sctp_auth.c
Method	sctp_compute_hmac(uint16_t hmac_algo, sctp_key_t *key, uint8_t *text,

```
....
1084.             memcpy(key->key, temp, key->keylen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=92
Status	New

The size of the buffer used by sctp_compute_hmac_m in key, at line 1092 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_compute_hmac_m passes to key, at line 1092 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1118	1118
Object	key	key

Code Snippet	
File Name	freebsd-src-4/sctp_auth.c
Method	sctp_compute_hmac_m(uint16_t hmac_algo, sctp_key_t *key, struct mbuf *m,

```
....
1118.            memcpy(key->key, temp, key->keylen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=93
Status	New

The size of the buffer used by sctp_auth_get_cookie_params in keylen, at line 1362 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_auth_get_cookie_params passes to keylen, at line 1362 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1463	1463
Object	keylen	keylen

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_auth_get_cookie_params(struct sctp_tcb *stcb, struct mbuf *m,

```
....
1463.                    memcpy(new_key->key, p_random, keylen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=94
Status	New

The size of the buffer used by sctp_hmac in blocklen, at line 915 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_hmac passes to blocklen, at line 915 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	947	947
Object	blocklen	blocklen

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_hmac(uint16_t hmac_algo, uint8_t *key, uint32_t keylen,

```
....
947.        memset(ipad, 0, blocklen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=95
Status	New

The size of the buffer used by sctp_hmac in blocklen, at line 915 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_hmac passes to blocklen, at line 915 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	948	948
Object	blocklen	blocklen

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_hmac(uint16_t hmac_algo, uint8_t *key, uint32_t keylen,

```
....
948.        memset(opad, 0, blocklen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=96
Status	New

The size of the buffer used by sctp_hmac_m in blocklen, at line 975 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_hmac_m passes to blocklen, at line 975 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1007	1007
Object	blocklen	blocklen

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_hmac_m(uint16_t hmac_algo, uint8_t *key, uint32_t keylen,

```
....
1007.       memset(ipad, 0, blocklen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=97
Status	New

The size of the buffer used by sctp_hmac_m in blocklen, at line 975 of freebsd-src-4/sctp_auth.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sctp_hmac_m passes to blocklen, at line 975 of freebsd-src-4/sctp_auth.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1008	1008
Object	blocklen	blocklen

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_hmac_m(uint16_t hmac_algo, uint8_t *key, uint32_t keylen,

```
....
1008.       memset(opad, 0, blocklen);
```

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=1070094&projectid=70084&pathid=98
Status	New

Calling free() (line 1307) on a variable that was not dynamically allocated (line 1307) in file freebsd-src-4/dtrace.c may result with a crash.

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1714	1714
Object	v	v

Code Snippet

File Name	freebsd-src-4/dtrace.c
Method	main(int argc, char *argv[])

```
....
1714.                           free(v);
```

MemoryFree on StackVariable\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=99
Status	New

Calling free() (line 246) on a variable that was not dynamically allocated (line 246) in file freebsd-src-4/localzone.c may result with a crash.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	260	260
Object	nm	nm

Code Snippet	
File Name	freebsd-src-4/localzone.c
Method	lz_enter_zone(struct local_zones* zones, const char* name, const char* type,

```
....
260.                free(nm);
```

MemoryFree on StackVariable\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=100
Status	New

Calling free() (line 604) on a variable that was not dynamically allocated (line 604) in file freebsd-src-4/localzone.c may result with a crash.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	625	625
Object	nm	nm

Code Snippet	
File Name	freebsd-src-4/localzone.c
Method	lz_enter_rr_into_zone(struct local_zone* z, const char* rrstr)

```
....
625.                free(nm);
```

MemoryFree on StackVariable\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=101
Status	New

Calling free() (line 604) on a variable that was not dynamically allocated (line 604) in file freebsd-src-4/localzone.c may result with a crash.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	631	631
Object	nm	nm

Code Snippet
File Name freebsd-src-4/localzone.c
Method lz_enter_rr_into_zone(struct local_zone* z, const char* rrstr)

```
....
631.         free(nm);
```

MemoryFree on StackVariable\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=102
Status	New

Calling free() (line 637) on a variable that was not dynamically allocated (line 637) in file freebsd-src-4/localzone.c may result with a crash.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	658	658
Object	rr_name	rr_name

Code Snippet
File Name freebsd-src-4/localzone.c
Method lz_enter_rr_str(struct local_zones* zones, const char* rr)

```
....
658.        free(rr_name);
```

MemoryFree on StackVariable\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=103
Status	New

Calling free() (line 1023) on a variable that was not dynamically allocated (line 1023) in file freebsd-src-4/localzone.c may result with a crash.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1066	1066
Object	rr_name	rr_name

Code Snippet
File Name freebsd-src-4/localzone.c
Method lz_setup_implicit(struct local_zones* zones, struct config_file* cfg)

```
....
1066.                                   free(rr_name);
```

MemoryFree on StackVariable\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=104
Status	New

Calling free() (line 1023) on a variable that was not dynamically allocated (line 1023) in file freebsd-src-4/localzone.c may result with a crash.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1076	1076
Object	rr_name	rr_name

Code Snippet
File Name freebsd-src-4/localzone.c
Method lz_setup_implicit(struct local_zones* zones, struct config_file* cfg)

```
....
1076.                                   free(rr_name);
```

MemoryFree on StackVariable\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=105
Status	New

Calling free() (line 1023) on a variable that was not dynamically allocated (line 1023) in file freebsd-src-4/localzone.c may result with a crash.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1080	1080
Object	rr_name	rr_name

Code Snippet
File Name freebsd-src-4/localzone.c
Method lz_setup_implicit(struct local_zones* zones, struct config_file* cfg)

```
....
1080.                               free(rr_name);
```

MemoryFree on StackVariable\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=106
Status	New

Calling free() (line 1023) on a variable that was not dynamically allocated (line 1023) in file freebsd-src-4/localzone.c may result with a crash.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1094	1094
Object	rr_name	rr_name

Code Snippet
File Name freebsd-src-4/localzone.c
Method lz_setup_implicit(struct local_zones* zones, struct config_file* cfg)

```
....
1094.                                   free(rr_name);
```

MemoryFree on StackVariable\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=107
Status	New

Calling free() (line 1023) on a variable that was not dynamically allocated (line 1023) in file freebsd-src-4/localzone.c may result with a crash.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1102	1102
Object	rr_name	rr_name

Code Snippet
File Name freebsd-src-4/localzone.c
Method lz_setup_implicit(struct local_zones* zones, struct config_file* cfg)

```
....
1102.                               free(rr_name);
```

MemoryFree on StackVariable\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=108
Status	New

Calling free() (line 1023) on a variable that was not dynamically allocated (line 1023) in file freebsd-src-4/localzone.c may result with a crash.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1106	1106
Object	rr_name	rr_name

Code Snippet
File Name freebsd-src-4/localzone.c
Method lz_setup_implicit(struct local_zones* zones, struct config_file* cfg)

```
....
1106.              } else free(rr_name);
```

MemoryFree on StackVariable\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=109
Status	New

Calling free() (line 1023) on a variable that was not dynamically allocated (line 1023) in file freebsd-src-4/localzone.c may result with a crash.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1118	1118
Object	nm	nm

Code Snippet
File Name freebsd-src-4/localzone.c
Method lz_setup_implicit(struct local_zones* zones, struct config_file* cfg)

```
....
1118.              free(nm);
```

MemoryFree on StackVariable\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=110
Status	New

Calling free() (line 2053) on a variable that was not dynamically allocated (line 2053) in file freebsd-src-4/localzone.c may result with a crash.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	2077	2077
Object	rr_name	rr_name

Code Snippet
File Name freebsd-src-4/localzone.c
Method local_zones_add_RR(struct local_zones* zones, const char* rr)

```
....
2077.             free(rr_name);
```

MemoryFree on StackVariable\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=111
Status	New

Calling free() (line 53) on a variable that was not dynamically allocated (line 53) in file freebsd-src-4/port-uw.c may result with a crash.

	Source	Destination
File	freebsd-src-4/port-uw.c	freebsd-src-4/port-uw.c
Line	88	88
Object	pw_password	pw_password

Code Snippet
File Name freebsd-src-4/port-uw.c
Method sys_auth_passwd(struct ssh *ssh, const char *password)

```
....
88.         free(pw_password);
```

MemoryFree on StackVariable\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=112
Status	New

Calling free() (line 3269) on a variable that was not dynamically allocated (line 3269) in file freebsd-src-4/readconf.c may result with a crash.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	3284	3284
Object	all_key	all_key

Code Snippet
File Name freebsd-src-4/readconf.c
Method dump_client_config(Options *o, const char *host)

```
....
3284.       free(all_key);
```

MemoryFree on StackVariable\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=113
Status	New

Calling free() (line 340) on a variable that was not dynamically allocated (line 340) in file freebsd-src-4/readconf.c may result with a crash.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	349	349
Object	all_key	all_key

Code Snippet
File Name freebsd-src-4/readconf.c
Method kex_default_pk_alg(void)

```
....
349.              free(all_key);
```

MemoryFree on StackVariable\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=114
Status	New

Calling free() (line 591) on a variable that was not dynamically allocated (line 591) in file freebsd-src-4/readconf.c may result with a crash.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	719	719
Object	conn_hash_hex	conn_hash_hex

Code Snippet
File Name freebsd-src-4/readconf.c
Method match_cfg_line(Options *options, char **condition, struct passwd *pw,

```
....
719.                    free(conn_hash_hex);
```

MemoryFree on StackVariable\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=115
Status	New

Calling free() (line 2249) on a variable that was not dynamically allocated (line 2249) in file freebsd-src-4/readconf.c may result with a crash.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	2295	2295
Object	line	line

Code Snippet
File Name freebsd-src-4/readconf.c
Method read_config_file_depth(const char *filename, struct passwd *pw,

```
....
2295.        free(line);
```

MemoryFree on StackVariable\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=116
Status	New

Calling free() (line 2471) on a variable that was not dynamically allocated (line 2471) in file freebsd-src-4/readconf.c may result with a crash.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	2717	2717
Object	all_cipher	all_cipher

Code Snippet
File Name freebsd-src-4/readconf.c
Method fill_default_options(Options * options)

```
....
2717.       free(all_cipher);
```

MemoryFree on StackVariable\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=117
Status	New

Calling free() (line 2471) on a variable that was not dynamically allocated (line 2471) in file freebsd-src-4/readconf.c may result with a crash.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	2718	2718
Object	all_mac	all_mac

Code Snippet
File Name freebsd-src-4/readconf.c
Method fill_default_options(Options * options)

```
....
2718.       free(all_mac);
```

MemoryFree on StackVariable\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=118
Status	New

Calling free() (line 2471) on a variable that was not dynamically allocated (line 2471) in file freebsd-src-4/readconf.c may result with a crash.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	2719	2719
Object	all_kex	all_kex

Code Snippet
File Name freebsd-src-4/readconf.c
Method fill_default_options(Options * options)

```
....
2719.       free(all_kex);
```

MemoryFree on StackVariable\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=119
Status	New

Calling free() (line 2471) on a variable that was not dynamically allocated (line 2471) in file freebsd-src-4/readconf.c may result with a crash.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	2720	2720
Object	all_key	all_key

Code Snippet
File Name freebsd-src-4/readconf.c
Method fill_default_options(Options * options)

```
....
2720.       free(all_key);
```

MemoryFree on StackVariable\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=120
Status	New

Calling free() (line 2471) on a variable that was not dynamically allocated (line 2471) in file freebsd-src-4/readconf.c may result with a crash.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	2721	2721
Object	all_sig	all_sig

Code Snippet
File Name freebsd-src-4/readconf.c
Method fill_default_options(Options * options)

```
....
2721.       free(all_sig);
```

MemoryFree on StackVariable\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=121
Status	New

Calling free() (line 2471) on a variable that was not dynamically allocated (line 2471) in file freebsd-src-4/readconf.c may result with a crash.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	2722	2722
Object	def_cipher	def_cipher

Code Snippet
File Name freebsd-src-4/readconf.c
Method fill_default_options(Options * options)

```
....
2722.       free(def_cipher);
```

MemoryFree on StackVariable\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=122
Status	New

Calling free() (line 2471) on a variable that was not dynamically allocated (line 2471) in file freebsd-src-4/readconf.c may result with a crash.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	2723	2723
Object	def_mac	def_mac

Code Snippet
File Name freebsd-src-4/readconf.c
Method fill_default_options(Options * options)

```
....
2723.       free(def_mac);
```

MemoryFree on StackVariable\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=123
Status	New

Calling free() (line 2471) on a variable that was not dynamically allocated (line 2471) in file freebsd-src-4/readconf.c may result with a crash.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	2724	2724
Object	def_kex	def_kex

Code Snippet
File Name freebsd-src-4/readconf.c
Method fill_default_options(Options * options)

```
....
2724.       free(def_kex);
```

MemoryFree on StackVariable\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=124
Status	New

Calling free() (line 2471) on a variable that was not dynamically allocated (line 2471) in file freebsd-src-4/readconf.c may result with a crash.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	2725	2725
Object	def_key	def_key

Code Snippet
File Name freebsd-src-4/readconf.c
Method fill_default_options(Options * options)

```
....
2725.       free(def_key);
```

MemoryFree on StackVariable\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=125
Status	New

Calling free() (line 2471) on a variable that was not dynamically allocated (line 2471) in file freebsd-src-4/readconf.c may result with a crash.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	2726	2726
Object	def_sig	def_sig

Code Snippet
File Name freebsd-src-4/readconf.c
Method fill_default_options(Options * options)

```
....
2726.       free(def_sig);
```

MemoryFree on StackVariable\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=126
Status	New

Calling free() (line 2891) on a variable that was not dynamically allocated (line 2891) in file freebsd-src-4/readconf.c may result with a crash.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	2982	2982
Object	p	p

Code Snippet
File Name freebsd-src-4/readconf.c
Method parse_forward(struct Forward *fwd, const char *fwdspec, int dynamicfwd, int remotefwd)

```
....
2982.       free(p);
```

MemoryFree on StackVariable\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=127
Status	New

Calling free() (line 3032) on a variable that was not dynamically allocated (line 3032) in file freebsd-src-4/readconf.c may result with a crash.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	3091	3091
Object	user	user

Code Snippet
File Name freebsd-src-4/readconf.c
Method parse_jump(const char *s, Options *o, int active)

```
....
3091.       free(user);
```

MemoryFree on StackVariable\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=128
Status	New

Calling free() (line 3032) on a variable that was not dynamically allocated (line 3032) in file freebsd-src-4/readconf.c may result with a crash.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	3092	3092
Object	host	host

Code Snippet
File Name freebsd-src-4/readconf.c
Method parse_jump(const char *s, Options *o, int active)

```
....
3092.       free(host);
```

MemoryFree on StackVariable\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=129
Status	New

Calling free() (line 3097) on a variable that was not dynamically allocated (line 3097) in file freebsd-src-4/readconf.c may result with a crash.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	3117	3117
Object	user	user

Code Snippet
File Name freebsd-src-4/readconf.c
Method parse_ssh_uri(const char *uri, char **userp, char **hostp, int *portp)

```
....
3117.       free(user);
```

MemoryFree on StackVariable\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=130
Status	New

Calling free() (line 3097) on a variable that was not dynamically allocated (line 3097) in file freebsd-src-4/readconf.c may result with a crash.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	3118	3118
Object	host	host

Code Snippet
File Name freebsd-src-4/readconf.c
Method parse_ssh_uri(const char *uri, char **userp, char **hostp, int *portp)

```
....
3118.        free(host);
```

MemoryFree on StackVariable\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=131
Status	New

Calling free() (line 3097) on a variable that was not dynamically allocated (line 3097) in file freebsd-src-4/readconf.c may result with a crash.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	3119	3119
Object	path	path

Code Snippet
File Name freebsd-src-4/readconf.c
Method parse_ssh_uri(const char *uri, char **userp, char **hostp, int *portp)

```
....
3119.        free(path);
```

MemoryFree on StackVariable\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=132
Status	New

Calling free() (line 162) on a variable that was not dynamically allocated (line 162) in file freebsd-src-4/test_pac.c may result with a crash.

	Source	Destination
File	freebsd-src-4/test_pac.c	freebsd-src-4/test_pac.c
Line	244	244
Object	list	list

Code Snippet
File Name freebsd-src-4/test_pac.c
Method main(int argc, char **argv)

```
....
244.        free(list);
```

MemoryFree on StackVariable\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=133
Status	New

Calling free() (line 162) on a variable that was not dynamically allocated (line 162) in file freebsd-src-4/test_pac.c may result with a crash.

	Source	Destination
File	freebsd-src-4/test_pac.c	freebsd-src-4/test_pac.c
Line	372	372
Object	list	list

Code Snippet
File Name freebsd-src-4/test_pac.c
Method main(int argc, char **argv)

```
....
372.        free(list);
```

MemoryFree on StackVariable\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=134
Status	New

Calling free() (line 207) on a variable that was not dynamically allocated (line 207) in file freebsd-src-4/trans_udp.c may result with a crash.

	Source	Destination
File	freebsd-src-4/trans_udp.c	freebsd-src-4/trans_udp.c
Line	213	213
Object	port	port

Code Snippet
File Name freebsd-src-4/trans_udp.c
Method udp_close_port(struct tport *tp)

```
....
213.        free(port);
```

Use of Uninitialized Pointer

Query Path:
CPP\Cx\CPP Medium Threat\Use of Uninitialized Pointer Version:0

Categories

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

Description

Use of Uninitialized Pointer\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=308
Status	New

The variable declared in d at freebsd-src-4/localzone.c in line 61 is not initialized when it is used by name at freebsd-src-4/localzone.c in line 61.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	63	67
Object	d	name

Code Snippet
File Name freebsd-src-4/localzone.c
Method local_zone_out(struct local_zone* z)

```
....
63.   struct local_data* d;
....
67.               log_nametypeclass(NO_VERBOSE, "rrset", d->name,
```

Use of Uninitialized Pointer\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=309
Status	New

The variable declared in d at freebsd-src-4/localzone.c in line 61 is not initialized when it is used by rrsets at freebsd-src-4/localzone.c in line 61.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c

Line	63	66
Object	d	rrsets

Code Snippet
File Name freebsd-src-4/localzone.c
Method local_zone_out(struct local_zone* z)

```
....
63.   struct local_data* d;
....
66.         for(p = d->rrsets; p; p = p->next) {
```

Use of Uninitialized Pointer\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=310
Status	New

The variable declared in new_key at freebsd-src-4/sctp_auth.c in line 244 is not initialized when it is used by keylen at freebsd-src-4/sctp_auth.c in line 244.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	246	254
Object	new_key	keylen

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_alloc_key(uint32_t keylen)

```
....
246.        sctp_key_t *new_key;
....
254.        new_key->keylen = keylen;
```

Use of Uninitialized Pointer\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=311
Status	New

The variable declared in new_key at freebsd-src-4/sctp_auth.c in line 244 is not initialized when it is used by new_key at freebsd-src-4/sctp_auth.c in line 244.

	Source	Destination

File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	246	250
Object	new_key	new_key

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_alloc_key(uint32_t keylen)

```
....
246.        sctp_key_t *new_key;
....
250.        if (new_key == NULL) {
```

Use of Uninitialized Pointer\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=312
Status	New

The variable declared in new_key at freebsd-src-4/sctp_auth.c in line 244 is not initialized when it is used by new_key at freebsd-src-4/sctp_auth.c in line 244.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	246	255
Object	new_key	new_key

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_alloc_key(uint32_t keylen)

```
....
246.        sctp_key_t *new_key;
....
255.        return (new_key);
```

Use of Uninitialized Pointer\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=313
Status	New

The variable declared in new_key at freebsd-src-4/sctp_auth.c in line 457 is not initialized when it is used by keyid at freebsd-src-4/sctp_auth.c in line 457.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	459	467
Object	new_key	keyid

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_alloc_sharedkey(void)

```
....
459.         sctp_sharedkey_t *new_key;
....
467.         new_key->keyid = 0;
```

Use of Uninitialized Pointer\Path 7:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=314
Status New

The variable declared in new_key at freebsd-src-4/sctp_auth.c in line 457 is not initialized when it is used by new_key at freebsd-src-4/sctp_auth.c in line 457.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	459	463
Object	new_key	new_key

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_alloc_sharedkey(void)

```
....
459.         sctp_sharedkey_t *new_key;
....
463.         if (new_key == NULL) {
```

Use of Uninitialized Pointer\Path 8:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=315
Status New

The variable declared in new_key at freebsd-src-4/sctp_auth.c in line 457 is not initialized when it is used by key at freebsd-src-4/sctp_auth.c in line 457.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	459	468
Object	new_key	key

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_alloc_sharedkey(void)

```
....
459.         sctp_sharedkey_t *new_key;
....
468.         new_key->key = NULL;
```

Use of Uninitialized Pointer\Path 9:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=316
Status New

The variable declared in new_key at freebsd-src-4/sctp_auth.c in line 457 is not initialized when it is used by refcount at freebsd-src-4/sctp_auth.c in line 457.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	459	469
Object	new_key	refcount

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_alloc_sharedkey(void)

```
....
459.         sctp_sharedkey_t *new_key;
....
469.         new_key->refcount = 1;
```

Use of Uninitialized Pointer\Path 10:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=317
Status New

The variable declared in new_key at freebsd-src-4/sctp_auth.c in line 457 is not initialized when it is used by deactivated at freebsd-src-4/sctp_auth.c in line 457.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	459	470
Object	new_key	deactivated

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_alloc_sharedkey(void)

```
....
459.        sctp_sharedkey_t *new_key;
....
470.        new_key->deactivated = 0;
```

Use of Uninitialized Pointer\Path 11:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=318
Status New

The variable declared in new_key at freebsd-src-4/sctp_auth.c in line 457 is not initialized when it is used by new_key at freebsd-src-4/sctp_auth.c in line 457.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	459	471
Object	new_key	new_key

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_alloc_sharedkey(void)

```
....
459.        sctp_sharedkey_t *new_key;
....
471.        return (new_key);
```

Use of Uninitialized Pointer\Path 12:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=319
Status New

The variable declared in skey at freebsd-src-4/sctp_auth.c in line 488 is not initialized when it is used by skey at freebsd-src-4/sctp_auth.c in line 488.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	490	494
Object	skey	skey

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_find_sharedkey(struct sctp_keyhead *shared_keys, uint16_t key_id)

```
....
490.        sctp_sharedkey_t *skey;
....
494.                    return (skey);
```

Use of Uninitialized Pointer\Path 13:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=320
Status New

The variable declared in skey at freebsd-src-4/sctp_auth.c in line 488 is not initialized when it is used by keyid at freebsd-src-4/sctp_auth.c in line 488.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	490	493
Object	skey	keyid

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_find_sharedkey(struct sctp_keyhead *shared_keys, uint16_t key_id)

```
....
490.        sctp_sharedkey_t *skey;
....
493.              if (skey->keyid == key_id)
```

Use of Uninitialized Pointer\Path 14:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=321
Status New

The variable declared in new_list at freebsd-src-4/sctp_auth.c in line 630 is not initialized when it is used by max_algo at freebsd-src-4/sctp_auth.c in line 630.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	632	642
Object	new_list	max_algo

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_alloc_hmaclist(uint16_t num_hmacs)

```
....
632.        sctp_hmaclist_t *new_list;
....
642.        new_list->max_algo = num_hmacs;
```

Use of Uninitialized Pointer\Path 15:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=322
Status New

The variable declared in new_list at freebsd-src-4/sctp_auth.c in line 630 is not initialized when it is used by new_list at freebsd-src-4/sctp_auth.c in line 630.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	632	638
Object	new_list	new_list

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_alloc_hmaclist(uint16_t num_hmacs)

```
....
632.        sctp_hmaclist_t *new_list;
....
638.        if (new_list == NULL) {
```

Use of Uninitialized Pointer\Path 16:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=323
Status New

The variable declared in new_list at freebsd-src-4/sctp_auth.c in line 630 is not initialized when it is used by num_algo at freebsd-src-4/sctp_auth.c in line 630.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	632	643
Object	new_list	num_algo

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_alloc_hmaclist(uint16_t num_hmacs)

```
....
632.        sctp_hmaclist_t *new_list;
....
643.        new_list->num_algo = 0;
```

Use of Uninitialized Pointer\Path 17:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=324
Status New

The variable declared in new_list at freebsd-src-4/sctp_auth.c in line 630 is not initialized when it is used by new_list at freebsd-src-4/sctp_auth.c in line 630.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	632	644
Object	new_list	new_list

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_alloc_hmaclist(uint16_t num_hmacs)

```
....
632.        sctp_hmaclist_t *new_list;
....
644.        return (new_list);
```

Use of Uninitialized Pointer\Path 18:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=325
Status New

The variable declared in cause at freebsd-src-4/sctp_auth.c in line 1590 is not initialized when it is used by length at freebsd-src-4/sctp_auth.c in line 1590.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1619	1637
Object	cause	length

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_handle_auth(struct sctp_tcb *stcb, struct sctp_auth_chunk *auth,

```
....
1619.             struct sctp_error_auth_invalid_hmac *cause;
....
1637.                    cause->cause.length = htons(sizeof(struct
sctp_error_auth_invalid_hmac));
```

Use of Uninitialized Pointer\Path 19:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=326
Status New

The variable declared in cause at freebsd-src-4/sctp_auth.c in line 1590 is not initialized when it is used by code at freebsd-src-4/sctp_auth.c in line 1590.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1619	1636
Object	cause	code

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_handle_auth(struct sctp_tcb *stcb, struct sctp_auth_chunk *auth,

```
....
1619.             struct sctp_error_auth_invalid_hmac *cause;
....
1636.                    cause->cause.code =
htons(SCTP_CAUSE_UNSUPPORTED_HMACID);
```

Use of Uninitialized Pointer\Path 20:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=327
Status New

The variable declared in cause at freebsd-src-4/sctp_auth.c in line 1590 is not initialized when it is used by cause at freebsd-src-4/sctp_auth.c in line 1590.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1619	1638
Object	cause	cause

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_handle_auth(struct sctp_tcb *stcb, struct sctp_auth_chunk *auth,

```
....
1619.             struct sctp_error_auth_invalid_hmac *cause;
....
1638.                   cause->hmac_id = ntohs(hmac_id);
```

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=1070094&projectid=70084&pathid=330
Status	New

The variable declared in res at freebsd-src-4/localzone.c in line 1226 is not initialized when it is used by res at freebsd-src-4/localzone.c in line 1226.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1230	1245
Object	res	res

Code Snippet
File Name freebsd-src-4/localzone.c
Method local_zones_tags_lookup(struct local_zones* zones,

```
....
1230.       rbnode_type* res = NULL;
....
1245.       result = (struct local_zone*)res;
```

Use of Zero Initialized Pointer\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=331
Status	New

The variable declared in tmp_meth at freebsd-src-4/rand_lib.c in line 848 is not initialized when it is used by tmp_meth at freebsd-src-4/rand_lib.c in line 848.

	Source	Destination
File	freebsd-src-4/rand_lib.c	freebsd-src-4/rand_lib.c
Line	850	873
Object	tmp_meth	tmp_meth

Code Snippet
File Name freebsd-src-4/rand_lib.c
Method const RAND_METHOD *RAND_get_rand_method(void)

```
....
850.        const RAND_METHOD *tmp_meth = NULL;
....
873.        tmp_meth = default_RAND_meth;
```

Use of Zero Initialized Pointer\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=332
Status	New

The variable declared in p_random at freebsd-src-4/sctp_auth.c in line 1362 is not initialized when it is used by p_random at freebsd-src-4/sctp_auth.c in line 1362.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1368	1454
Object	p_random	p_random

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_auth_get_cookie_params(struct sctp_tcb *stcb, struct mbuf *m,

```
....
1368.        struct sctp_auth_random *p_random = NULL;
....
1454.        keylen = sizeof(*p_random) + random_len + sizeof(*hmacs) +
hmacs_len;
```

Use of Zero Initialized Pointer\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=333
Status	New

The variable declared in hmacs at freebsd-src-4/sctp_auth.c in line 1362 is not initialized when it is used by hmacs at freebsd-src-4/sctp_auth.c in line 1362.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1371	1454
Object	hmacs	hmacs

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_auth_get_cookie_params(struct sctp_tcb *stcb, struct mbuf *m,

```
....
1371.       struct sctp_auth_hmac_algo *hmacs = NULL;
....
1454.       keylen = sizeof(*p_random) + random_len + sizeof(*hmacs) +
hmacs_len;
```

Use of Zero Initialized Pointer\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=334
Status	New

The variable declared in res at freebsd-src-4/localzone.c in line 1226 is not initialized when it is used by rrsets at freebsd-src-4/localzone.c in line 359.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1230	370
Object	res	rrsets

Code Snippet
File Name freebsd-src-4/localzone.c
Method local_zones_tags_lookup(struct local_zones* zones,

```
....
1230.       rbnode_type* res = NULL;
```

File Name freebsd-src-4/localzone.c

Method new_local_rrset(struct regional* region, struct local_data* node,

```
....
370.        node->rrsets = rrset;
```

Use of Zero Initialized Pointer\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=335
Status	New

The variable declared in res at freebsd-src-4/localzone.c in line 1226 is not initialized when it is used by soa at freebsd-src-4/localzone.c in line 510.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1230	523
Object	res	soa

Code Snippet

File Name freebsd-src-4/localzone.c

Method local_zones_tags_lookup(struct local_zones* zones,

```
....
1230.       rbnode_type* res = NULL;
```

File Name freebsd-src-4/localzone.c

Method lz_mark_soa_for_zone(struct local_zone* z, struct ub_packed_rrset_key* soa_rrset,

```
....
523.        z->soa = soa_rrset;
```

Use of Zero Initialized Pointer\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=336
Status	New

The variable declared in res at freebsd-src-4/localzone.c in line 1226 is not initialized when it is used by soa_negative at freebsd-src-4/localzone.c in line 510.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1230	542
Object	res	soa_negative

Code Snippet
File Name freebsd-src-4/localzone.c
Method local_zones_tags_lookup(struct local_zones* zones,

```
....
1230.        rbnode_type* res = NULL;
```

File Name freebsd-src-4/localzone.c

Method lz_mark_soa_for_zone(struct local_zone* z, struct ub_packed_rrset_key* soa_rrset,

```
....
542.         z->soa_negative = rrset_negative;
```

Use of Zero Initialized Pointer\Path 8:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=337
Status New

The variable declared in parent at freebsd-src-4/localzone.c in line 986 is not initialized when it is used by parent at freebsd-src-4/localzone.c in line 986.

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	993	1009
Object	parent	parent

Code Snippet
File Name freebsd-src-4/localzone.c
Method init_parents(struct local_zones* zones)

```
....
993.                    node->parent = NULL;
....
1009.                                    node->parent = p;
```

Use of Zero Initialized Pointer\Path 9:

Severity Medium
Result State To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=338
Status	New

The variable declared in user at freebsd-src-4/readconf.c in line 3097 is not initialized when it is used by jump_user at freebsd-src-4/readconf.c in line 3032.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	3099	3077
Object	user	jump_user

Code Snippet

File Name freebsd-src-4/readconf.c

Method parse_ssh_uri(const char *uri, char **userp, char **hostp, int *portp)

```
....
3099.        char *user = NULL, *host = NULL, *path = NULL;
```

File Name freebsd-src-4/readconf.c

Method parse_jump(const char *s, Options *o, int active)

```
....
3077.                    o->jump_user = user;
```

Use of Zero Initialized Pointer\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=339
Status	New

The variable declared in host at freebsd-src-4/readconf.c in line 3032 is not initialized when it is used by jump_host at freebsd-src-4/readconf.c in line 3032.

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	3035	3078
Object	host	jump_host

Code Snippet

File Name freebsd-src-4/readconf.c

Method parse_jump(const char *s, Options *o, int active)

```
....
3035.        char *host = NULL, *user = NULL;
....
3078.                    o->jump_host = host;
```

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=1070094&projectid=70084&pathid=300
Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	310	310
Object	argv	argv

Code Snippet
File Name freebsd-src-4/dtrace.c
Method make_argv(char *s)

```
....
310.        char **argv = malloc(sizeof (char *) * (strlen(s) / 2 + 1));
```

Memory Leak\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=301
Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1334	1334
Object	g_argv	g_argv

Code Snippet
File Name freebsd-src-4/dtrace.c

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

```
....
1334.       if ((g_argv = malloc(sizeof (char *) * argc)) == NULL ||
```

Memory Leak\Path 3:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=302
Status New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1335	1335
Object	g_cmdv	g_cmdv

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1335.           (g_cmdv = malloc(sizeof (dtrace_cmd_t) * argc)) == NULL
||
```

Memory Leak\Path 4:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=303
Status New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1336	1336
Object	g_psv	g_psv

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1336.           (g_psv = malloc(sizeof (struct ps_prochandle *) * argc))
== NULL)
```

Memory Leak\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=304
Status	New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	246	246
Object	core	core

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method main(int argc, char **argv)

```
....
246.                    core = strdup(optarg);
```

Memory Leak\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=305
Status	New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	250	250
Object	kernel	kernel

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method main(int argc, char **argv)

```
....
250.                    kernel = strdup(optarg);
```

Memory Leak\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=306
Status	New

	Source	Destination

File	freebsd-src-4/trans_udp.c	freebsd-src-4/trans_udp.c
Line	167	167
Object	port	port

Code Snippet
File Name freebsd-src-4/trans_udp.c
Method udp_open_port(u_int8_t *addr, u_int32_t udp_port, struct udp_port **pp)

```
....
167.        if ((port = malloc(sizeof(*port))) == NULL)
```

Memory Leak\Path 8:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=307
Status New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	347	347
Object	sz	sz

Code Snippet
File Name freebsd-src-4/dtrace.c
Method dof_prune(const char *fname)

```
....
347.        if ((buf = malloc((sz = sbuf.st_size) + 1)) == NULL)
```

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=1070094&projectid=70084&pathid=291
Status New

	Source	Destination

File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1080	1094
Object	rr_name	rr_name

Code Snippet
File Name freebsd-src-4/localzone.c
Method lz_setup_implicit(struct local_zones* zones, struct config_file* cfg)

```
....
1080.                           free(rr_name);
....
1094.                                 free(rr_name);
```

Double Free\Path 2:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=292
Status New

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1102	1102
Object	rr_name	rr_name

Code Snippet
File Name freebsd-src-4/localzone.c
Method lz_setup_implicit(struct local_zones* zones, struct config_file* cfg)

```
....
1102.                           free(rr_name);
```

Double Free\Path 3:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=293
Status New

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	724	724
Object	cmd	cmd

Code Snippet

File Name	freebsd-src-4/readconf.c
Method	match_cfg_line(Options *options, char **condition, struct passwd *pw,

```
....
724.                            free(cmd);
```

Double Free\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=294
Status	New

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	734	734
Object	cmd	cmd

Code Snippet

File Name	freebsd-src-4/readconf.c
Method	match_cfg_line(Options *options, char **condition, struct passwd *pw,

```
....
734.                        free(cmd);
```

Double Free\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=295
Status	New

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	747	747
Object	criteria	criteria

Code Snippet

File Name	freebsd-src-4/readconf.c
Method	match_cfg_line(Options *options, char **condition, struct passwd *pw,

```
....
747.                free(criteria);
```

Double Free\Path 6:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=296
Status	New

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	1896	1896
Object	arg2	arg2

Code Snippet
File Name freebsd-src-4/readconf.c
Method process_config_line_depth(Options *options, struct passwd *pw, const char *host,

```
....
1896.                           free(arg2);
```

Double Free\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=297
Status	New

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	1903	1903
Object	arg2	arg2

Code Snippet
File Name freebsd-src-4/readconf.c
Method process_config_line_depth(Options *options, struct passwd *pw, const char *host,

```
....
1903.                       free(arg2);
```

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=1070094&projectid=700

	84&pathid=144
Status	New

The function needed in freebsd-src-4/ifmcstat.c at line 794 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	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	825	825
Object	needed	needed

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method inm_print_sources_sysctl(uint32_t ifindex, struct in_addr gina)

```
....
825.                if ((buf = malloc(needed)) == NULL) {
```

Wrong Size t Allocation\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=145
Status	New

The function needed in freebsd-src-4/ifmcstat.c at line 892 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	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	929	929
Object	needed	needed

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method in6m_print_sources_sysctl(uint32_t ifindex, struct in6_addr *pgroup)

```
....
929.                if ((buf = malloc(needed)) == NULL) {
```

Wrong Size t Allocation\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=146
Status	New

The function len in freebsd-src-4/readconf.c at line 948 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	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	1348	1348
Object	len	len

Code Snippet
File Name freebsd-src-4/readconf.c
Method process_config_line_depth(Options *options, struct passwd *pw, const char *host,

```
....
1348.                    *charptr = xstrdup(str + len);
```

Wrong Size t Allocation\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=147
Status	New

The function sz in freebsd-src-4/dtrace.c at line 328 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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	347	347
Object	sz	sz

Code Snippet
File Name freebsd-src-4/dtrace.c
Method dof_prune(const char *fname)

```
....
347.        if ((buf = malloc((sz = sbuf.st_size) + 1)) == NULL)
```

Inadequate Encryption Strength

Query Path:
CPP\Cx\CPP Medium Threat\Inadequate Encryption Strength Version:1

Categories

FISMA 2014: Configuration Management
NIST SP 800-53: SC-13 Cryptographic Protection (P1)
OWASP Top 10 2017: A3-Sensitive Data Exposure

Description

Inadequate Encryption Strength\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=340
Status	New

The application uses a weak cryptographic algorithm, SCTP_SHA1_UPDATE at line 869 of freebsd-src-4/sctp_auth.c, to protect sensitive personal information authinfo, from freebsd-src-4/sctp_auth.c at line 1502.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1530	874
Object	authinfo	SCTP_SHA1_UPDATE

Code Snippet

File Name freebsd-src-4/sctp_auth.c

Method sctp_fill_hmac_digest_m(struct mbuf *m, uint32_t auth_offset,

```
....
1530.             stcb->asoc.authinfo.assoc_key =
```

File Name freebsd-src-4/sctp_auth.c

Method sctp_hmac_update(uint16_t hmac_algo, sctp_hash_context_t *ctx,

```
....
874.              SCTP_SHA1_UPDATE(&ctx->sha1, text, textlen);
```

Inadequate Encryption Strength\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=341
Status	New

The application uses a weak cryptographic algorithm, SCTP_SHA1_UPDATE at line 869 of freebsd-src-4/sctp_auth.c, to protect sensitive personal information authinfo, from freebsd-src-4/sctp_auth.c at line 1590.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1672	874
Object	authinfo	SCTP_SHA1_UPDATE

Code Snippet

File Name freebsd-src-4/sctp_auth.c

Method sctp_handle_auth(struct sctp_tcb *stcb, struct sctp_auth_chunk *auth,

```
....
1672.            stcb->asoc.authinfo.recv_key =
```

File Name freebsd-src-4/sctp_auth.c

Method sctp_hmac_update(uint16_t hmac_algo, sctp_hash_context_t *ctx,

```
....
874.             SCTP_SHA1_UPDATE(&ctx->sha1, text, textlen);
```

Inadequate Encryption Strength\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=342
Status	New

The application uses a weak cryptographic algorithm, SCTP_SHA1_FINAL at line 887 of freebsd-src-4/sctp_auth.c, to protect sensitive personal information auth, from freebsd-src-4/sctp_auth.c at line 1502.

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	1548	892
Object	auth	SCTP_SHA1_FINAL

Code Snippet

File Name freebsd-src-4/sctp_auth.c

Method sctp_fill_hmac_digest_m(struct mbuf *m, uint32_t auth_offset,

```
....
1548.           m, auth_offset, auth->hmac);
```

File Name freebsd-src-4/sctp_auth.c

Method sctp_hmac_final(uint16_t hmac_algo, sctp_hash_context_t *ctx,

```
....
892.             SCTP_SHA1_FINAL(digest, &ctx->sha1);
```

Heap Inspection

Query Path:

CPP\Cx\CPP Medium Threat\Heap Inspection Version:1

Categories

OWASP Top 10 2013: A6-Sensitive Data Exposure
FISMA 2014: Media Protection
NIST SP 800-53: SC-4 Information in Shared Resources (P1)
OWASP Top 10 2017: A3-Sensitive Data Exposure

Description

Heap Inspection\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=298
Status	New

Method sys_auth_passwd at line 53 of freebsd-src-4/port-uw.c defines pw_password, which is designated to contain user passwords. However, while plaintext passwords are later assigned to pw_password, this variable is never cleared from memory.

	Source	Destination
File	freebsd-src-4/port-uw.c	freebsd-src-4/port-uw.c
Line	61	61
Object	pw_password	pw_password

Code Snippet

File Name freebsd-src-4/port-uw.c

Method sys_auth_passwd(struct ssh *ssh, const char *password)

```
....
61.   char *pw_password = authctxt->valid ? shadow_pw(pw) : pw-
>pw_passwd;
```

Heap Inspection\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=299
Status	New

Method SRP_VBASE_init at line 382 of freebsd-src-4/srp_vfy.c defines user_pwd, which is designated to contain user passwords. However, while plaintext passwords are later assigned to user_pwd, this variable is never cleared from memory.

	Source	Destination
File	freebsd-src-4/srp_vfy.c	freebsd-src-4/srp_vfy.c
Line	391	391
Object	user_pwd	user_pwd

Code Snippet

File Name freebsd-src-4/srp_vfy.c

Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
391.      SRP_user_pwd *user_pwd = NULL;
```

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=1070094&projectid=70084&pathid=328
Status	New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	196	274
Object	error	error

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method main(int argc, char **argv)

```
....
196.        int c, error;
....
274.        if (error != 0)
```

Use of Uninitialized Variable\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=329
Status	New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	196	271
Object	error	error

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method main(int argc, char **argv)

```
....
196.        int c, error;
....
271.        if (!Kflag || (error != 0 && (core == NULL && kernel ==
NULL)))
```

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=1070094&projectid=70084&pathid=186
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 41 of freebsd-src-4/srp_vfy.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	freebsd-src-4/srp_vfy.c	freebsd-src-4/srp_vfy.c
Line	117	117
Object	AssignExpr	AssignExpr

Code Snippet
File Name freebsd-src-4/srp_vfy.c
Method static int t_fromb64(unsigned char *a, size_t alen, const char *src)

```
....
117.            outl -= padsize;
```

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=1070094&projectid=70084&pathid=343
Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	350	350
Object	buf	buf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method dof_prune(const char *fname)

```
....
350.         if (read(fd, buf, sz) != sz)
```

Improper Resource Access Authorization\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=344
Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1356	1356
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1356.                                   (void) fprintf(stderr,
```

Improper Resource Access Authorization\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=345
Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c

Line	1367	1367
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1367.                                   (void) fprintf(stderr,
```

Improper Resource Access Authorization\Path 4:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=346
Status New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1429	1429
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1429.               (void) fprintf(stderr, "%s: only one of the [-AGhlV]
options "
```

Improper Resource Access Authorization\Path 5:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=347
Status New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1686	1686
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1686.              (void) fprintf(stderr, "%s: -B not valid in
combination"
```

Improper Resource Access Authorization\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=348
Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1692	1692
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1692.              (void) fprintf(stderr, "%s: -B not valid in
combination"
```

Improper Resource Access Authorization\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=349
Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1848	1848
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1848.                    (void) fprintf(stderr, "%s: -G requires one or
more "
```

Improper Resource Access Authorization\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=350
Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1891	1891
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1891.                    (void) fprintf(stderr, "%s: -h requires one or
more "
```

Improper Resource Access Authorization\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=351
Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1901	1901
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1901.                          (void) fprintf(stderr, "%s: -h requires an
"
```

Improper Resource Access Authorization\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=352
Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1910	1910
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1910.                          (void) fprintf(stderr, "%s: -h requires an
"
```

Improper Resource Access Authorization\Path 11:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=353
Status New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	133	133
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method usage(FILE *fp)

```
....
133.        (void) fprintf(fp, "Usage: %s [-32|-64] [-aACdeFGhHlqSvVwZ]
"
```

Improper Resource Access Authorization\Path 12:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=354
Status New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	144	144
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method usage(FILE *fp)

```
....
144.        (void) fprintf(fp, "\tpredicate -> '/' D-expression '/'\n");
```

Improper Resource Access Authorization\Path 13:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=355
Status New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	145	145
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method usage(FILE *fp)

```
....
145.        (void) fprintf(fp, "\t   action -> '{' D-statements '}'\n");
```

Improper Resource Access Authorization\Path 14:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=356
Status New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	147	147
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method usage(FILE *fp)

```
....
147.        (void) fprintf(fp, "\n"
```

Improper Resource Access Authorization\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=357
Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	191	191
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method verror(const char *fmt, va_list ap)

```
....
191.         (void) fprintf(stderr, "%s: ", g_pname);
```

Improper Resource Access Authorization\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=358
Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	195	195
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method verror(const char *fmt, va_list ap)

```
....
195.                 (void) fprintf(stderr, ": %s\n", strerror(error));
```

Improper Resource Access Authorization\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=359
Status	New

	Source	Destination

File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	230	230
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method dfatal(const char *fmt, ...)

```
....
230.         (void) fprintf(stderr, "%s: ", g_pname);
```

Improper Resource Access Authorization\Path 18:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=360
Status New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	237	237
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method dfatal(const char *fmt, ...)

```
....
237.                 (void) fprintf(stderr, ": %s\n",
```

Improper Resource Access Authorization\Path 19:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=361
Status New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	240	240
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method dfatal(const char *fmt, ...)

```
....
240.              (void) fprintf(stderr, "%s\n",
```

Improper Resource Access Authorization\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=362
Status	New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1023	1023
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1023.                   fprintf(stderr,
```

Improper Resource Access Authorization\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=363
Status	New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1047	1047
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1047.                   fprintf(stdout, "%s:\n", thisifname);
```

Improper Resource Access Authorization\Path 22:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=364
Status	New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1133	1133
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1133.                    fprintf(stdout, "\t%s %s", pafname, addrbuf);
```

Improper Resource Access Authorization\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=365
Status	New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1137	1137
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1137.                            fprintf(stdout, " scopeid 0x%x",
```

Improper Resource Access Authorization\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=366
Status	New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1140	1140

Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1140.                    fprintf(stdout, "\n");
```

Improper Resource Access Authorization\Path 25:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=367
Status New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1212	1212
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1212.                fprintf(stdout, "\t\tgroup %s", addrbuf);
```

Improper Resource Access Authorization\Path 26:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=368
Status New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1216	1216
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1216.                   fprintf(stdout, " scopeid 0x%x",
```

Improper Resource Access Authorization\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=369
Status	New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1231	1231
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1231.             fprintf(stdout, "\n");
```

Improper Resource Access Authorization\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=370
Status	New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1238	1238
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1238.                   fprintf(stdout, "\t\t\tmcast-macaddr %s\n",
addrbuf);
```

Improper Resource Access Authorization\Path 29:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=371
Status	New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	177	177
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method usage()

```
....
177.        fprintf(stderr,
```

Improper Resource Access Authorization\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=372
Status	New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	206	206
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method main(int argc, char **argv)

```
....
206.                        fprintf(stderr, "%s: unknown interface\n",
```

Improper Resource Access Authorization\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=373
Status	New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c

Line	229	229
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method main(int argc, char **argv)

```
....
229.                    fprintf(stderr, "%s: unknown address family\n",
optarg);
```

Improper Resource Access Authorization\Path 32:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=374
Status New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	864	864
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method inm_print_sources_sysctl(uint32_t ifindex, struct in_addr gina)

```
....
864.                fprintf(stdout, "%s%s", (i == 0 ? "" : ","),
```

Improper Resource Access Authorization\Path 33:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=375
Status New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	869	869
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method inm_print_sources_sysctl(uint32_t ifindex, struct in_addr gina)

```
....
869.                fprintf(stderr, "warning: %u trailing bytes from
%s\n",
```

Improper Resource Access Authorization\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=376
Status	New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	970	970
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method in6m_print_sources_sysctl(uint32_t ifindex, struct in6_addr *pgroup)

```
....
970.                fprintf(stdout, "%s%s", (i == 0 ? "" : ","), addrbuf);
```

Improper Resource Access Authorization\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=377
Status	New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	974	974
Object	fprintf	fprintf

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method in6m_print_sources_sysctl(uint32_t ifindex, struct in6_addr *pgroup)

```
....
974.                fprintf(stderr, "warning: %u trailing bytes from
%s\n",
```

NULL Pointer Dereference

Query Path:

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=1070094&projectid=70084&pathid=148
Status	New

The variable declared in null at freebsd-src-4/dtrace.c in line 1307 is not initialized when it is used by dc_name at freebsd-src-4/dtrace.c in line 678.

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1823	683
Object	null	dc_name

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1823.             anon_prog(NULL, dtrace_geterr_dof(g_dtp), i++);
```

File Name freebsd-src-4/dtrace.c
Method anon_prog(const dtrace_cmd_t *dcp, dof_hdr_t *dof, int n)

```
....
683.              dfatal("failed to create DOF image for '%s'", dcp-
>dc_name);
```

NULL Pointer Dereference\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=149
Status	New

The variable declared in null at freebsd-src-4/dtrace.c in line 1307 is not initialized when it is used by dc_name at freebsd-src-4/dtrace.c in line 678.

	Source	Destination

File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1824	683
Object	null	dc_name

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1824.             anon_prog(NULL, dtrace_getopt_dof(g_dtp), i++);
```

File Name freebsd-src-4/dtrace.c
Method anon_prog(const dtrace_cmd_t *dcp, dof_hdr_t *dof, int n)

```
....
683.              dfatal("failed to create DOF image for '%s'", dcp-
>dc_name);
```

NULL Pointer Dereference\Path 3:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=150
Status New

The variable declared in null at freebsd-src-4/ifmcstat.c in line 985 is not initialized when it is used by ss at freebsd-src-4/ifmcstat.c in line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1057	1086
Object	null	ss

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1057.             pifasa = NULL;
....
1086.             if (lastifasa.ss.ss_family == AF_UNSPEC ||
```

NULL Pointer Dereference\Path 4:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=700

	84&pathid=151
Status	New

The variable declared in null at freebsd-src-4/ifmcstat.c in line 985 is not initialized when it is used by ss at freebsd-src-4/ifmcstat.c in line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1074	1086
Object	null	ss

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1074.                           pifasa = NULL;
....
1086.               if (lastifasa.ss.ss_family == AF_UNSPEC ||
```

NULL Pointer Dereference\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=152
Status	New

The variable declared in null at freebsd-src-4/ifmcstat.c in line 985 is not initialized when it is used by ss at freebsd-src-4/ifmcstat.c in line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1057	1087
Object	null	ss

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1057.               pifasa = NULL;
....
1087.                    ((lastifasa.ss.ss_family == AF_LINK &&
```

NULL Pointer Dereference\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=700

	84&pathid=153
Status	New

The variable declared in null at freebsd-src-4/ifmcstat.c in line 985 is not initialized when it is used by ss at freebsd-src-4/ifmcstat.c in line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1074	1087
Object	null	ss

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1074.                           pifasa = NULL;
....
1087.                   ((lastifasa.ss.ss_family == AF_LINK &&
```

NULL Pointer Dereference\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=154
Status	New

The variable declared in null at freebsd-src-4/ifmcstat.c in line 985 is not initialized when it is used by sa at freebsd-src-4/ifmcstat.c in line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1057	1170
Object	null	sa

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1057.              pifasa = NULL;
....
1170.                    if (pifasa->sa.sa_family == AF_INET6) {
```

NULL Pointer Dereference\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=700

	84&pathid=155
Status	New

The variable declared in null at freebsd-src-4/ifmcstat.c in line 985 is not initialized when it is used by sa at freebsd-src-4/ifmcstat.c in line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1074	1170
Object	null	sa

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1074.                           pifasa = NULL;
....
1170.                   if (pifasa->sa.sa_family == AF_INET6) {
```

NULL Pointer Dereference\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=156
Status	New

The variable declared in null at freebsd-src-4/ifmcstat.c in line 985 is not initialized when it is used by sa at freebsd-src-4/ifmcstat.c in line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1057	1145
Object	null	sa

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1057.              pifasa = NULL;
....
1145.                    if (pifasa->sa.sa_family == AF_INET) {
```

NULL Pointer Dereference\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=700

	84&pathid=157
Status	New

The variable declared in null at freebsd-src-4/ifmcstat.c in line 985 is not initialized when it is used by sa at freebsd-src-4/ifmcstat.c in line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1074	1145
Object	null	sa

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1074.                           pifasa = NULL;
....
1145.                   if (pifasa->sa.sa_family == AF_INET) {
```

NULL Pointer Dereference\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=158
Status	New

The variable declared in null at freebsd-src-4/ifmcstat.c in line 985 is not initialized when it is used by sin6 at freebsd-src-4/ifmcstat.c in line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1057	1138
Object	null	sin6

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1057.              pifasa = NULL;
....
1138.                               pifasa->sin6.sin6_scope_id);
```

NULL Pointer Dereference\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=700

	84&pathid=159
Status	New

The variable declared in null at freebsd-src-4/ifmcstat.c in line 985 is not initialized when it is used by sin6 at freebsd-src-4/ifmcstat.c in line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1074	1138
Object	null	sin6

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1074.                           pifasa = NULL;
....
1138.                               pifasa->sin6.sin6_scope_id);
```

NULL Pointer Dereference\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=160
Status	New

The variable declared in null at freebsd-src-4/ifmcstat.c in line 985 is not initialized when it is used by sa at freebsd-src-4/ifmcstat.c in line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1057	1135
Object	null	sa

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1057.              pifasa = NULL;
....
1135.                    if (pifasa->sa.sa_family == AF_INET6 &&
```

NULL Pointer Dereference\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=700

	84&pathid=161
Status	New

The variable declared in null at freebsd-src-4/ifmcstat.c in line 985 is not initialized when it is used by sa at freebsd-src-4/ifmcstat.c in line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1074	1135
Object	null	sa

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1074.                           pifasa = NULL;
....
1135.                   if (pifasa->sa.sa_family == AF_INET6 &&
```

NULL Pointer Dereference\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=162
Status	New

The variable declared in null at freebsd-src-4/ifmcstat.c in line 985 is not initialized when it is used by sin6 at freebsd-src-4/ifmcstat.c in line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1057	1136
Object	null	sin6

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1057.              pifasa = NULL;
....
1136.                          pifasa->sin6.sin6_scope_id)
```

NULL Pointer Dereference\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=700

	84&pathid=163
Status	New

The variable declared in null at freebsd-src-4/ifmcstat.c in line 985 is not initialized when it is used by sin6 at freebsd-src-4/ifmcstat.c in line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1074	1136
Object	null	sin6

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1074.                           pifasa = NULL;
....
1136.                         pifasa->sin6.sin6_scope_id)
```

NULL Pointer Dereference\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=164
Status	New

The variable declared in null at freebsd-src-4/ifmcstat.c in line 985 is not initialized when it is used by sa at freebsd-src-4/ifmcstat.c in line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1057	1082
Object	null	sa

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1057.              pifasa = NULL;
....
1082.              if (!vflag && pifasa->sa.sa_family == AF_LINK)
```

NULL Pointer Dereference\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=700

	84&pathid=165
Status	New

The variable declared in null at freebsd-src-4/ifmcstat.c in line 985 is not initialized when it is used by sa at freebsd-src-4/ifmcstat.c in line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1074	1082
Object	null	sa

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1074.                           pifasa = NULL;
....
1082.              if (!vflag && pifasa->sa.sa_family == AF_LINK)
```

NULL Pointer Dereference\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=166
Status	New

The variable declared in null at freebsd-src-4/srp_vfy.c in line 382 is not initialized when it is used by info at freebsd-src-4/srp_vfy.c in line 176.

	Source	Destination
File	freebsd-src-4/srp_vfy.c	freebsd-src-4/srp_vfy.c
Line	391	183
Object	null	info

Code Snippet
File Name freebsd-src-4/srp_vfy.c
Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
391.      SRP_user_pwd *user_pwd = NULL;
```

File Name freebsd-src-4/srp_vfy.c
Method void SRP_user_pwd_free(SRP_user_pwd *user_pwd)

```
....
183.      OPENSSL_free(user_pwd->info);
```

NULL Pointer Dereference\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=167
Status	New

The variable declared in null at freebsd-src-4/srp_vfy.c in line 382 is not initialized when it is used by info at freebsd-src-4/srp_vfy.c in line 176.

	Source	Destination
File	freebsd-src-4/srp_vfy.c	freebsd-src-4/srp_vfy.c
Line	455	183
Object	null	info

Code Snippet

File Name freebsd-src-4/srp_vfy.c

Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
455.                  user_pwd = NULL; /* abandon responsibility */
```

File Name freebsd-src-4/srp_vfy.c

Method void SRP_user_pwd_free(SRP_user_pwd *user_pwd)

```
....
183.      OPENSSL_free(user_pwd->info);
```

NULL Pointer Dereference\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=168
Status	New

The variable declared in null at freebsd-src-4/srp_vfy.c in line 382 is not initialized when it is used by id at freebsd-src-4/srp_vfy.c in line 176.

	Source	Destination
File	freebsd-src-4/srp_vfy.c	freebsd-src-4/srp_vfy.c
Line	391	182

Object	null	id

Code Snippet
File Name freebsd-src-4/srp_vfy.c
Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
391.      SRP_user_pwd *user_pwd = NULL;
```

File Name freebsd-src-4/srp_vfy.c

Method void SRP_user_pwd_free(SRP_user_pwd *user_pwd)

```
....
182.      OPENSSL_free(user_pwd->id);
```

NULL Pointer Dereference\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=169
Status	New

The variable declared in null at freebsd-src-4/srp_vfy.c in line 382 is not initialized when it is used by id at freebsd-src-4/srp_vfy.c in line 176.

	Source	Destination
File	freebsd-src-4/srp_vfy.c	freebsd-src-4/srp_vfy.c
Line	455	182
Object	null	id

Code Snippet
File Name freebsd-src-4/srp_vfy.c
Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
455.                  user_pwd = NULL; /* abandon responsibility */
```

File Name freebsd-src-4/srp_vfy.c

Method void SRP_user_pwd_free(SRP_user_pwd *user_pwd)

```
....
182.      OPENSSL_free(user_pwd->id);
```

NULL Pointer Dereference\Path 23:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=170
Status	New

The variable declared in null at freebsd-src-4/srp_vfy.c in line 382 is not initialized when it is used by v at freebsd-src-4/srp_vfy.c in line 176.

	Source	Destination
File	freebsd-src-4/srp_vfy.c	freebsd-src-4/srp_vfy.c
Line	391	181
Object	null	v

Code Snippet

File Name freebsd-src-4/srp_vfy.c

Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
391.      SRP_user_pwd *user_pwd = NULL;
```

File Name freebsd-src-4/srp_vfy.c

Method void SRP_user_pwd_free(SRP_user_pwd *user_pwd)

```
....
181.      BN_clear_free(user_pwd->v);
```

NULL Pointer Dereference\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=171
Status	New

The variable declared in null at freebsd-src-4/srp_vfy.c in line 382 is not initialized when it is used by v at freebsd-src-4/srp_vfy.c in line 176.

	Source	Destination
File	freebsd-src-4/srp_vfy.c	freebsd-src-4/srp_vfy.c
Line	455	181
Object	null	v

Code Snippet

File Name freebsd-src-4/srp_vfy.c

Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
455.                    user_pwd = NULL; /* abandon responsibility */
```

File Name freebsd-src-4/srp_vfy.c

Method void SRP_user_pwd_free(SRP_user_pwd *user_pwd)

```
....
181.        BN_clear_free(user_pwd->v);
```

NULL Pointer Dereference\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=172
Status	New

The variable declared in null at freebsd-src-4/srp_vfy.c in line 382 is not initialized when it is used by s at freebsd-src-4/srp_vfy.c in line 176.

	Source	Destination
File	freebsd-src-4/srp_vfy.c	freebsd-src-4/srp_vfy.c
Line	391	180
Object	null	s

Code Snippet

File Name freebsd-src-4/srp_vfy.c

Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
391.        SRP_user_pwd *user_pwd = NULL;
```

File Name freebsd-src-4/srp_vfy.c

Method void SRP_user_pwd_free(SRP_user_pwd *user_pwd)

```
....
180.        BN_free(user_pwd->s);
```

NULL Pointer Dereference\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=173
Status	New

The variable declared in null at freebsd-src-4/srp_vfy.c in line 382 is not initialized when it is used by s at freebsd-src-4/srp_vfy.c in line 176.

	Source	Destination
File	freebsd-src-4/srp_vfy.c	freebsd-src-4/srp_vfy.c
Line	455	180
Object	null	s

Code Snippet
File Name freebsd-src-4/srp_vfy.c
Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
455.                    user_pwd = NULL; /* abandon responsibility */
```

File Name freebsd-src-4/srp_vfy.c

Method void SRP_user_pwd_free(SRP_user_pwd *user_pwd)

```
....
180.      BN_free(user_pwd->s);
```

NULL Pointer Dereference\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=174
Status	New

The variable declared in gN at freebsd-src-4/srp_vfy.c in line 382 is not initialized when it is used by N at freebsd-src-4/srp_vfy.c in line 382.

	Source	Destination
File	freebsd-src-4/srp_vfy.c	freebsd-src-4/srp_vfy.c
Line	390	422
Object	gN	N

Code Snippet
File Name freebsd-src-4/srp_vfy.c
Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
390.      SRP_gN *gN = NULL;
....
422.                    || (gN->N = SRP_gN_place_bn(vb->gN_cache,
pp[DB_srpverifier]))
```

NULL Pointer Dereference\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=175
Status	New

The variable declared in gN at freebsd-src-4/srp_vfy.c in line 382 is not initialized when it is used by id at freebsd-src-4/srp_vfy.c in line 382.

	Source	Destination
File	freebsd-src-4/srp_vfy.c	freebsd-src-4/srp_vfy.c
Line	390	421
Object	gN	id

Code Snippet
File Name freebsd-src-4/srp_vfy.c
Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
390.      SRP_gN *gN = NULL;
....
421.              if ((gN->id = OPENSSL_strdup(pp[DB_srpid])) == NULL
```

NULL Pointer Dereference\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=176
Status	New

The variable declared in gN at freebsd-src-4/srp_vfy.c in line 382 is not initialized when it is used by g at freebsd-src-4/srp_vfy.c in line 382.

	Source	Destination
File	freebsd-src-4/srp_vfy.c	freebsd-src-4/srp_vfy.c
Line	390	424
Object	gN	g

Code Snippet
File Name freebsd-src-4/srp_vfy.c
Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
390.      SRP_gN *gN = NULL;
....
424.                  || (gN->g = SRP_gN_place_bn(vb->gN_cache,
pp[DB_srpsalt]))
```

NULL Pointer Dereference\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=177
Status	New

The variable declared in gN at freebsd-src-4/srp_vfy.c in line 382 is not initialized when it is used by g at freebsd-src-4/srp_vfy.c in line 382.

	Source	Destination
File	freebsd-src-4/srp_vfy.c	freebsd-src-4/srp_vfy.c
Line	390	467
Object	gN	g

Code Snippet
File Name freebsd-src-4/srp_vfy.c
Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
390.      SRP_gN *gN = NULL;
....
467.          vb->default_g = gN->g;
```

NULL Pointer Dereference\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=178
Status	New

The variable declared in gN at freebsd-src-4/srp_vfy.c in line 382 is not initialized when it is used by N at freebsd-src-4/srp_vfy.c in line 382.

	Source	Destination
File	freebsd-src-4/srp_vfy.c	freebsd-src-4/srp_vfy.c
Line	390	468
Object	gN	N

Code Snippet
File Name freebsd-src-4/srp_vfy.c
Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
390.      SRP_gN *gN = NULL;
....
468.          vb->default_N = gN->N;
```

NULL Pointer Dereference\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=179
Status	New

The variable declared in gN at freebsd-src-4/srp_vfy.c in line 382 is not initialized when it is used by id at freebsd-src-4/srp_vfy.c in line 382.

	Source	Destination
File	freebsd-src-4/srp_vfy.c	freebsd-src-4/srp_vfy.c
Line	390	480
Object	gN	id

Code Snippet
File Name freebsd-src-4/srp_vfy.c
Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
390.      SRP_gN *gN = NULL;
....
480.          OPENSSL_free(gN->id);
```

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=1070094&projectid=70084&pathid=13
Status	New

The main method calls the snprintf function, at line 1307 of freebsd-src-4/dtrace.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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1919	1919
Object	snprintf	snprintf

Code Snippet	
File Name	freebsd-src-4/dtrace.c
Method	main(int argc, char *argv[])

```
....
1919.                    (void) snprintf(p, sizeof (g_cmdv[0].dc_ofile),
```

Unchecked Return Value\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=14
Status	New

The link_prog method calls the snprintf function, at line 719 of freebsd-src-4/dtrace.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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	728	728
Object	snprintf	snprintf

Code Snippet	
File Name	freebsd-src-4/dtrace.c
Method	link_prog(dtrace_cmd_t *dcp)

```
....
728.                (void) snprintf(dcp->dc_ofile, sizeof (dcp->dc_ofile),
```

Unchecked Return Value\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=15
Status	New

The link_prog method calls the snprintf function, at line 719 of freebsd-src-4/dtrace.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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	731	731
Object	snprintf	snprintf

Code Snippet

File Name	freebsd-src-4/dtrace.c
Method	link_prog(dtrace_cmd_t *dcp)

```
....
731.              (void) snprintf(dcp->dc_ofile, sizeof (dcp->dc_ofile),
```

Unchecked Return Value\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=16
Status	New

The link_prog method calls the snprintf function, at line 719 of freebsd-src-4/dtrace.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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	734	734
Object	snprintf	snprintf

Code Snippet	
File Name	freebsd-src-4/dtrace.c
Method	link_prog(dtrace_cmd_t *dcp)

```
....
734.              (void) snprintf(dcp->dc_ofile, sizeof (dcp->dc_ofile),
```

Unchecked Return Value\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=17
Status	New

The bufhandler method calls the snprintf function, at line 961 of freebsd-src-4/dtrace.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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	999	999
Object	snprintf	snprintf

Code Snippet	
File Name	freebsd-src-4/dtrace.c

Method	bufhandler(const dtrace_bufdata_t *bufdata, void *arg)

```
....
999.        (void) snprintf(c, end - c, "0x%x ", bufdata->dtbda_flags);
```

Unchecked Return Value\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=18
Status	New

The bufhandler method calls the snprintf function, at line 961 of freebsd-src-4/dtrace.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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1006	1006
Object	snprintf	snprintf

Code Snippet	
File Name	freebsd-src-4/dtrace.c
Method	bufhandler(const dtrace_bufdata_t *bufdata, void *arg)

```
....
1006.             (void) snprintf(c, end - c,
```

Unchecked Return Value\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=19
Status	New

The bufhandler method calls the snprintf function, at line 961 of freebsd-src-4/dtrace.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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1013	1013
Object	snprintf	snprintf

Code Snippet	
File Name	freebsd-src-4/dtrace.c
Method	bufhandler(const dtrace_bufdata_t *bufdata, void *arg)

```
....
1013.              (void) snprintf(c, end - c, ")");
```

Unchecked Return Value\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=20
Status	New

The bufhandler method calls the sprintf function, at line 961 of freebsd-src-4/dtrace.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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1036	1036
Object	sprintf	sprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method bufhandler(const dtrace_bufdata_t *bufdata, void *arg)

```
....
1036.                    (void) sprintf(buf, "%d (data: ", rec-
>dtrd_offset);
```

Unchecked Return Value\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=21
Status	New

The bufhandler method calls the snprintf function, at line 961 of freebsd-src-4/dtrace.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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1045	1045
Object	snprintf	snprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method bufhandler(const dtrace_bufdata_t *bufdata, void *arg)

```
....
1045.                          (void) snprintf(c, end - c, "%s%02x",
```

Unchecked Return Value\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=22
Status	New

The bufhandler method calls the snprintf function, at line 961 of freebsd-src-4/dtrace.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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1050	1050
Object	snprintf	snprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method bufhandler(const dtrace_bufdata_t *bufdata, void *arg)

```
....
1050.                    (void) snprintf(c, end - c,
```

Unchecked Return Value\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=23
Status	New

The chew method calls the snprintf function, at line 1105 of freebsd-src-4/dtrace.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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1132	1132
Object	snprintf	snprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method chew(const dtrace_probedata_t *data, void *arg)

```
....
1132.                    (void) snprintf(name, sizeof (name), "%s:%s",
```

Unchecked Return Value\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=24
Status	New

The chew method calls the snprintf function, at line 1105 of freebsd-src-4/dtrace.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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1145	1145
Object	snprintf	snprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method chew(const dtrace_probedata_t *data, void *arg)

```
....
1145.                    (void) snprintf(name, len, "%*s%s%s:%s", indent,
"",
```

Unchecked Return Value\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=25
Status	New

The chew method calls the snprintf function, at line 1105 of freebsd-src-4/dtrace.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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1151	1151
Object	snprintf	snprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method chew(const dtrace_probedata_t *data, void *arg)

```
....
1151.                   (void) snprintf(name, len, "%*s%s%s", indent,
"",
```

Unchecked Return Value\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=26
Status	New

The local_zone_print method calls the snprintf function, at line 75 of freebsd-src-4/localzone.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	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	79	79
Object	snprintf	snprintf

Code Snippet
File Name freebsd-src-4/localzone.c
Method local_zone_print(struct local_zone* z)

```
....
79.   snprintf(buf, sizeof(buf), "%s zone",
```

Unchecked Return Value\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=27
Status	New

The add_empty_default method calls the snprintf function, at line 845 of freebsd-src-4/localzone.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	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	854	854
Object	snprintf	snprintf

Code Snippet
File Name freebsd-src-4/localzone.c
Method add_empty_default(struct local_zones* zones, struct config_file* cfg,

```
....
854.        snprintf(str, sizeof(str), "%s 10800 IN SOA localhost. "
```

Unchecked Return Value\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=28
Status	New

The add_empty_default method calls the snprintf function, at line 845 of freebsd-src-4/localzone.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	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	860	860
Object	snprintf	snprintf

Code Snippet
File Name freebsd-src-4/localzone.c
Method add_empty_default(struct local_zones* zones, struct config_file* cfg,

```
....
860.        snprintf(str, sizeof(str), "%s 10800 IN NS localhost. ",
name);
```

Unchecked Return Value\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=29
Status	New

The local_data_find_tag_datas method calls the snprintf function, at line 1354 of freebsd-src-4/localzone.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	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1369	1369
Object	snprintf	snprintf

Code Snippet
File Name freebsd-src-4/localzone.c
Method local_data_find_tag_datas(const struct query_info* qinfo,

```
....
1369.             snprintf(buf, sizeof(buf), ". %s", p->str);
```

Unchecked Return Value\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=30
Status	New

The lz_inform_print method calls the snprintf function, at line 1746 of freebsd-src-4/localzone.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	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1754	1754
Object	snprintf	snprintf

Code Snippet
File Name freebsd-src-4/localzone.c
Method lz_inform_print(struct local_zone* z, struct query_info* qinfo,

```
....
1754.       snprintf(txt, sizeof(txt), "%s %s %s@%u", zname,
local_zone_type2str(z->type), ip,
```

Unchecked Return Value\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=31
Status	New

The dump_client_config method calls the snprintf function, at line 3269 of freebsd-src-4/readconf.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	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	3473	3473
Object	snprintf	snprintf

Code Snippet
File Name freebsd-src-4/readconf.c
Method dump_client_config(Options *o, const char *host)

```
....
3473.             snprintf(buf, sizeof(buf), "%d", o->jump_port);
```

Unchecked Return Value\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=32
Status	New

The match_cfg_line method calls the snprintf function, at line 591 of freebsd-src-4/readconf.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	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	698	698
Object	snprintf	snprintf

Code Snippet
File Name freebsd-src-4/readconf.c
Method match_cfg_line(Options *options, char **condition, struct passwd *pw,

```
....
698.                     snprintf(portstr, sizeof(portstr), "%d", port);
```

Unchecked Return Value\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=33
Status	New

The match_cfg_line method calls the snprintf function, at line 591 of freebsd-src-4/readconf.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	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	699	699
Object	snprintf	snprintf

Code Snippet
File Name freebsd-src-4/readconf.c
Method match_cfg_line(Options *options, char **condition, struct passwd *pw,

```
....
699.                    snprintf(uidstr, sizeof(uidstr), "%llu",
```

Unchecked Return Value\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=34
Status	New

The process_config_line_depth method calls the snprintf function, at line 948 of freebsd-src-4/readconf.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	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	1553	1553
Object	snprintf	snprintf

Code Snippet
File Name freebsd-src-4/readconf.c
Method process_config_line_depth(Options *options, struct passwd *pw, const char *host,

```
....
1553.                           snprintf(fwdarg, sizeof(fwdarg), "%s:%s",
arg,
```

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=1070094&projectid=70084&pathid=189
Status	New

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1048	1048
Object	blacklist	sizeof

Code Snippet
File Name freebsd-src-4/ed25519_ref10.c

Method	ge25519_has_small_order(const unsigned char s[32])

```
....
1048.      for (i = 0; i < sizeof blacklist / sizeof blacklist[0]; i++)
{
```

Sizeof Pointer Argument\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=190
Status	New

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1042	1048
Object	blacklist	sizeof

Code Snippet	
File Name	freebsd-src-4/ed25519_ref10.c
Method	ge25519_has_small_order(const unsigned char s[32])

```
....
1042.      COMPILER_ASSERT(7 == sizeof blacklist / sizeof blacklist[0]);
....
1048.      for (i = 0; i < sizeof blacklist / sizeof blacklist[0]; i++)
{
```

Sizeof Pointer Argument\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=191
Status	New

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1044	1048
Object	blacklist	sizeof

Code Snippet	
File Name	freebsd-src-4/ed25519_ref10.c
Method	ge25519_has_small_order(const unsigned char s[32])

```
....
1044.           for (i = 0; i < sizeof blacklist / sizeof blacklist[0];
i++) {
....
1048.       for (i = 0; i < sizeof blacklist / sizeof blacklist[0]; i++)
{
```

Sizeof Pointer Argument\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=192
Status	New

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1042	1048
Object	blacklist	sizeof

Code Snippet
File Name freebsd-src-4/ed25519_ref10.c
Method ge25519_has_small_order(const unsigned char s[32])

```
....
1042.       COMPILER_ASSERT(7 == sizeof blacklist / sizeof blacklist[0]);
....
1048.       for (i = 0; i < sizeof blacklist / sizeof blacklist[0]; i++)
{
```

Sizeof Pointer Argument\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=193
Status	New

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1044	1048
Object	blacklist	sizeof

Code Snippet
File Name freebsd-src-4/ed25519_ref10.c
Method ge25519_has_small_order(const unsigned char s[32])

```
....
1044.          for (i = 0; i < sizeof blacklist / sizeof blacklist[0];
i++) {
....
1048.      for (i = 0; i < sizeof blacklist / sizeof blacklist[0]; i++)
{
```

Sizeof Pointer Argument\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=194
Status	New

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1048	1048
Object	blacklist	sizeof

Code Snippet
File Name freebsd-src-4/ed25519_ref10.c
Method ge25519_has_small_order(const unsigned char s[32])

```
....
1048.      for (i = 0; i < sizeof blacklist / sizeof blacklist[0]; i++)
{
```

Sizeof Pointer Argument\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=195
Status	New

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1052	1052
Object	blacklist	sizeof

Code Snippet
File Name freebsd-src-4/ed25519_ref10.c
Method ge25519_has_small_order(const unsigned char s[32])

```
....
1052.      for (i = 0; i < sizeof blacklist / sizeof blacklist[0]; i++)
{
```

Sizeof Pointer Argument\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=196
Status	New

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1048	1052
Object	blacklist	sizeof

Code Snippet
File Name freebsd-src-4/ed25519_ref10.c
Method ge25519_has_small_order(const unsigned char s[32])

```
....
1048.       for (i = 0; i < sizeof blacklist / sizeof blacklist[0]; i++)
{
....
1052.       for (i = 0; i < sizeof blacklist / sizeof blacklist[0]; i++)
{
```

Sizeof Pointer Argument\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=197
Status	New

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1042	1052
Object	blacklist	sizeof

Code Snippet
File Name freebsd-src-4/ed25519_ref10.c
Method ge25519_has_small_order(const unsigned char s[32])

```
....
1042.       COMPILER_ASSERT(7 == sizeof blacklist / sizeof blacklist[0]);
....
1052.       for (i = 0; i < sizeof blacklist / sizeof blacklist[0]; i++)
{
```

Sizeof Pointer Argument\Path 10:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=198
Status	New

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1044	1052
Object	blacklist	sizeof

Code Snippet
File Name freebsd-src-4/ed25519_ref10.c
Method ge25519_has_small_order(const unsigned char s[32])

```
....
1044.           for (i = 0; i < sizeof blacklist / sizeof blacklist[0];
i++) {
....
1052.       for (i = 0; i < sizeof blacklist / sizeof blacklist[0]; i++)
{
```

Sizeof Pointer Argument\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=199
Status	New

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1048	1052
Object	blacklist	sizeof

Code Snippet
File Name freebsd-src-4/ed25519_ref10.c
Method ge25519_has_small_order(const unsigned char s[32])

```
....
1048.       for (i = 0; i < sizeof blacklist / sizeof blacklist[0]; i++)
{
....
1052.       for (i = 0; i < sizeof blacklist / sizeof blacklist[0]; i++)
{
```

Sizeof Pointer Argument\Path 12:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=200
Status	New

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1042	1052
Object	blacklist	sizeof

Code Snippet
File Name freebsd-src-4/ed25519_ref10.c
Method ge25519_has_small_order(const unsigned char s[32])

```
....
1042.       COMPILER_ASSERT(7 == sizeof blacklist / sizeof blacklist[0]);
....
1052.       for (i = 0; i < sizeof blacklist / sizeof blacklist[0]; i++)
{
```

Sizeof Pointer Argument\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=201
Status	New

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1044	1052
Object	blacklist	sizeof

Code Snippet
File Name freebsd-src-4/ed25519_ref10.c
Method ge25519_has_small_order(const unsigned char s[32])

```
....
1044.           for (i = 0; i < sizeof blacklist / sizeof blacklist[0];
i++) {
....
1052.       for (i = 0; i < sizeof blacklist / sizeof blacklist[0]; i++)
{
```

Sizeof Pointer Argument\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=202
Status	New

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1052	1052
Object	blacklist	sizeof

Code Snippet
File Name freebsd-src-4/ed25519_ref10.c
Method ge25519_has_small_order(const unsigned char s[32])

```
....
1052.       for (i = 0; i < sizeof blacklist / sizeof blacklist[0]; i++)
{
```

Sizeof Pointer Argument\Path 15:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=203
Status New

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1044	1044
Object	blacklist	sizeof

Code Snippet
File Name freebsd-src-4/ed25519_ref10.c
Method ge25519_has_small_order(const unsigned char s[32])

```
....
1044.           for (i = 0; i < sizeof blacklist / sizeof blacklist[0];
i++) {
```

Sizeof Pointer Argument\Path 16:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=204
Status New

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1042	1044
Object	blacklist	sizeof

Code Snippet
File Name freebsd-src-4/ed25519_ref10.c
Method ge25519_has_small_order(const unsigned char s[32])

```
....
1042.      COMPILER_ASSERT(7 == sizeof blacklist / sizeof blacklist[0]);
....
1044.          for (i = 0; i < sizeof blacklist / sizeof blacklist[0];
i++) {
```

Sizeof Pointer Argument\Path 17:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=205
Status New

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1044	1044
Object	blacklist	sizeof

Code Snippet
File Name freebsd-src-4/ed25519_ref10.c
Method ge25519_has_small_order(const unsigned char s[32])

```
....
1044.          for (i = 0; i < sizeof blacklist / sizeof blacklist[0];
i++) {
```

Sizeof Pointer Argument\Path 18:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=206
Status New

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1042	1044
Object	blacklist	sizeof

Code Snippet
File Name freebsd-src-4/ed25519_ref10.c
Method ge25519_has_small_order(const unsigned char s[32])

```
....
1042.       COMPILER_ASSERT(7 == sizeof blacklist / sizeof blacklist[0]);
....
1044.           for (i = 0; i < sizeof blacklist / sizeof blacklist[0];
i++) {
```

Sizeof Pointer Argument\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=207
Status	New

	Source	Destination
File	freebsd-src-4/ed25519_ref10.c	freebsd-src-4/ed25519_ref10.c
Line	1042	1042
Object	blacklist	sizeof

Code Snippet
File Name freebsd-src-4/ed25519_ref10.c
Method ge25519_has_small_order(const unsigned char s[32])

```
....
1042.       COMPILER_ASSERT(7 == sizeof blacklist / sizeof blacklist[0]);
```

Sizeof Pointer Argument\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=208
Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1919	1919
Object	dc_ofile	sizeof

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1919.                   (void) snprintf(p, sizeof (g_cmdv[0].dc_ofile),
```

Exposure of System Data to Unauthorized Control Sphere

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=1070094&projectid=70084&pathid=385
Status	New

The system data read by ifmcstat_getifmaddrs in the file freebsd-src-4/ifmcstat.c at line 985 is potentially exposed by ifmcstat_getifmaddrs found in freebsd-src-4/ifmcstat.c at line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1126	1126
Object	perror	perror

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1126.                                   perror("getnameinfo");
```

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=1070094&projectid=70084&pathid=386
Status	New

The system data read by ifmcstat_getifmaddrs in the file freebsd-src-4/ifmcstat.c at line 985 is potentially exposed by ifmcstat_getifmaddrs found in freebsd-src-4/ifmcstat.c at line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1153	1153
Object	perror	perror

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1153.                                   perror("sysctlnametomib");
```

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=1070094&projectid=70084&pathid=387
Status	New

The system data read by ifmcstat_getifmaddrs in the file freebsd-src-4/ifmcstat.c at line 985 is potentially exposed by ifmcstat_getifmaddrs found in freebsd-src-4/ifmcstat.c at line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1160	1160
Object	perror	perror

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1160.                                   perror("sysctl
net.inet.igmp.ifinfo");
```

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=1070094&projectid=70084&pathid=388
Status	New

The system data read by ifmcstat_getifmaddrs in the file freebsd-src-4/ifmcstat.c at line 985 is potentially exposed by ifmcstat_getifmaddrs found in freebsd-src-4/ifmcstat.c at line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1178	1178
Object	perror	perror

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1178.                                   perror("sysctlnametomib");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=389
Status	New

The system data read by ifmcstat_getifmaddrs in the file freebsd-src-4/ifmcstat.c at line 985 is potentially exposed by ifmcstat_getifmaddrs found in freebsd-src-4/ifmcstat.c at line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1185	1185
Object	perror	perror

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1185.                                   perror("sysctl
net.inet6.mld.ifinfo");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=390
Status	New

The system data read by ifmcstat_getifmaddrs in the file freebsd-src-4/ifmcstat.c at line 985 is potentially exposed by ifmcstat_getifmaddrs found in freebsd-src-4/ifmcstat.c at line 985.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1209	1209
Object	perror	perror

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1209.                           perror("getnameinfo");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=391
Status	New

The system data read by ifmcstat_kvm in the file freebsd-src-4/ifmcstat.c at line 331 is potentially exposed by ifmcstat_kvm found in freebsd-src-4/ifmcstat.c at line 331.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	338	338
Object	perror	perror

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_kvm(const char *kernel, const char *core)

```
....
338.                perror("kvm_openfiles");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=392
Status	New

The system data read by ifmcstat_kvm in the file freebsd-src-4/ifmcstat.c at line 331 is potentially exposed by ifmcstat_kvm found in freebsd-src-4/ifmcstat.c at line 331.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	342	342
Object	perror	perror

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_kvm(const char *kernel, const char *core)

```
....
342.                perror("kvm_nlist");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=393
Status	New

The system data read by kread in the file freebsd-src-4/ifmcstat.c at line 373 is potentially exposed by kread found in freebsd-src-4/ifmcstat.c at line 373.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	377	377
Object	perror	perror

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method kread(u_long addr, void *buf, int len)

```
....
377.                perror("kvm_read");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=394
Status	New

The system data read by inm_print_sources_sysctl in the file freebsd-src-4/ifmcstat.c at line 794 is potentially exposed by inm_print_sources_sysctl found in freebsd-src-4/ifmcstat.c at line 794.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	812	812
Object	perror	perror

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method inm_print_sources_sysctl(uint32_t ifindex, struct in_addr gina)

```
....
812.              perror("sysctlnametomib");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=395
Status	New

The system data read by inm_print_sources_sysctl in the file freebsd-src-4/ifmcstat.c at line 794 is potentially exposed by inm_print_sources_sysctl found in freebsd-src-4/ifmcstat.c at line 794.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	822	822
Object	perror	perror

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method inm_print_sources_sysctl(uint32_t ifindex, struct in_addr gina)

```
....
822.                    perror("sysctl net.inet.ip.mcast.filters");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=396
Status	New

The system data read by inm_print_sources_sysctl in the file freebsd-src-4/ifmcstat.c at line 794 is potentially exposed by inm_print_sources_sysctl found in freebsd-src-4/ifmcstat.c at line 794.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	826	826
Object	perror	perror

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method inm_print_sources_sysctl(uint32_t ifindex, struct in_addr gina)

```
....
826.                    perror("malloc");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=397
Status	New

The system data read by inm_print_sources_sysctl in the file freebsd-src-4/ifmcstat.c at line 794 is potentially exposed by inm_print_sources_sysctl found in freebsd-src-4/ifmcstat.c at line 794.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	831	831
Object	perror	perror

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method inm_print_sources_sysctl(uint32_t ifindex, struct in_addr gina)

```
....
831.                            perror("sysctl");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=398
Status	New

The system data read by inm_print_sources_sysctl in the file freebsd-src-4/ifmcstat.c at line 794 is potentially exposed by inm_print_sources_sysctl found in freebsd-src-4/ifmcstat.c at line 794.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	841	841
Object	perror	perror

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method inm_print_sources_sysctl(uint32_t ifindex, struct in_addr gina)

```
....
841.              perror("sysctl");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=399
Status	New

The system data read by in6m_print_sources_sysctl in the file freebsd-src-4/ifmcstat.c at line 892 is potentially exposed by in6m_print_sources_sysctl found in freebsd-src-4/ifmcstat.c at line 892.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	913	913
Object	perror	perror

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method in6m_print_sources_sysctl(uint32_t ifindex, struct in6_addr *pgroup)

```
....
913.              perror("sysctlnametomib");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=400
Status	New

The system data read by in6m_print_sources_sysctl in the file freebsd-src-4/ifmcstat.c at line 892 is potentially exposed by in6m_print_sources_sysctl found in freebsd-src-4/ifmcstat.c at line 892.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	926	926
Object	perror	perror

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method in6m_print_sources_sysctl(uint32_t ifindex, struct in6_addr *pgroup)

```
....
926.                    perror("sysctl net.inet6.ip6.mcast.filters");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=401
Status	New

The system data read by in6m_print_sources_sysctl in the file freebsd-src-4/ifmcstat.c at line 892 is potentially exposed by in6m_print_sources_sysctl found in freebsd-src-4/ifmcstat.c at line 892.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	930	930
Object	perror	perror

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method in6m_print_sources_sysctl(uint32_t ifindex, struct in6_addr *pgroup)

```
....
930.                    perror("malloc");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=402
Status	New

The system data read by in6m_print_sources_sysctl in the file freebsd-src-4/ifmcstat.c at line 892 is potentially exposed by in6m_print_sources_sysctl found in freebsd-src-4/ifmcstat.c at line 892.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	935	935
Object	perror	perror

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method in6m_print_sources_sysctl(uint32_t ifindex, struct in6_addr *pgroup)

```
....
935.                            perror("sysctl");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=403
Status	New

The system data read by in6m_print_sources_sysctl in the file freebsd-src-4/ifmcstat.c at line 892 is potentially exposed by in6m_print_sources_sysctl found in freebsd-src-4/ifmcstat.c at line 892.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	945	945
Object	perror	perror

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method in6m_print_sources_sysctl(uint32_t ifindex, struct in6_addr *pgroup)

```
....
945.                perror("sysctl");
```

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=404
Status	New

The system data read by verror in the file freebsd-src-4/dtrace.c at line 187 is potentially exposed by verror found in freebsd-src-4/dtrace.c at line 187.

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	189	195
Object	errno	fprintf

Code Snippet
File Name freebsd-src-4/dtrace.c
Method verror(const char *fmt, va_list ap)

```
....
189.        int error = errno;
....
195.              (void) fprintf(stderr, ": %s\n", strerror(error));
```

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=1070094&projectid=70084&pathid=35
Status	New

	Source	Destination
File	freebsd-src-4/print-ldp.c	freebsd-src-4/print-ldp.c
Line	573	626
Object	ldp_msg_header	sizeof

Code Snippet
File Name freebsd-src-4/print-ldp.c
Method ldp_pdu_print(netdissect_options *ndo,

```
....
573.      const struct ldp_msg_header *ldp_msg_header;
....
626.          if (msg_len < sizeof(struct ldp_msg_header)-4) {
```

Use of Sizeof On a Pointer Type\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=36
Status	New

	Source	Destination
File	freebsd-src-4/print-ldp.c	freebsd-src-4/print-ldp.c
Line	573	620
Object	ldp_msg_header	sizeof

Code Snippet
File Name freebsd-src-4/print-ldp.c
Method ldp_pdu_print(netdissect_options *ndo,

```
....
573.      const struct ldp_msg_header *ldp_msg_header;
....
620.          ND_TCHECK_LEN(tptr, sizeof(struct ldp_msg_header));
```

Use of Sizeof On a Pointer Type\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=37
Status	New

	Source	Destination
File	freebsd-src-4/print-ldp.c	freebsd-src-4/print-ldp.c
Line	573	635
Object	ldp_msg_header	sizeof

Code Snippet
File Name freebsd-src-4/print-ldp.c
Method ldp_pdu_print(netdissect_options *ndo,

```
....
573.      const struct ldp_msg_header *ldp_msg_header;
....
635.                          sizeof(struct ldp_msg_header)-4);
```

Use of Sizeof On a Pointer Type\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=38
Status	New

	Source	Destination
File	freebsd-src-4/print-ldp.c	freebsd-src-4/print-ldp.c
Line	573	649
Object	ldp_msg_header	sizeof

Code Snippet
File Name freebsd-src-4/print-ldp.c
Method ldp_pdu_print(netdissect_options *ndo,

```
....
573.      const struct ldp_msg_header *ldp_msg_header;
....
649.          msg_tptr=tptr+sizeof(struct ldp_msg_header);
```

Use of Sizeof On a Pointer Type\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=39
Status	New

	Source	Destination
File	freebsd-src-4/print-ldp.c	freebsd-src-4/print-ldp.c
Line	573	650
Object	ldp_msg_header	sizeof

Code Snippet
File Name freebsd-src-4/print-ldp.c
Method ldp_pdu_print(netdissect_options *ndo,

```
....
573.      const struct ldp_msg_header *ldp_msg_header;
....
650.          msg_tlen=msg_len-(sizeof(struct ldp_msg_header)-4); /*
Type & Length fields not included */
```

Use of Sizeof On a Pointer Type\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=40
Status	New

	Source	Destination
File	freebsd-src-4/print-ldp.c	freebsd-src-4/print-ldp.c
Line	573	691
Object	ldp_msg_header	sizeof

Code Snippet
File Name freebsd-src-4/print-ldp.c
Method ldp_pdu_print(netdissect_options *ndo,

```
....
573.      const struct ldp_msg_header *ldp_msg_header;
....
691.              print_unknown_data(ndo, tptr+sizeof(struct
ldp_msg_header), "\n\t  ",
```

Use of Sizeof On a Pointer Type\Path 7:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=41
Status	New

	Source	Destination
File	freebsd-src-4/print-ldp.c	freebsd-src-4/print-ldp.c
Line	250	277
Object	ldp_tlv_header	sizeof

Code Snippet
File Name freebsd-src-4/print-ldp.c
Method ldp_tlv_print(netdissect_options *ndo,

```
....
250.      const struct ldp_tlv_header *ldp_tlv_header;
....
277.      tptr+=sizeof(struct ldp_tlv_header);
```

Use of Sizeof On a Pointer Type\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=42
Status	New

	Source	Destination
File	freebsd-src-4/rand_lib.c	freebsd-src-4/rand_lib.c
Line	129	175
Object	drbg	sizeof

Code Snippet
File Name freebsd-src-4/rand_lib.c
Method size_t rand_drbg_get_entropy(RAND_DRBG *drbg,

```
....
129.  size_t rand_drbg_get_entropy(RAND_DRBG *drbg,
....
175.                                      (unsigned char *)&drbg,
sizeof(drbg)) != 0) {
```

Use of Sizeof On a Pointer Type\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=43
Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	310	310
Object	sizeof	sizeof

Code Snippet
File Name freebsd-src-4/dtrace.c
Method make_argv(char *s)

```
....
310.        char **argv = malloc(sizeof (char *) * (strlen(s) / 2 + 1));
```

Use of Sizeof On a Pointer Type\Path 10:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=44
Status New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1334	1334
Object	sizeof	sizeof

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1334.       if ((g_argv = malloc(sizeof (char *) * argc)) == NULL ||
```

Use of Sizeof On a Pointer Type\Path 11:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=45
Status New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1336	1336
Object	sizeof	sizeof

Code Snippet

File Name	freebsd-src-4/dtrace.c
Method	main(int argc, char *argv[])

```
....
1336.           (g_psv = malloc(sizeof (struct ps_prochandle *) * argc))
== NULL)
```

Use of Sizeof On a Pointer Type\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=46
Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1858	1858
Object	sizeof	sizeof

Code Snippet

File Name	freebsd-src-4/dtrace.c
Method	main(int argc, char *argv[])

```
....
1858.                   char **objv = alloca(g_cmdc * sizeof (char *));
```

Use of Sizeof On a Pointer Type\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=47
Status	New

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1392	1392
Object	sizeof	sizeof

Code Snippet

File Name	freebsd-src-4/localzone.c
Method	local_data_find_tag_datas(const struct query_info* qinfo,

```
....
1392.                           + sizeof(size_t) + sizeof(uint8_t*) +
```

Use of Sizeof On a Pointer Type\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=48
Status	New

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1414	1414
Object	sizeof	sizeof

Code Snippet
File Name freebsd-src-4/localzone.c
Method local_data_find_tag_datas(const struct query_info* qinfo,

```
....
1414.                          (d->count+1)*sizeof(uint8_t*));
```

Use of Sizeof On a Pointer Type\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=49
Status	New

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1423	1423
Object	sizeof	sizeof

Code Snippet
File Name freebsd-src-4/localzone.c
Method local_data_find_tag_datas(const struct query_info* qinfo,

```
....
1423.                    memmove(d->rr_data, olddata, d-
>count*sizeof(uint8_t*));
```

Use of Sizeof On a Pointer Type\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=50
Status	New

	Source	Destination
File	freebsd-src-4/localzone.c	freebsd-src-4/localzone.c
Line	1558	1558
Object	sizeof	sizeof

Code Snippet
File Name freebsd-src-4/localzone.c
Method local_data_answer(struct local_zone* z, struct module_env* env,

```
....
1558.                           sizeof(uint8_t*) + sizeof(time_t) +
sizeof(uint16_t)
```

Reliance on DNS Lookups in a Decision

Query Path:
CPP\Cx\CPP Low Visibility\Reliance on DNS Lookups in a Decision Version:0

Categories

FISMA 2014: Identification And Authentication
NIST SP 800-53: SC-23 Session Authenticity (P1)

Description
Reliance on DNS Lookups in a Decision\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=135
Status	New

The ifmcstat_getifmaddrs method performs a reverse DNS lookup with getnameinfo, at line 985 of freebsd-src-4/ifmcstat.c. The application then makes a security decision, error, in freebsd-src-4/ifmcstat.c line 985, even though this hostname is not reliable and can be easily spoofed.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1121	1125
Object	getnameinfo	error

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1121.                           error = getnameinfo(&pifasa->sa,
....
1125.                           if (error)
```

Reliance on DNS Lookups in a Decision\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=136
Status	New

The ifmcstat_getifmaddrs method performs a reverse DNS lookup with getnameinfo, at line 985 of freebsd-src-4/ifmcstat.c. The application then makes a security decision, error, in freebsd-src-4/ifmcstat.c line 194, even though this hostname is not reliable and can be easily spoofed.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1121	274
Object	getnameinfo	error

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1121.                           error = getnameinfo(&pifasa->sa,
```

File Name freebsd-src-4/ifmcstat.c
Method main(int argc, char **argv)

```
....
274.        if (error != 0)
```

Reliance on DNS Lookups in a Decision\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=137
Status	New

The ifmcstat_getifmaddrs method performs a reverse DNS lookup with getnameinfo, at line 985 of freebsd-src-4/ifmcstat.c. The application then makes a security decision, error, in freebsd-src-4/ifmcstat.c line 194, even though this hostname is not reliable and can be easily spoofed.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1206	274
Object	getnameinfo	error

Code Snippet
File Name freebsd-src-4/ifmcstat.c

Method ifmcstat_getifmaddrs(void)

```
....
1206.                   error = getnameinfo(&pgsa->sa, pgsa->sa.sa_len,
```

File Name freebsd-src-4/ifmcstat.c

Method main(int argc, char **argv)

```
....
274.        if (error != 0)
```

Reliance on DNS Lookups in a Decision\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=138
Status	New

The ifmcstat_getifmaddrs method performs a reverse DNS lookup with getnameinfo, at line 985 of freebsd-src-4/ifmcstat.c. The application then makes a security decision, error, in freebsd-src-4/ifmcstat.c line 194, even though this hostname is not reliable and can be easily spoofed.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1236	274
Object	getnameinfo	error

Code Snippet

File Name freebsd-src-4/ifmcstat.c

Method ifmcstat_getifmaddrs(void)

```
....
1236.                   error = getnameinfo(&pllsa->sa, pllsa-
>sa.sa_len,
```

File Name freebsd-src-4/ifmcstat.c

Method main(int argc, char **argv)

```
....
274.        if (error != 0)
```

Reliance on DNS Lookups in a Decision\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=700

	84&pathid=139
Status	New

The ifmcstat_getifmaddrs method performs a reverse DNS lookup with getnameinfo, at line 985 of freebsd-src-4/ifmcstat.c. The application then makes a security decision, !=, in freebsd-src-4/ifmcstat.c line 194, even though this hostname is not reliable and can be easily spoofed.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1121	274
Object	getnameinfo	!=

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1121.                           error = getnameinfo(&pifasa->sa,
```

File Name freebsd-src-4/ifmcstat.c
Method main(int argc, char **argv)

```
....
274.        if (error != 0)
```

Reliance on DNS Lookups in a Decision\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=140
Status	New

The ifmcstat_getifmaddrs method performs a reverse DNS lookup with getnameinfo, at line 985 of freebsd-src-4/ifmcstat.c. The application then makes a security decision, !=, in freebsd-src-4/ifmcstat.c line 194, even though this hostname is not reliable and can be easily spoofed.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1206	274
Object	getnameinfo	!=

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1206.                   error = getnameinfo(&pgsa->sa, pgsa->sa.sa_len,
```

File Name freebsd-src-4/ifmcstat.c

Method main(int argc, char **argv)

```
....
274.        if (error != 0)
```

Reliance on DNS Lookups in a Decision\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=141
Status	New

The ifmcstat_getifmaddrs method performs a reverse DNS lookup with getnameinfo, at line 985 of freebsd-src-4/ifmcstat.c. The application then makes a security decision, !=, in freebsd-src-4/ifmcstat.c line 194, even though this hostname is not reliable and can be easily spoofed.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1236	274
Object	getnameinfo	!=

Code Snippet

File Name freebsd-src-4/ifmcstat.c

Method ifmcstat_getifmaddrs(void)

```
....
1236.                   error = getnameinfo(&pllsa->sa, pllsa-
>sa.sa_len,
```

File Name freebsd-src-4/ifmcstat.c

Method main(int argc, char **argv)

```
....
274.        if (error != 0)
```

Reliance on DNS Lookups in a Decision\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=142

Status	New

The ifmcstat_getifmaddrs method performs a reverse DNS lookup with getnameinfo, at line 985 of freebsd-src-4/ifmcstat.c. The application then makes a security decision, error, in freebsd-src-4/ifmcstat.c line 985, even though this hostname is not reliable and can be easily spoofed.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	1206	1208
Object	getnameinfo	error

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method ifmcstat_getifmaddrs(void)

```
....
1206.                   error = getnameinfo(&pgsa->sa, pgsa->sa.sa_len,
....
1208.                   if (error)
```

Reliance on DNS Lookups in a Decision\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=143
Status	New

The inet6_n2a method performs a reverse DNS lookup with getnameinfo, at line 769 of freebsd-src-4/ifmcstat.c. The application then makes a security decision, ==, in freebsd-src-4/ifmcstat.c line 769, even though this hostname is not reliable and can be easily spoofed.

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	780	781
Object	getnameinfo	==

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method inet6_n2a(struct in6_addr *p, uint32_t scope_id)

```
....
780.        if (getnameinfo((struct sockaddr *)&sin6, sin6.sin6_len,
781.            buf, sizeof(buf), NULL, 0, niflags) == 0) {
```

TOCTOU

Query Path:
CPP\Cx\CPP Low Visibility\TOCTOU Version:1
Description

TOCTOU\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=405
Status	New

The main method in freebsd-src-4/dtrace.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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1777	1777
Object	fopen	fopen

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1777.             if (g_ofile != NULL && (g_ofp = fopen(g_ofile, "a"))
== NULL)
```

TOCTOU\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=406
Status	New

The main method in freebsd-src-4/dtrace.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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1811	1811
Object	fopen	fopen

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1811.             if ((g_ofp = fopen(g_ofile, "a")) == NULL)
```

TOCTOU\Path 3:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=407
Status	New

The main method in freebsd-src-4/dtrace.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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1872	1872
Object	fopen	fopen

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1872.             if (g_ofile != NULL && (g_ofp = fopen(g_ofile, "a"))
== NULL)
```

TOCTOU\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=408
Status	New

The main method in freebsd-src-4/dtrace.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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1923	1923
Object	fopen	fopen

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1923.             if ((g_ofp = fopen(g_ofile, "w")) == NULL)
```

TOCTOU\Path 5:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=409
Status	New

The compile_file method in freebsd-src-4/dtrace.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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	801	801
Object	fopen	fopen

Code Snippet
File Name freebsd-src-4/dtrace.c
Method compile_file(dtrace_cmd_t *dcp)

```
....
801.        if ((fp = fopen(dcp->dc_arg, "r")) == NULL)
```

TOCTOU\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=410
Status	New

The nischeck method in freebsd-src-4/port-uw.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	freebsd-src-4/port-uw.c	freebsd-src-4/port-uw.c
Line	101	101
Object	fopen	fopen

Code Snippet
File Name freebsd-src-4/port-uw.c
Method nischeck(char *namep)

```
....
101.        if ((fd = fopen (password_file, "r")) == NULL) {
```

TOCTOU\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=411
Status	New

The read_config_file_depth method in freebsd-src-4/readconf.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	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	2262	2262
Object	fopen	fopen

Code Snippet
File Name freebsd-src-4/readconf.c
Method read_config_file_depth(const char *filename, struct passwd *pw,

```
....
2262.        if ((f = fopen(filename, "r")) == NULL)
```

TOCTOU\Path 8:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=412
Status New

The dof_prune method in freebsd-src-4/dtrace.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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	335	335
Object	open	open

Code Snippet
File Name freebsd-src-4/dtrace.c
Method dof_prune(const char *fname)

```
....
335.        if ((fd = open(fname, O_RDONLY)) == -1) {
```

TOCTOU\Path 9:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=413
Status New

The dof_prune method in freebsd-src-4/dtrace.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	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	356	356
Object	open	open

Code Snippet
File Name freebsd-src-4/dtrace.c
Method dof_prune(const char *fname)

```
....
356.        if ((fd = open(fname, O_WRONLY | O_TRUNC)) == -1)
```

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=1070094&projectid=70084&pathid=378
Status New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1777	1777
Object	g_ofp	g_ofp

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1777.            if (g_ofile != NULL && (g_ofp = fopen(g_ofile, "a"))
== NULL)
```

Incorrect Permission Assignment For Critical Resources\Path 2:

Severity Low
Result State To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=379
Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1811	1811
Object	g_ofp	g_ofp

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1811.             if ((g_ofp = fopen(g_ofile, "a")) == NULL)
```

Incorrect Permission Assignment For Critical Resources\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=380
Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1872	1872
Object	g_ofp	g_ofp

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1872.             if (g_ofile != NULL && (g_ofp = fopen(g_ofile, "a"))
== NULL)
```

Incorrect Permission Assignment For Critical Resources\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=381
Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c

Line	1923	1923
Object	g_ofp	g_ofp

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1923.             if ((g_ofp = fopen(g_ofile, "w")) == NULL)
```

Incorrect Permission Assignment For Critical Resources\Path 5:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=382
Status New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	801	801
Object	fp	fp

Code Snippet
File Name freebsd-src-4/dtrace.c
Method compile_file(dtrace_cmd_t *dcp)

```
....
801.         if ((fp = fopen(dcp->dc_arg, "r")) == NULL)
```

Incorrect Permission Assignment For Critical Resources\Path 6:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=383
Status New

	Source	Destination
File	freebsd-src-4/port-uw.c	freebsd-src-4/port-uw.c
Line	101	101
Object	fd	fd

Code Snippet
File Name freebsd-src-4/port-uw.c
Method nischeck(char *namep)

```
....
101.        if ((fd = fopen (password_file, "r")) == NULL) {
```

Incorrect Permission Assignment For Critical Resources\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=384
Status	New

	Source	Destination
File	freebsd-src-4/readconf.c	freebsd-src-4/readconf.c
Line	2262	2262
Object	f	f

Code Snippet
File Name freebsd-src-4/readconf.c
Method read_config_file_depth(const char *filename, struct passwd *pw,

```
....
2262.       if ((f = fopen(filename, "r")) == NULL)
```

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=1070094&projectid=70084&pathid=180
Status	New

The size of the buffer used by main in argc, at line 1307 of freebsd-src-4/dtrace.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 1307 of freebsd-src-4/dtrace.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1307	1336
Object	argc	argc

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1307.  main(int argc, char *argv[])
....
1336.           (g_psv = malloc(sizeof (struct ps_prochandle *) * argc))
== NULL)
```

Heuristic Buffer Overflow malloc\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=181
Status	New

The size of the buffer used by main in BinaryExpr, at line 1307 of freebsd-src-4/dtrace.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 1307 of freebsd-src-4/dtrace.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1307	1336
Object	argc	BinaryExpr

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1307.  main(int argc, char *argv[])
....
1336.           (g_psv = malloc(sizeof (struct ps_prochandle *) * argc))
== NULL)
```

Heuristic Buffer Overflow malloc\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=182
Status	New

The size of the buffer used by main in argc, at line 1307 of freebsd-src-4/dtrace.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 1307 of freebsd-src-4/dtrace.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c

Line	1307	1335
Object	argc	argc

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1307.  main(int argc, char *argv[])
....
1335.           (g_cmdv = malloc(sizeof (dtrace_cmd_t) * argc)) == NULL
||
```

Heuristic Buffer Overflow malloc\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=183
Status	New

The size of the buffer used by main in BinaryExpr, at line 1307 of freebsd-src-4/dtrace.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 1307 of freebsd-src-4/dtrace.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1307	1335
Object	argc	BinaryExpr

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1307.  main(int argc, char *argv[])
....
1335.           (g_cmdv = malloc(sizeof (dtrace_cmd_t) * argc)) == NULL
||
```

Heuristic Buffer Overflow malloc\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=184
Status	New

The size of the buffer used by main in argc, at line 1307 of freebsd-src-4/dtrace.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 1307 of freebsd-src-4/dtrace.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1307	1334
Object	argc	argc

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1307.  main(int argc, char *argv[])
....
1334.        if ((g_argv = malloc(sizeof (char *) * argc)) == NULL ||
```

Heuristic Buffer Overflow malloc\Path 6:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=185
Status New

The size of the buffer used by main in BinaryExpr, at line 1307 of freebsd-src-4/dtrace.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 1307 of freebsd-src-4/dtrace.c, to overwrite the target buffer.

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1307	1334
Object	argc	BinaryExpr

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1307.  main(int argc, char *argv[])
....
1334.        if ((g_argv = malloc(sizeof (char *) * argc)) == NULL ||
```

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=1070094&projectid=70084&pathid=9

Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1352	1352
Object	getopt	getopt

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1352.             while ((c = getopt(argc, argv, DTRACE_OPTSTR)) != -1)
{
```

Inconsistent Implementations\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=10
Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1542	1542
Object	getopt	getopt

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1542.             while ((c = getopt(argc, argv, DTRACE_OPTSTR)) != -1)
{
```

Inconsistent Implementations\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=11
Status	New

	Source	Destination
File	freebsd-src-4/dtrace.c	freebsd-src-4/dtrace.c
Line	1703	1703

Object	getopt	getopt

Code Snippet
File Name freebsd-src-4/dtrace.c
Method main(int argc, char *argv[])

```
....
1703.             while ((c = getopt(argc, argv, DTRACE_OPTSTR)) != -1)
{
```

Inconsistent Implementations\Path 4:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=12
Status New

	Source	Destination
File	freebsd-src-4/ifmcstat.c	freebsd-src-4/ifmcstat.c
Line	202	202
Object	getopt	getopt

Code Snippet
File Name freebsd-src-4/ifmcstat.c
Method main(int argc, char **argv)

```
....
202.        while ((c = getopt(argc, argv, options)) != -1) {
```

Use Of Hardcoded Password

Query Path:
CPP\Cx\CPP Low Visibility\Use Of Hardcoded Password Version:0

Categories

OWASP Top 10 2013: A2-Broken Authentication and Session Management
FISMA 2014: Identification And Authentication
NIST SP 800-53: SC-28 Protection of Information at Rest (P1)
OWASP Top 10 2017: A2-Broken Authentication

Description

Use Of Hardcoded Password\Path 1:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=187
Status New

The application uses a single, hard-coded password strcmp for authentication purposes, either using it to verify users' identities, or to access another remote system. This password at line 53 of freebsd-src-4/port-uw.c appears in the code as plaintext, and cannot be changed without rebuilding the application.

	Source	Destination
File	freebsd-src-4/port-uw.c	freebsd-src-4/port-uw.c
Line	67	67
Object	strcmp	strcmp

Code Snippet
File Name freebsd-src-4/port-uw.c
Method sys_auth_passwd(struct ssh *ssh, const char *password)

```
....
67.   if (strcmp(pw_password, "") == 0 && strcmp(password, "") == 0)
```

Use Of Hardcoded Password\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1070094&projectid=70084&pathid=188
Status	New

The application uses a single, hard-coded password strcmp for authentication purposes, either using it to verify users' identities, or to access another remote system. This password at line 53 of freebsd-src-4/port-uw.c appears in the code as plaintext, and cannot be changed without rebuilding the application.

	Source	Destination
File	freebsd-src-4/port-uw.c	freebsd-src-4/port-uw.c
Line	67	67
Object	strcmp	strcmp

Code Snippet
File Name freebsd-src-4/port-uw.c
Method sys_auth_passwd(struct ssh *ssh, const char *password)

```
....
67.   if (strcmp(pw_password, "") == 0 && strcmp(password, "") == 0)
```

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=1070094&projectid=70084&pathid=209
Status	New

	Source	Destination
File	freebsd-src-4/sctp_auth.c	freebsd-src-4/sctp_auth.c
Line	200	200
Object	index	index

Code Snippet
File Name freebsd-src-4/sctp_auth.c
Method sctp_pack_auth_chunks(const sctp_auth_chklist_t *list, uint8_t *ptr)

```
....
200.                            ptr[index] |= (1 << offset);
```

Buffer Overflow LongString

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.
Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

CPP

Overflowing Buffers

```
const int BUFFER_SIZE = 10;
char buffer[BUFFER_SIZE];

void copyStringToBuffer(char* inputString)
{
    strcpy(buffer, inputString);
}
```

Checked Buffers

```
const int BUFFER_SIZE = 10;
const int MAX_INPUT_SIZE = 256;
char buffer[BUFFER_SIZE];

void copyStringToBuffer(char* inputString)
{
    if (strnlen(inputString, MAX_INPUT_SIZE) < sizeof(buffer))
    {
        strncpy(buffer, inputString, sizeof(buffer));
    }
}
```

Format String Attack

Risk

What might happen

In environments with unmanaged memory, allowing attackers to control format strings could enable them to access areas of memory to which they should not have access, including reading other restricted variables, misrepresenting data, and possibly even overwriting unauthorized areas of memory. It is even possible this could further lead to buffer overflows and arbitrary code execution under certain circumstance.

Cause

How does it happen

The application allows user input to influence the string argument used for formatted print functions. This family of functions expects the first argument to designate the relative format of dynamically constructed output string, including how to represent each of the other arguments.

Allowing an external user or attacker to control this string, allows them to control the functioning of the printing function, and thus to access unexpected areas of memory.

General Recommendations

How to avoid it

Generic Guidance:

- Do not allow user input or any other external data to influence the format strings.
- Ensure that all string format functions are called with a static string as the format parameter, and that the correct number of arguments are passed to the function, according to the static format string.
- Alternatively, validate all user input before using it in the format string parameter to print format functions, and ensure formatting tokens are not included in the input.

Specific Recommendations:

- Do not include user input directly in the format string parameter (often the first or second argument) to formatting functions.
- Alternatively, use controlled information derived from the input, such as size or length, in the format string - but not the actual contents of the input itself.

Source Code Examples

CPP

Dynamic Formatting String - First Parameter of printf

```
printf("Hello, ");
printf(name); // If name contains tokens, it could retrieve arbitrary values from memory or
```

```
cause a crash
```

Static Formatting String - First Parameter of printf is Static

```
printf("Hello, %s", name);
```

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 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 boundcpy WrongSizeParam

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.
Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

MemoryFree on StackVariable

Risk

What might happen

Undefined Behavior may result with a crash. Crashes may give an attacker valuable information about the system and the program internals. Furthermore, it may leave unprotected files (e.g memory) that may be exploited.

Cause

How does it happen

Calling free() on a variable that was not dynamically allocated (e.g. malloc) will result with an Undefined Behavior.

General Recommendations

How to avoid it

Use free() only on dynamically allocated variables in order to prevent unexpected behavior from the compiler.

Source Code Examples

CPP

Bad - Calling free() on a static variable

```
void clean_up(){
  char temp[256];
  do_something();
  free(tmp);
  return;
}
```

Good - Calling free() only on variables that were dynamically allocated

```
void clean_up(){
  char *buff;
  buff = (char*) malloc(1024);
  free(buff);
  return;
}
```

Wrong Size t Allocation

Risk

What might happen

Incorrect allocation of memory may result in unexpected behavior by either overwriting sections of memory with unexpected values. Under certain conditions where both an incorrect allocation of memory and the values being written can be controlled by an attacker, such an issue may result in execution of malicious code.

Cause

How does it happen

Some memory allocation functions require a size value to be provided as a parameter. The allocated size should be derived from the provided value, by providing the length value of the intended source, multiplied by the size of that length. Failure to perform the correct arithmetic to obtain the exact size of the value will likely result in the source overflowing its destination.

General Recommendations

How to avoid it

- Always perform the correct arithmetic to determine size.
- Specifically for memory allocation, calculate the allocation size from the allocation source:
    - Derive the size value from the length of intended source to determine the amount of units to be processed.
    - Always programmatically consider the size of the each unit and their conversion to memory units - for example, by using sizeof() on the unit's type.
    - Memory allocation should be a multiplication of the amount of units being written, times the size of each unit.

Source Code Examples

CPP

Allocating and Assigning Memory without Sizeof Arithmetic

```
int *ptr;
ptr = (int*)malloc(5);
for (int i = 0; i < 5; i++)
{
      ptr[i] = i * 2 + 1;
}
```

Allocating and Assigning Memory with Sizeof Arithmetic

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
```

```
for (int i = 0; i < 5; i++)
{
      ptr[i] = i * 2 + 1;
}
```

Incorrect Arithmetic of Multi-Byte String Allocation

```
wchar_t * dest;
dest = (wchar_t *)malloc(wcslen(source) + 1); // Would not crash for a short "source"
wcscpy((wchar_t *)dest, source);
wprintf(L"Dest: %s\r\n", dest);
```

Correct Arithmetic of Multi-Byte String Allocation

```
wchar_t * dest;
dest = (wchar_t *)malloc((wcslen(source) + 1) * sizeof(wchar_t));
wcscpy((wchar_t *)dest, source);
wprintf(L"Dest: %s\r\n", dest);
```

Integer Overflow

Risk

What might happen

Assigning large data types into smaller data types, without proper checks and explicit casting, will lead to undefined behavior and unintentional effects, such as data corruption (e.g. value wraparound, wherein maximum values become minimum values); system crashes; infinite loops; logic errors, such as bypassing of security mechanisms; or even buffer overflows leading to arbitrary code execution.

Cause

How does it happen

This flaw can occur when implicitly casting numerical data types of a larger size, into a variable with a data type of a smaller size. This forces the program to discard some bits of information from the number. Depending on how the numerical data types are stored in memory, this is often the bits with the highest value, causing substantial corruption of the stored number. Alternatively, the sign bit of a signed integer could be lost, completely reversing the intention of the number.

General Recommendations

How to avoid it

- Avoid casting larger data types to smaller types.
- Prefer promoting the target variable to a large enough data type.
- If downcasting is necessary, always check that values are valid and in range of the target type, before casting

---

Source Code Examples

CPP

Unsafe Downsize Casting

```
int unsafe_addition(short op1, int op2) {

    // op2 gets forced from int into a short
    short total = op1 + op2;

    return total;
}
```

Safer Use of Proper Data Types

```
int safe_addition(short op1, int op2) {

    // total variable is of type int, the largest type that is needed
    int total = 0;

    // check if total will overflow available integer size
    if (INT_MAX - abs(op2) > op1)
```

```
    {
        total = op1 + op2;
    }
    else
    {
        // instead of overflow, saturate (but this is not always a good thing)
        total = INT_MAX
    }

    return total;
}
```

Dangerous Functions

Risk

What might happen

Use of dangerous functions may expose varying risks associated with each particular function, with potential impact of improper usage of these functions varying significantly. The presence of such functions indicates a flaw in code maintenance policies and adherence to secure coding practices, in a way that has allowed introducing known dangerous code into the application.

Cause

How does it happen

A dangerous function has been identified within the code. Functions are often deemed dangerous to use for numerous reasons, as there are different sets of vulnerabilities associated with usage of such functions. For example, some string copy and concatenation functions are vulnerable to Buffer Overflow, Memory Disclosure, Denial of Service and more. Use of these functions is not recommended.

General Recommendations

How to avoid it

- Deploy a secure and recommended alternative to any functions that were identified as dangerous.
    - If no secure alternative is found, conduct further researching and testing to identify whether current usage successfully sanitizes and verifies values, and thus successfully avoids the use-cases for whom the function is indeed dangerous
- Conduct a periodical review of methods that are in use, to ensure that all external libraries and built-in functions are up-to-date and whose use has not been excluded from best secure coding practices.

Source Code Examples

CPP

Buffer Overflow in gets()

```
int main()
{
	char buf[10];

	printf("Please enter your name: ");
	gets(buf); // veryveryverylongname
	if (buf == ACCEPTED_NAME)
	{
		// Do something
	}
	return 0;
}
```

Safe reading from user

```
int main()
{
	char buf[10];

	printf("Please enter your name: ");
	fgets(buf, sizeof(buf), stdin); //setting the amount of bytes to read
	if (buf == ACCEPTED_NAME)
	{
		//Do something
	}
	return 0;
}
```

Unsafe function for string copy

```
int main(int argc, char* argv[])
{
	char buf[10];
	strcpy(buf, argv[1]); // overflow occurs when len(argv[1]) > 10 bytes

	return 0;
}
```

Safe string copy

```
int main(int argc, char* argv[])
{
	char buf[10];
	strncpy(buf, argv[1], sizeof(buf));
	buf[9]= '\0'; //strncpy doesn't NULL terminates

	return 0;
}
```

Unsafe format string

```
int main(int argc, char* argv[])
{
	printf(argv[1]); // If argv[1] contains a format token, such as %s,%x or %d, will cause
an access violation
	return 0;
}
```

Safe format string

```
int main(int argc, char* argv[])
{
      printf("%s", argv[1]); // Second parameter is not a formattable string
      return 0;
}
```

Double Free

Weakness ID: 415 *(Weakness Variant)* **Status:** Draft

Description

Description Summary

The product calls free() twice on the same memory address, potentially leading to modification of unexpected memory locations.

Extended Description

When a program calls free() twice with the same argument, the program's memory management data structures become corrupted. This corruption can cause the program to crash or, in some circumstances, cause two later calls to malloc() to return the same pointer. If malloc() returns the same value twice and the program later gives the attacker control over the data that is written into this doubly-allocated memory, the program becomes vulnerable to a buffer overflow attack.

Alternate Terms

Double-free

Time of Introduction

- Architecture and Design
- Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

Scope	Effect
Access Control	Doubly freeing memory may result in a write-what-where condition, allowing an attacker to execute arbitrary code.

Likelihood of Exploit

Low to Medium

Demonstrative Examples

Example 1

The following code shows a simple example of a double free vulnerability.

(Bad Code)

Example Language: **C**

```
char* ptr = (char*)malloc (SIZE);
...
if (abrt) {
free(ptr);
}
...
free(ptr);
```

Double free vulnerabilities have two common (and sometimes overlapping) causes:

- Error conditions and other exceptional circumstances
- Confusion over which part of the program is responsible for freeing the memory

Although some double free vulnerabilities are not much more complicated than the previous example, most are spread out across hundreds of lines of code or even different files. Programmers seem particularly susceptible to freeing global variables

more than once.

Example 2

While contrived, this code should be exploitable on Linux distributions which do not ship with heap-chunk check summing turned on.

(Bad Code)

Example Language: **C**

```
#include <stdio.h>
#include <unistd.h>
#define BUFSIZE1 512
#define BUFSIZE2 ((BUFSIZE1/2) - 8)

int main(int argc, char **argv) {
char *buf1R1;
char *buf2R1;
char *buf1R2;
buf1R1 = (char *) malloc(BUFSIZE2);
buf2R1 = (char *) malloc(BUFSIZE2);
free(buf1R1);
free(buf2R1);
buf1R2 = (char *) malloc(BUFSIZE1);
strncpy(buf1R2, argv[1], BUFSIZE1-1);
free(buf2R1);
free(buf1R2);
}
```

Observed Examples

Reference	Description
CVE-2004-0642	Double free resultant from certain error conditions.
CVE-2004-0772	Double free resultant from certain error conditions.
CVE-2005-1689	Double free resultant from certain error conditions.
CVE-2003-0545	Double free from invalid ASN.1 encoding.
CVE-2003-1048	Double free from malformed GIF.
CVE-2005-0891	Double free from malformed GIF.
CVE-2002-0059	Double free from malformed compressed data.

Potential Mitigations

Phase: Architecture and Design

Choose a language that provides automatic memory management.

Phase: Implementation

Ensure that each allocation is freed only once. After freeing a chunk, set the pointer to NULL to ensure the pointer cannot be freed again. In complicated error conditions, be sure that clean-up routines respect the state of allocation properly. If the language is object oriented, ensure that object destructors delete each chunk of memory only once.

Phase: Implementation

Use a static analysis tool to find double free instances.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	398	Indicator of Poor Code Quality	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Category	399	Resource Management Errors	**Development Concepts (primary)699**
ChildOf	Category	633	Weaknesses that Affect Memory	**Resource-specific Weaknesses (primary)631**
ChildOf	Weakness Base	666	Operation on Resource in Wrong Phase of	**Research Concepts (primary)1000**

			Lifetime	
ChildOf	Weakness Class	675	Duplicate Operations on Resource	Research Concepts1000
ChildOf	Category	742	CERT C Secure Coding Section 08 - Memory Management (MEM)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
PeerOf	Weakness Base	123	Write-what-where Condition	Research Concepts1000
PeerOf	Weakness Base	416	Use After Free	Development Concepts699 Research Concepts1000
MemberOf	View	630	Weaknesses Examined by SAMATE	**Weaknesses Examined by SAMATE (primary)630**
PeerOf	Weakness Base	364	Signal Handler Race Condition	Research Concepts1000

Relationship Notes

This is usually resultant from another weakness, such as an unhandled error or race condition between threads. It could also be primary to weaknesses such as buffer overflows.

Affected Resources

- Memory

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
PLOVER			DFREE - Double-Free Vulnerability
7 Pernicious Kingdoms			Double Free
CLASP			Doubly freeing memory
CERT C Secure Coding	MEM00-C		Allocate and free memory in the same module, at the same level of abstraction
CERT C Secure Coding	MEM01-C		Store a new value in pointers immediately after free()
CERT C Secure Coding	MEM31-C		Free dynamically allocated memory exactly once

White Box Definitions

A weakness where code path has:

1. start statement that relinquishes a dynamically allocated memory resource
2. end statement that relinquishes the dynamically allocated memory resource

Maintenance Notes

It could be argued that Double Free would be most appropriately located as a child of "Use after Free", but "Use" and "Release" are considered to be distinct operations within vulnerability theory, therefore this is more accurately "Release of a Resource after Expiration or Release", which doesn't exist yet.

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	PLOVER		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Eric Dalci	Cigital	External
	updated Potential Mitigations, Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Description, Maintenance Notes, Relationships, Other Notes, Relationship Notes, Taxonomy Mappings		
2008-11-24	CWE Content Team	MITRE	Internal

	updated Relationships, Taxonomy Mappings		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Other Notes		

BACK TO TOP

Heap Inspection

Risk

What might happen

All variables stored by the application in unencrypted memory can potentially be retrieved by an unauthorized user, with privlieged access to the machine. For example, a privileged attacker could attach a debugger to the running process, or retrieve the process's memory from the swapfile or crash dump file.
Once the attacker finds the user passwords in memory, these can be reused to easily impersonate the user to the system.

Cause

How does it happen

String variables are immutable - in other words, once a string variable is assigned, its value cannot be changed or removed. Thus, these strings may remain around in memory, possibly in multiple locations, for an indefinite period of time until the garbage collector happens to remove it. Sensitive data, such as passwords, will remain exposed in memory as plaintext with no control over their lifetime.

General Recommendations

How to avoid it

Generic Guidance:

- Do not store senstiive data, such as passwords or encryption keys, in memory in plaintext, even for a short period of time.
- Prefer to use specialized classes that store encrypted memory.
- Alternatively, store secrets temporarily in mutable data types, such as byte arrays, and then promptly zeroize the memory locations.

Specific Recommendations - Java:

- Instead of storing passwords in immutable strings, prefer to use an encrypted memory object, such as SealedObject.

Specific Recommendations - .NET:

- Instead of storing passwords in immutable strings, prefer to use an encrypted memory object, such as SecureString or ProtectedData.

Source Code Examples

Java

Plaintext Password in Immutable String

```
class Heap_Inspection
{
  private string password;

  void setPassword()
```

```
  {
      password = System.console().readLine("Enter your password: ");
   }
}
```

Password Protected in Memory

```
class Heap_Inspection_Fixed

{
  private SealedObject password;

  void setPassword()

 {
      byte[] sKey = getKeyFromConfig();
      Cipher c = Cipher.getInstance("AES");
      c.init(Cipher.ENCRYPT_MODE, sKey);

      char[] input = System.console().readPassword("Enter your password: ");
      password = new SealedObject(Arrays.asList(input), c);

    //Zero out the possible password, for security.
    Arrays.fill(password, '0');
  }
}
```

CPP

Vulnerable C code

```
/* Vulnerable to heap inspection */

#include <stdio.h>


void somefunc(){
      printf("Yea, I'm just being called for the heap of it..\n");
}

void authfunc(){
        char* password = (char *) malloc(256);
        char ch;
        ssize_t k;
            int i=0;
        while(k = read(0, &ch, 1) > 0)
        {
                if (ch == '\n'){
                        password[i]='\0';
                        break;
                } else{
                        password[i++]=ch;
                        fflush(0);
                }
        }
        printf("Password: %s\n",&password[0]);
}

int main()

{

    printf("Please enter a password:\n");

      authfunc();
      printf("You can now dump memory to find this password!");
      somefunc();
```

```
	gets();

}
```

Safe C code

```
/* Pesumably safe heap */

#include <stdio.h>
#include <string.h>

#define STDIN_FILENO 0

void somefunc(){
        printf("Yea, I'm just being called for the heap of it..\n");
}

void authfunc(){
	char* password = (char*) malloc(256);
	int i=0;
	char ch;
	ssize_t k;
	while(k = read(STDIN_FILENO, &ch, 1) > 0)
	{
		if (ch == '\n'){
			password[i]='\0';
			break;
		} else{
			password[i++]=ch;
			fflush(0);
		}
	}
	i=0;
	memset(password,'\0',256);
}

int main()

{

	printf("Please enter a password:\n");
	authfunc();
	somefunc();
	char ch;
	while(read(STDIN_FILENO, &ch, 1) > 0)
	{
		if (ch == '\n')
			break;
	}
}
```

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 of Uninitialized 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

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());
```

Inadequate Encryption Strength

Risk

What might happen

Using weak or outdated cryptography does not provide sufficient protection for sensitive data. An attacker that gains access to the encrypted data would likely be able to break the encryption, using either cryptanalysis or brute force attacks. Thus, the attacker would be able to steal user passwords and other personal data. This could lead to user impersonation or identity theft.

Cause

How does it happen

The application uses a weak algorithm, that is considered obselete since it is relatively easy to break. These obselete algorithms are vulnerable to several different kinds of attacks, including brute force.

General Recommendations

How to avoid it

Generic Guidance:

- Always use strong, modern algorithms for encryption, hashing, and so on.
- Do not use weak, outdated, or obsolete algorithms.
- Ensure you select the correct cryptographic mechanism according to the specific requirements.
- Passwords should be protected with a dedicated password protection scheme, such as bcrypt, scrypt, PBKDF2, or Argon2.

Specific Recommendations:

- Do not use SHA-1, MD5, or any other weak hash algorithm to protect passwords or personal data. Instead, use a stronger hash such as SHA-256 when a secure hash is required.
- Do not use DES, Triple-DES, RC2, or any other weak encryption algorithm to protect passwords or personal data. Instead, use a stronger encryption algorithm such as AES to protect personal data.
- Do not use weak encryption modes such as ECB, or rely on insecure defaults. Explicitly specify a stronger encryption mode, such as GCM.
- For symmetric encryption, use a key length of at least 256 bits.

Source Code Examples

Java

Weakly Hashed PII

```
string protectSSN(HttpServletRequest req) {
    string socialSecurityNum = req.getParameter("SocialSecurityNo");

    return DigestUtils.md5Hex(socialSecurityNum);
}
```

Stronger Hash for PII

```
string protectSSN(HttpServletRequest req) {
    string socialSecurityNum = req.getParameter("SocialSecurityNo");

    return DigestUtils.sha256Hex(socialSecurityNum);
}
```

Use of Function with Inconsistent Implementations

Weakness ID: 474 *(Weakness Base)* **Status:** Draft

Description

Description Summary

The code uses a function that has inconsistent implementations across operating systems and versions, which might cause security-relevant portability problems.

Time of Introduction

- Architecture and Design
- Implementation

Applicable Platforms

Languages

C: *(Often)*

PHP: *(Often)*

All

Potential Mitigations

Do not accept inconsistent behavior from the API specifications when the deviant behavior increase the risk level.

Other Notes

The behavior of functions in this category varies by operating system, and at times, even by operating system version. Implementation differences can include:

- Slight differences in the way parameters are interpreted leading to inconsistent results.
- Some implementations of the function carry significant security risks.
- The function might not be defined on all platforms.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	398	Indicator of Poor Code Quality	**Development Concepts (primary)699 Seven Pernicious Kingdoms (primary)700 Research Concepts (primary)1000**
ParentOf	Weakness Variant	589	Call to Non-ubiquitous API	**Research Concepts (primary)1000**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
7 Pernicious Kingdoms			Inconsistent Implementations

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	7 Pernicious Kingdoms		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Eric Dalci	Cigital	External
	updated Potential Mitigations, Time of Introduction		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Relationships, Other Notes, Taxonomy Mappings		
Previous Entry Names			
Change Date	**Previous Entry Name**		
2008-04-11	Inconsistent Implementations		

BACK TO TOP

Unchecked Return Value

Risk

What might happen

A program that does not check function return values could cause the application to enter an undefined state. This could lead to unexpected behavior and unintended consequences, including inconsistent data, system crashes or other error-based exploits.

Cause

How does it happen

The application calls a system function, but does not receive or check the result of this funciton. These functions often return error codes in the result, or share other status codes with it's caller. The application simply ignores this result value, losing this vital information.

General Recommendations

How to avoid it

- Always check the result of any called function that returns a value, and verify the result is an expected value.

- Ensure the calling function responds to all possible return values.

- Expect runtime errors and handle them gracefully. Explicitly define a mechanism for handling unexpected errors.

Source Code Examples

CPP

Unchecked Memory Allocation

```
buff = (char*) malloc(size);
strncpy(buff, source, size);
```

Safer Memory Allocation

```
buff = (char*) malloc(size+1);
if (buff==NULL) exit(1);

strncpy(buff, source, size);
buff[size] = '\0';
```

Use of sizeof() on a Pointer Type

Weakness ID: 467 *(Weakness Variant)* **Status:** Draft

Description

Description Summary

The code calls sizeof() on a malloced pointer type, which always returns the wordsize/8. This can produce an unexpected result if the programmer intended to determine how much memory has been allocated.

Time of Introduction

- Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

Scope	Effect
Integrity	This error can often cause one to allocate a buffer that is much smaller than what is needed, leading to resultant weaknesses such as buffer overflows.

Likelihood of Exploit

High

Demonstrative Examples

Example 1

Care should be taken to ensure sizeof returns the size of the data structure itself, and not the size of the pointer to the data structure.

In this example, sizeof(foo) returns the size of the pointer.

(Bad Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(foo));
```

In this example, sizeof(*foo) returns the size of the data structure and not the size of the pointer.

(Good Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(*foo));
```

Example 2

This example defines a fixed username and password. The AuthenticateUser() function is intended to accept a username and a password from an untrusted user, and check to ensure that it matches the username and password. If the username and password match, AuthenticateUser() is intended to indicate that authentication succeeded.

(Bad Code)

```
/* Ignore CWE-259 (hard-coded password) and CWE-309 (use of password system for authentication) for this example. */

char *username = "admin";
char *pass = "password";

int AuthenticateUser(char *inUser, char *inPass) {
```

```
printf("Sizeof username = %d\n", sizeof(username));
printf("Sizeof pass = %d\n", sizeof(pass));

if (strncmp(username, inUser, sizeof(username))) {
printf("Auth failure of username using sizeof\n");
return(AUTH_FAIL);
}
/* Because of CWE-467, the sizeof returns 4 on many platforms and architectures. */
if (! strncmp(pass, inPass, sizeof(pass))) {
printf("Auth success of password using sizeof\n");
return(AUTH_SUCCESS);
}
else {
printf("Auth fail of password using sizeof\n");
return(AUTH_FAIL);
}
}

int main (int argc, char **argv)
{
int authResult;

if (argc < 3) {
ExitError("Usage: Provide a username and password");
}
authResult = AuthenticateUser(argv[1], argv[2]);
if (authResult != AUTH_SUCCESS) {
ExitError("Authentication failed");
}
else {
DoAuthenticatedTask(argv[1]);
}
}
```

In AuthenticateUser(), because sizeof() is applied to a parameter with an array type, the sizeof() call might return 4 on many modern architectures. As a result, the strncmp() call only checks the first four characters of the input password, resulting in a partial comparison (CWE-187), leading to improper authentication (CWE-287).

Because of the partial comparison, any of these passwords would still cause authentication to succeed for the "admin" user:

(Attack)

```
pass5
passABCDEFGH
passWORD
```

Because only 4 characters are checked, this significantly reduces the search space for an attacker, making brute force attacks more feasible.

The same problem also applies to the username, so values such as "adminXYZ" and "administrator" will succeed for the username.

Potential Mitigations

Phase: Implementation

Use expressions such as "sizeof(*pointer)" instead of "sizeof(pointer)", unless you intend to run sizeof() on a pointer type to gain some platform independence or if you are allocating a variable on the stack.

Other Notes

The use of sizeof() on a pointer can sometimes generate useful information. An obvious case is to find out the wordsize on a platform. More often than not, the appearance of sizeof(pointer) indicates a bug.

Weakness Ordinalities

Ordinality	Description
Primary	*(where the weakness exists independent of other weaknesses)*

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	465	Pointer Issues	**Development Concepts (primary)699**
ChildOf	Weakness Class	682	Incorrect Calculation	**Research Concepts (primary)1000**
ChildOf	Category	737	CERT C Secure Coding Section 03 - Expressions (EXP)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
ChildOf	Category	740	CERT C Secure Coding Section 06 - Arrays (ARR)	Weaknesses Addressed by the CERT C Secure Coding Standard734
CanPrecede	Weakness Base	131	Incorrect Calculation of Buffer Size	Research Concepts1000

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Use of sizeof() on a pointer type
CERT C Secure Coding	ARR01-C		Do not apply the sizeof operator to a pointer when taking the size of an array
CERT C Secure Coding	EXP01-C		Do not take the size of a pointer to determine the size of the pointed-to type

White Box Definitions

A weakness where code path has:

1. end statement that passes an identity of a dynamically allocated memory resource to a sizeof operator
2. start statement that allocates the dynamically allocated memory resource

References

Robert Seacord. "EXP01-A. Do not take the sizeof a pointer to determine the size of a type". <https://www.securecoding.cert.org/confluence/display/seccode/EXP01-A.+Do+not+take+the+sizeof+a+pointer+to+determine+the+size+of+a+type>.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	CLASP		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		

BACK TO TOP

Reliance on DNS Lookups in a Decision

Risk

What might happen

Relying on reverse DNS records, without verifying domain ownership via cryptographic certificates or protocols, is not a sufficient authentication mechanism. Basing any security decisions on the registered hostname could allow an external attacker to control the application flow. The attacker could possibly perform restricted operations, bypass access controls, and even spoof the user's identity, inject a bogus hostname into the security log, and possibly other logic attacks.

Cause

How does it happen

The application performs a reverse DNS resolution, based on the remote IP address, and performs a security check based on the returned hostname. However, it is relatively easy to spoof DNS names, or cause them to be misreported, depending on the context of the specific environment. If the remote server is controlled by the attacker, it can be configured to report a bogus hostname. Additionally, the attacker could also spoof the hostname if she controls the associated DNS server, or by attacking the legitimate DNS server, or by poisoning the server's DNS cache, or by modifying unprotected DNS traffic to the server. Regardless of the vector, a remote attacker can alter the detected network address, faking the authentication details.

General Recommendations

How to avoid it

- Do not rely on DNS records, network addresses, or system hostnames as a form of authentication, or any other security-related decision.
- Do not perform reverse DNS resolution over an unprotected protocol without record validation.
- Implement a proper authentication mechanism, such as passwords, cryptographic certificates, or public key digital signatures.
- Consider using proposed protocol extensions to cryptographically protect DNS, e.g. DNSSEC (though note the limited support and other drawbacks).

Source Code Examples

Java

Using Reverse DNS as Authentication

```
private boolean isInternalEmployee(ServletRequest req) {
      boolean isCompany = false;

      String ip = req.getRemoteAddr();
      InetAddress address = InetAddress.getByName(ip);

      if (address.getHostName().endsWith(COMPANYNAME)) {
            isCompany = true;
    }
      return isCompany;
```

```
}
```

Verify Authenticated User's Identity

```
private boolean isInternalEmployee(ServletRequest req) {
      boolean isCompany = false;

      Principal user = req.getUserPrincipal();
      if (user != null) {
      if (user.getName().startsWith(COMPANYDOMAIN + "\\")) {
            isCompany = true;
         }
     }
      return isCompany;
}
```

NULL Pointer Dereference

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
- Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
- Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.

Source Code Examples

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

Use Of Hardcoded Password

Risk

What might happen

Hardcoded passwords expose the application to password leakage. If an attacker gains access to the source code, she will be able to steal the embedded passwords, and use them to impersonate a valid user. This could include impersonating end users to the application, or impersonating the application to a remote system, such as a database or a remote web service.
Once the attacker succeeds in impersonating the user or application, she will have full access to the system, and be able to do anything the impersonated identity could do.

Cause

How does it happen

The application codebase has string literal passwords embedded in the source code. This hardcoded value is used either to compare to user-provided credentials, or to authenticate downstream to a remote system (such as a database or a remote web service).
An attacker only needs to gain access to the source code to reveal the hardcoded password. Likewise, the attacker can reverse engineer the compiled application binaries, and easily retrieve the embedded password. Once found, the attacker can easily use the password in impersonation attacks, either directly on the application or to the remote system.
Furthermore, once stolen, this password cannot be easily changed to prevent further misuse, unless a new version of the application is compiled. Moreover, if this application is distributed to numerous systems, stealing the password from one system automatically allows a class break in to all the deployed systems.

General Recommendations

How to avoid it

- Do not hardcode any secret data in source code, especially not passwords.
- In particular, user passwords should be stored in a database or directory service, and protected with a strong password hash (e.g. bcrypt, scrypt, PBKDF2, or Argon2). Do not compare user passwords with a hardcoded value.
- Sytem passwords should be stored in a configuration file or the database, and protected with strong encryption (e.g. AES-256). Encryption keys should be securely managed, and not hardcoded.

Source Code Examples

Java

Hardcoded Admin Password

```
bool isAdmin(String username, String password) {
    bool isMatch = false;

    if (username.equals("admin")) {
        if (password.equals("P@ssw0rd"))
            return isMatch = true;
    }

    return isMatch;
```

```
}
```

No Hardcoded Credentials

```
bool isAdmin(String username, String password) {
    bool adminPrivs = false;

    if (authenticateUser(username, password)) {
        UserPrivileges privs = getUserPrivilieges(username);

        if (privs.isAdmin)
            adminPrivs = true;
    }

    return adminPrivs;
}
```

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