minix-1 Scan Report

Project Name	minix-1
Scan Start	Friday, June 21, 2024 3:39:31 PM
Preset	Checkmarx Default
Scan Time	00h:10m:28s
Lines Of Code Scanned	44332
Files Scanned	29
Report Creation Time	Friday, June 21, 2024 4:04:40 PM
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025
Team	CxServer
Checkmarx Version	8.7.0
Scan Type	Full
Source Origin	LocalPath
Density	1/100 (Vulnerabilities/LOC)
Visibility	Public

Filter Settings

Severity

Included: High, Medium, Low, Information

Excluded: None

Result State

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

Excluded: None

Assigned to

Included: All

Categories

Included:

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

Excluded:

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

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

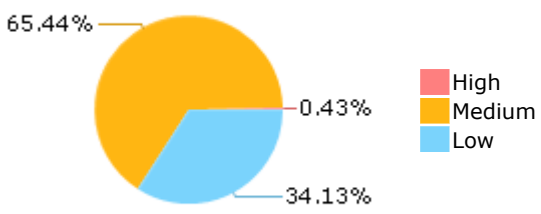
Results Limit

Results limit per query was set to 50

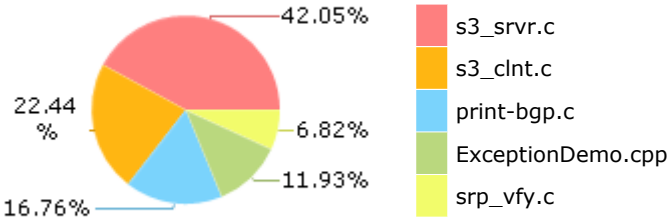
Selected Queries

Selected queries are listed in Result Summary

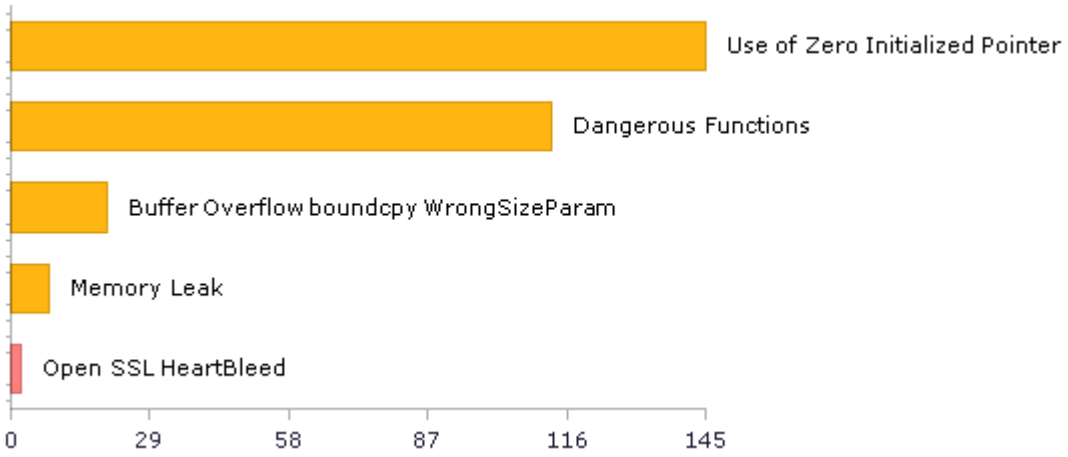
Result Summary



Most Vulnerable Files



Top 5 Vulnerabilities



Scan Summary - OWASP Top 10 2017

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

Category	Threat Agent	Exploitability	Weakness Prevalence	Weakness Detectability	Technical Impact	Business Impact	Issues Found	Best Fix Locations
A1-Injection	App. Specific	EASY	COMMON	EASY	SEVERE	App. Specific	75	37
A2-Broken Authentication	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	50	50
A3-Sensitive Data Exposure	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	App. Specific	1	1
A4-XML External Entities (XXE)	App. Specific	AVERAGE	COMMON	EASY	SEVERE	App. Specific	0	0
A5-Broken Access Control*	App. Specific	AVERAGE	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A6-Security Misconfiguration	App. Specific	EASY	WIDESPREAD	EASY	MODERATE	App. Specific	2	2
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	113	113
A10-Insufficient Logging & Monitoring	App. Specific	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	App. Specific	0	0

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

Scan Summary - OWASP Top 10 2013

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

Category	Threat Agent	Attack Vectors	Weakness Prevalence	Weakness Detectability	Technical Impact	Business Impact	Issues Found	Best Fix Locations
A1-Injection	EXTERNAL, INTERNAL, ADMIN USERS	EASY	COMMON	AVERAGE	SEVERE	ALL DATA	0	0
A2-Broken Authentication and Session Management	EXTERNAL, INTERNAL USERS	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	AFFECTED DATA AND FUNCTIONS	0	0
A3-Cross-Site Scripting (XSS)	EXTERNAL, INTERNAL, ADMIN USERS	AVERAGE	VERY WIDESPREAD	EASY	MODERATE	AFFECTED DATA AND SYSTEM	0	0
A4-Insecure Direct Object References	SYSTEM USERS	EASY	COMMON	EASY	MODERATE	EXPOSED DATA	0	0
A5-Security Misconfiguration	EXTERNAL, INTERNAL, ADMIN USERS	EASY	COMMON	EASY	MODERATE	ALL DATA AND SYSTEM	2	2
A6-Sensitive Data Exposure	EXTERNAL, INTERNAL, ADMIN USERS, USERS BROWSERS	DIFFICULT	UNCOMMON	AVERAGE	SEVERE	EXPOSED DATA	1	1
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	113	113
A10-Unvalidated Redirects and Forwards	USERS BROWSERS	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	AFFECTED DATA AND FUNCTIONS	0	0

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

Scan Summary - PCI DSS v3.2

Category	Issues Found	Best Fix Locations
PCI DSS (3.2) - 6.5.1 - Injection flaws - particularly SQL injection	2	2
PCI DSS (3.2) - 6.5.2 - Buffer overflows	25	25
PCI DSS (3.2) - 6.5.3 - Insecure cryptographic storage	0	0
PCI DSS (3.2) - 6.5.4 - Insecure communications	0	0
PCI DSS (3.2) - 6.5.5 - Improper error handling*	0	0
PCI DSS (3.2) - 6.5.7 - Cross-site scripting (XSS)	0	0
PCI DSS (3.2) - 6.5.8 - Improper access control	0	0
PCI DSS (3.2) - 6.5.9 - Cross-site request forgery	0	0
PCI DSS (3.2) - 6.5.10 - Broken authentication and session management	0	0

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

Scan Summary - FISMA 2014

Category	Description	Issues Found	Best Fix Locations
Access Control	Organizations must limit information system access to authorized users, processes acting on behalf of authorized users, or devices (including other information systems) and to the types of transactions and functions that authorized users are permitted to exercise.	0	0
Audit And Accountability*	Organizations must: (i) create, protect, and retain information system audit records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful, unauthorized, or inappropriate information system activity; and (ii) ensure that the actions of individual information system users can be uniquely traced to those users so they can be held accountable for their actions.	0	0
Configuration Management	Organizations must: (i) establish and maintain baseline configurations and inventories of organizational information systems (including hardware, software, firmware, and documentation) throughout the respective system development life cycles; and (ii) establish and enforce security configuration settings for information technology products employed in organizational information systems.	1	1
Identification And Authentication*	Organizations must identify information system users, processes acting on behalf of users, or devices and authenticate (or verify) the identities of those users, processes, or devices, as a prerequisite to allowing access to organizational information systems.	50	50
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.	1	1
System And Communications Protection	Organizations must: (i) monitor, control, and protect organizational communications (i.e., information transmitted or received by organizational information systems) at the external boundaries and key internal boundaries of the information systems; and (ii) employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational information systems.	0	0
System And Information Integrity	Organizations must: (i) identify, report, and correct information and information system flaws in a timely manner; (ii) provide protection from malicious code at appropriate locations within organizational information systems; and (iii) monitor information system security alerts and advisories and take appropriate actions in response.	5	5

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

Scan Summary - NIST SP 800-53

Category	Issues Found	Best Fix Locations
AC-12 Session Termination (P2)	0	0
AC-3 Access Enforcement (P1)	51	51
AC-4 Information Flow Enforcement (P1)	0	0
AC-6 Least Privilege (P1)	0	0
AU-9 Protection of Audit Information (P1)	0	0
CM-6 Configuration Settings (P2)	0	0
IA-5 Authenticator Management (P1)	0	0
IA-6 Authenticator Feedback (P2)	0	0
IA-8 Identification and Authentication (Non-Organizational Users) (P1)	0	0
SC-12 Cryptographic Key Establishment and Management (P1)	0	0
SC-13 Cryptographic Protection (P1)	0	0
SC-17 Public Key Infrastructure Certificates (P1)	0	0
SC-18 Mobile Code (P2)	0	0
SC-23 Session Authenticity (P1)*	0	0
SC-28 Protection of Information at Rest (P1)	0	0
SC-4 Information in Shared Resources (P1)	1	1
SC-5 Denial of Service Protection (P1)*	208	44
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*	9	9
SI-11 Error Handling (P2)*	37	37
SI-15 Information Output Filtering (P0)	0	0
SI-16 Memory Protection (P1)	2	2

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

Scan Summary - OWASP Mobile Top 10 2016

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

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

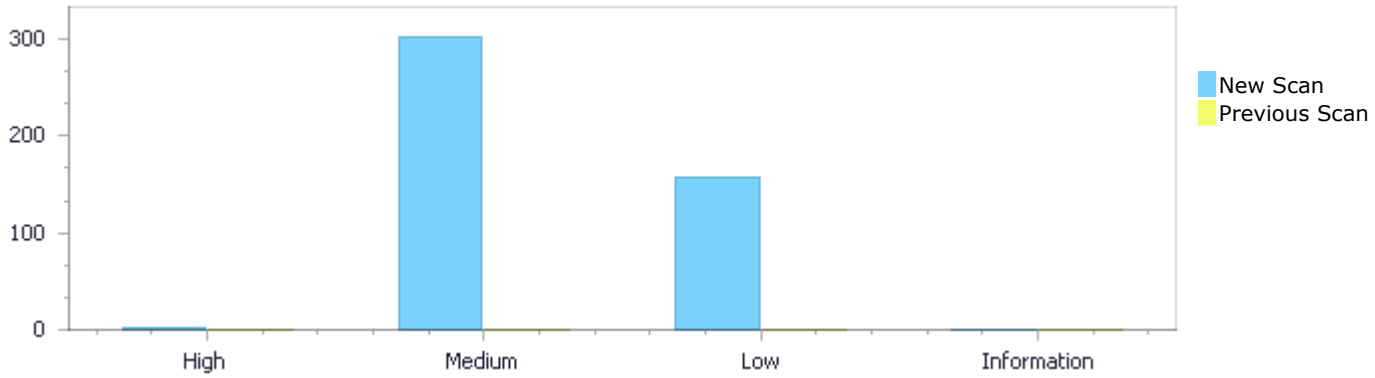
Scan Summary - Custom

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

Results Distribution By Status

First scan of the project

	High	Medium	Low	Information	Total
New Issues	2	303	158	0	463
Recurrent Issues	0	0	0	0	0
Total	2	303	158	0	463
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	2	303	158	0	463
Urgent	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0
Total	2	303	158	0	463

Result Summary

Vulnerability Type	Occurrences	Severity
Open SSL HeartBleed	2	High
Use of Zero Initialized Pointer	145	Medium
Dangerous Functions	113	Medium
Buffer Overflow boundcpy WrongSizeParam	20	Medium
Memory Leak	8	Medium

Integer Overflow	5	Medium
MemoryFree on StackVariable	5	Medium
Wrong Size t Allocation	4	Medium
Use of Uninitialized Variable	2	Medium
Heap Inspection	1	Medium
NULL Pointer Dereference	53	Low
Improper Resource Access Authorization	50	Low
Unchecked Return Value	37	Low
Use of Sizeof On a Pointer Type	9	Low
Sizeof Pointer Argument	4	Low
Potential Off by One Error in Loops	2	Low
Unchecked Array Index	2	Low
Exposure of System Data to Unauthorized Control Sphere	1	Low

10 Most Vulnerable Files

High and Medium Vulnerabilities

File Name	Issues Found
minix-2/s3_srvr.c	130
minix-2/s3_clnt.c	51
minix-2/print-bgp.c	28
minix-2/tree.c	17
minix-2/s2_clnt.c	17
minix-2/print-802_11.c	15
minix-2/str.c	10
minix-2/print-babel.c	9
minix-2/ExceptionDemo.cpp	7
minix-2/rdata.c	6

Scan Results Details

Open SSL HeartBleed

Query Path:
CPP\Cx\CPP Buffer Overflow\Open SSL HeartBleed Version:1

Categories

OWASP Top 10 2013: A5-Security Misconfiguration
NIST SP 800-53: SI-10 Information Input Validation (P1)
OWASP Top 10 2017: A6-Security Misconfiguration

Description

Open SSL HeartBleed\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=1
Status	New

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	2171	2187
Object	ticklen	ticklen

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_new_session_ticket(SSL *s)

```
....
2171.       n2s(p, ticklen);
....
2187.       memcpy(s->session->tlsext_tick, p, ticklen);
```

Open SSL HeartBleed\Path 2:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=2
Status	New

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	2752	2771
Object	i	i

Code Snippet

File Name	minix-2/s3_srvr.c
Method	int ssl3_get_client_key_exchange(SSL *s)

```
....
2752.           n2s(p, i);
....
2771.           memcpy(tmp_id, p, i);
```

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=1030030&projectid=30025&pathid=268
Status	New

The variable declared in exceptionCaughtFlag at minix-2/ExceptionDemo.cpp in line 1126 is not initialized when it is used by exceptionCaughtFlag at minix-2/ExceptionDemo.cpp in line 1126.

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	1185	1215
Object	exceptionCaughtFlag	exceptionCaughtFlag

Code Snippet

File Name	minix-2/ExceptionDemo.cpp
Method	llvm::Function *createCatchWrappedInvokeFunction(llvm::Module &module,

```
....
1185.    llvm::Value *exceptionCaughtFlag = NULL;
....
1215.                                        *exceptionCaughtFlag);
```

Use of Zero Initialized Pointer\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=269
Status	New

The variable declared in init_buf at minix-2/s2_clnt.c in line 144 is not initialized when it is used by buf at minix-2/s2_clnt.c in line 144.

	Source	Destination
File	minix-2/s2_clnt.c	minix-2/s2_clnt.c
Line	287	182
Object	init_buf	buf

Code Snippet
File Name minix-2/s2_clnt.c
Method int ssl2_connect(SSL *s)

```
....
287.                    s->init_buf = NULL;
....
182.                buf = s->init_buf;
```

Use of Zero Initialized Pointer\Path 3:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=270
Status New

The variable declared in buf at minix-2/s2_clnt.c in line 144 is not initialized when it is used by buf at minix-2/s2_clnt.c in line 144.

	Source	Destination
File	minix-2/s2_clnt.c	minix-2/s2_clnt.c
Line	194	182
Object	buf	buf

Code Snippet
File Name minix-2/s2_clnt.c
Method int ssl2_connect(SSL *s)

```
....
194.                buf = NULL;
....
182.                buf = s->init_buf;
```

Use of Zero Initialized Pointer\Path 4:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=271
Status New

The variable declared in buf at minix-2/s2_clnt.c in line 144 is not initialized when it is used by buf at minix-2/s2_clnt.c in line 144.

	Source	Destination
File	minix-2/s2_clnt.c	minix-2/s2_clnt.c
Line	189	182
Object	buf	buf

Code Snippet
File Name minix-2/s2_clnt.c
Method int ssl2_connect(SSL *s)

```
....
189.                        buf = NULL;
....
182.                buf = s->init_buf;
```

Use of Zero Initialized Pointer\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=272
Status	New

The variable declared in pref_cipher at minix-2/s3_clnt.c in line 838 is not initialized when it is used by x at minix-2/s3_clnt.c in line 1099.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	917	1155
Object	pref_cipher	x

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_hello(SSL *s)

```
....
917.          SSL_CIPHER *pref_cipher = NULL;
```

File Name minix-2/s3_clnt.c
Method int ssl3_get_server_certificate(SSL *s)

```
....
1155.           x = d2i_X509(NULL, &q, l);
```

Use of Zero Initialized Pointer\Path 6:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=273
Status	New

The variable declared in psk_identity_hint at minix-2/s3_clnt.c in line 1284 is not initialized when it is used by x at minix-2/s3_clnt.c in line 1099.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	1344	1155
Object	psk_identity_hint	x

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_key_exchange(SSL *s)

```
....
1344.              s->ctx->psk_identity_hint = NULL;
```

File Name minix-2/s3_clnt.c
Method int ssl3_get_server_certificate(SSL *s)

```
....
1155.          x = d2i_X509(NULL, &q, l);
```

Use of Zero Initialized Pointer\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=274
Status	New

The variable declared in peer at minix-2/s3_clnt.c in line 1099 is not initialized when it is used by x at minix-2/s3_clnt.c in line 1099.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	1264	1155
Object	peer	x

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_certificate(SSL *s)

```
....
1264.          s->session->peer = NULL;
....
1155.          x = d2i_X509(NULL, &q, l);
```

Use of Zero Initialized Pointer\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=275
Status	New

The variable declared in sk at minix-2/s3_clnt.c in line 1099 is not initialized when it is used by x at minix-2/s3_clnt.c in line 1099.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	1204	1155
Object	sk	x

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_certificate(SSL *s)

```
....
1204.      sk = NULL;
....
1155.          x = d2i_X509(NULL, &q, l);
```

Use of Zero Initialized Pointer\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=276
Status	New

The variable declared in sk at minix-2/s3_clnt.c in line 1099 is not initialized when it is used by x at minix-2/s3_clnt.c in line 1099.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	1204	1203
Object	sk	x

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_certificate(SSL *s)

```
....
1204.       sk = NULL;
....
1203.       x = sk_X509_value(sk, 0);
```

Use of Zero Initialized Pointer\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=277
Status	New

The variable declared in peer at minix-2/s3_clnt.c in line 1099 is not initialized when it is used by md at minix-2/s3_clnt.c in line 1284.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	1264	1831
Object	peer	md

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_certificate(SSL *s)

```
....
1264.           s->session->peer = NULL;
```

File Name minix-2/s3_clnt.c
Method int ssl3_get_key_exchange(SSL *s)

```
....
1831.               md = tls12_get_hash(p[0]);
```

Use of Zero Initialized Pointer\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=278
Status	New

The variable declared in psk_identity_hint at minix-2/s3_clnt.c in line 1284 is not initialized when it is used by md at minix-2/s3_clnt.c in line 1284.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c

Line	1344	1831
Object	psk_identity_hint	md

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_key_exchange(SSL *s)

```
....
1344.              s->ctx->psk_identity_hint = NULL;
....
1831.              md = tls12_get_hash(p[0]);
```

Use of Zero Initialized Pointer\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=279
Status	New

The variable declared in pref_cipher at minix-2/s3_clnt.c in line 838 is not initialized when it is used by md at minix-2/s3_clnt.c in line 1284.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	917	1831
Object	pref_cipher	md

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_hello(SSL *s)

```
....
917.          SSL_CIPHER *pref_cipher = NULL;
```

File Name minix-2/s3_clnt.c
Method int ssl3_get_key_exchange(SSL *s)

```
....
1831.              md = tls12_get_hash(p[0]);
```

Use of Zero Initialized Pointer\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=280
Status	New

The variable declared in peer at minix-2/s3_clnt.c in line 1099 is not initialized when it is used by t at minix-2/s3_clnt.c in line 2302.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	1264	2969
Object	peer	t

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_certificate(SSL *s)

```
....
1264.           s->session->peer = NULL;
```

File Name minix-2/s3_clnt.c
Method int ssl3_send_client_key_exchange(SSL *s)

```
....
2969.               t += psk_len;
```

Use of Zero Initialized Pointer\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=281
Status	New

The variable declared in pref_cipher at minix-2/s3_clnt.c in line 838 is not initialized when it is used by t at minix-2/s3_clnt.c in line 2302.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	917	2969
Object	pref_cipher	t

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_hello(SSL *s)

```
....
917.           SSL_CIPHER *pref_cipher = NULL;
```

File Name minix-2/s3_clnt.c

Method	int ssl3_send_client_key_exchange(SSL *s)

```
....
2969.               t += psk_len;
```

Use of Zero Initialized Pointer\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=282
Status	New

The variable declared in psk_identity_hint at minix-2/s3_clnt.c in line 1284 is not initialized when it is used by t at minix-2/s3_clnt.c in line 2302.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	1344	2969
Object	psk_identity_hint	t

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_key_exchange(SSL *s)

```
....
1344.               s->ctx->psk_identity_hint = NULL;
```

File Name minix-2/s3_clnt.c
Method int ssl3_send_client_key_exchange(SSL *s)

```
....
2969.               t += psk_len;
```

Use of Zero Initialized Pointer\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=283
Status	New

The variable declared in sk at minix-2/s3_clnt.c in line 1099 is not initialized when it is used by t at minix-2/s3_clnt.c in line 2302.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c

Line	1204	2969
Object	sk	t

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_certificate(SSL *s)

```
....
1204.       sk = NULL;
```

File Name minix-2/s3_clnt.c

Method int ssl3_send_client_key_exchange(SSL *s)

```
....
2969.               t += psk_len;
```

Use of Zero Initialized Pointer\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=284
Status	New

The variable declared in init_buf at minix-2/s3_srvr.c in line 212 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	841	1744
Object	init_buf	dh

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_accept(SSL *s)

```
....
841.               s->init_buf = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1744.                   dh->pub_key = BN_dup(dhp->pub_key);
```

Use of Zero Initialized Pointer\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=285
Status	New

The variable declared in BinaryExpr at minix-2/s3_srvr.c in line 975 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	986	1744
Object	BinaryExpr	dh

Code Snippet

File Name minix-2/s3_srvr.c

Method int ssl3_get_client_hello(SSL *s)

```
....
986.      STACK_OF(SSL_CIPHER) *ciphers = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1744.                   dh->pub_key = BN_dup(dhp->pub_key);
```

Use of Zero Initialized Pointer\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=286
Status	New

The variable declared in ciphers at minix-2/s3_srvr.c in line 975 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1438	1744
Object	ciphers	dh

Code Snippet

File Name minix-2/s3_srvr.c

Method int ssl3_get_client_hello(SSL *s)

```
....
1438.          ciphers = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1744.                  dh->pub_key = BN_dup(dhp->pub_key);
```

Use of Zero Initialized Pointer\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=287
Status	New

The variable declared in ciphers at minix-2/s3_srvr.c in line 975 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1312	1744
Object	ciphers	dh

Code Snippet

File Name minix-2/s3_srvr.c

Method int ssl3_get_client_hello(SSL *s)

```
....
1312.              ciphers = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1744.                  dh->pub_key = BN_dup(dhp->pub_key);
```

Use of Zero Initialized Pointer\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=288
Status	New

The variable declared in handshake_buffer at minix-2/s3_srvr.c in line 2961 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	3163	1744
Object	handshake_buffer	dh

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_cert_verify(SSL *s)

```
....
3163.           s->s3->handshake_buffer = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1744.                   dh->pub_key = BN_dup(dhp->pub_key);
```

Use of Zero Initialized Pointer\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=289
Status	New

The variable declared in rsa at minix-2/s3_srvr.c in line 2182 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	2189	1744
Object	rsa	dh

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_client_key_exchange(SSL *s)

```
....
2189.       RSA *rsa = NULL;
```

File Name minix-2/s3_srvr.c

Method	int ssl3_send_server_key_exchange(SSL *s)

```
....
1744.                   dh->pub_key = BN_dup(dhp->pub_key);
```

Use of Zero Initialized Pointer\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=290
Status	New

The variable declared in init_buf at minix-2/s3_srvr.c in line 212 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	841	1745
Object	init_buf	dh

Code Snippet

File Name	minix-2/s3_srvr.c
Method	int ssl3_accept(SSL *s)

```
....
841.                s->init_buf = NULL;
```

File Name	minix-2/s3_srvr.c
Method	int ssl3_send_server_key_exchange(SSL *s)

```
....
1745.                   dh->priv_key = BN_dup(dhp->priv_key);
```

Use of Zero Initialized Pointer\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=291
Status	New

The variable declared in BinaryExpr at minix-2/s3_srvr.c in line 975 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c

Line	986	1745
Object	BinaryExpr	dh

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_client_hello(SSL *s)

```
....
986.      STACK_OF(SSL_CIPHER) *ciphers = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1745.                  dh->priv_key = BN_dup(dhp->priv_key);
```

Use of Zero Initialized Pointer\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=292
Status	New

The variable declared in ciphers at minix-2/s3_srvr.c in line 975 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1438	1745
Object	ciphers	dh

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_client_hello(SSL *s)

```
....
1438.          ciphers = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1745.                  dh->priv_key = BN_dup(dhp->priv_key);
```

Use of Zero Initialized Pointer\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=293
Status	New

The variable declared in ciphers at minix-2/s3_srvr.c in line 975 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1312	1745
Object	ciphers	dh

Code Snippet

File Name minix-2/s3_srvr.c

Method int ssl3_get_client_hello(SSL *s)

```
....
1312.              ciphers = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1745.                  dh->priv_key = BN_dup(dhp->priv_key);
```

Use of Zero Initialized Pointer\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=294
Status	New

The variable declared in handshake_buffer at minix-2/s3_srvr.c in line 2961 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	3163	1745
Object	handshake_buffer	dh

Code Snippet

File Name minix-2/s3_srvr.c

Method int ssl3_get_cert_verify(SSL *s)

```
....
3163.          s->s3->handshake_buffer = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1745.                  dh->priv_key = BN_dup(dhp->priv_key);
```

Use of Zero Initialized Pointer\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=295
Status	New

The variable declared in init_buf at minix-2/s3_srvr.c in line 212 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	841	1751
Object	init_buf	dh

Code Snippet

File Name minix-2/s3_srvr.c

Method int ssl3_accept(SSL *s)

```
....
841.              s->init_buf = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1751.               r[0] = dh->p;
```

Use of Zero Initialized Pointer\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=296
Status	New

The variable declared in BinaryExpr at minix-2/s3_srvr.c in line 975 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	986	1751
Object	BinaryExpr	dh

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_client_hello(SSL *s)

```
....
986.      STACK_OF(SSL_CIPHER) *ciphers = NULL;
```

File Name minix-2/s3_srvr.c
Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1751.              r[0] = dh->p;
```

Use of Zero Initialized Pointer\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=297
Status	New

The variable declared in ciphers at minix-2/s3_srvr.c in line 975 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1438	1751
Object	ciphers	dh

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_client_hello(SSL *s)

```
....
1438.          ciphers = NULL;
```

File Name minix-2/s3_srvr.c

Method	int ssl3_send_server_key_exchange(SSL *s)

```
....
1751.               r[0] = dh->p;
```

Use of Zero Initialized Pointer\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=298
Status	New

The variable declared in ciphers at minix-2/s3_srvr.c in line 975 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1312	1751
Object	ciphers	dh

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_client_hello(SSL *s)

```
....
1312.               ciphers = NULL;
```

File Name minix-2/s3_srvr.c
Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1751.               r[0] = dh->p;
```

Use of Zero Initialized Pointer\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=299
Status	New

The variable declared in handshake_buffer at minix-2/s3_srvr.c in line 2961 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c

Line	3163	1751
Object	handshake_buffer	dh

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_cert_verify(SSL *s)

```
....
3163.          s->s3->handshake_buffer = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1751.              r[0] = dh->p;
```

Use of Zero Initialized Pointer\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=300
Status	New

The variable declared in init_buf at minix-2/s3_srvr.c in line 212 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	841	1752
Object	init_buf	dh

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_accept(SSL *s)

```
....
841.              s->init_buf = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1752.              r[1] = dh->g;
```

Use of Zero Initialized Pointer\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=301
Status	New

The variable declared in BinaryExpr at minix-2/s3_srvr.c in line 975 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	986	1752
Object	BinaryExpr	dh

Code Snippet

File Name minix-2/s3_srvr.c

Method int ssl3_get_client_hello(SSL *s)

```
....
986.      STACK_OF(SSL_CIPHER) *ciphers = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1752.             r[1] = dh->g;
```

Use of Zero Initialized Pointer\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=302
Status	New

The variable declared in ciphers at minix-2/s3_srvr.c in line 975 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1438	1752
Object	ciphers	dh

Code Snippet

File Name minix-2/s3_srvr.c

Method int ssl3_get_client_hello(SSL *s)

```
....
1438.          ciphers = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1752.              r[1] = dh->g;
```

Use of Zero Initialized Pointer\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=303
Status	New

The variable declared in ciphers at minix-2/s3_srvr.c in line 975 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1312	1752
Object	ciphers	dh

Code Snippet

File Name minix-2/s3_srvr.c

Method int ssl3_get_client_hello(SSL *s)

```
....
1312.              ciphers = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1752.              r[1] = dh->g;
```

Use of Zero Initialized Pointer\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=304
Status	New

The variable declared in handshake_buffer at minix-2/s3_srvr.c in line 2961 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	3163	1752
Object	handshake_buffer	dh

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_cert_verify(SSL *s)

```
....
3163.           s->s3->handshake_buffer = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1752.               r[1] = dh->g;
```

Use of Zero Initialized Pointer\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=305
Status	New

The variable declared in init_buf at minix-2/s3_srvr.c in line 212 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	841	1753
Object	init_buf	dh

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_accept(SSL *s)

```
....
841.              s->init_buf = NULL;
```

File Name minix-2/s3_srvr.c

Method	int ssl3_send_server_key_exchange(SSL *s)

```
....
1753.               r[2] = dh->pub_key;
```

Use of Zero Initialized Pointer\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=306
Status	New

The variable declared in BinaryExpr at minix-2/s3_srvr.c in line 975 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	986	1753
Object	BinaryExpr	dh

Code Snippet

File Name minix-2/s3_srvr.c

Method int ssl3_get_client_hello(SSL *s)

```
....
986.      STACK_OF(SSL_CIPHER) *ciphers = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1753.               r[2] = dh->pub_key;
```

Use of Zero Initialized Pointer\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=307
Status	New

The variable declared in ciphers at minix-2/s3_srvr.c in line 975 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c

Line	1438	1753
Object	ciphers	dh

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_client_hello(SSL *s)

```
....
1438.          ciphers = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1753.              r[2] = dh->pub_key;
```

Use of Zero Initialized Pointer\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=308
Status	New

The variable declared in ciphers at minix-2/s3_srvr.c in line 975 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1312	1753
Object	ciphers	dh

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_client_hello(SSL *s)

```
....
1312.              ciphers = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1753.              r[2] = dh->pub_key;
```

Use of Zero Initialized Pointer\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=309
Status	New

The variable declared in handshake_buffer at minix-2/s3_srvr.c in line 2961 is not initialized when it is used by dh at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	3163	1753
Object	handshake_buffer	dh

Code Snippet

File Name minix-2/s3_srvr.c

Method int ssl3_get_cert_verify(SSL *s)

```
....
3163.           s->s3->handshake_buffer = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1753.               r[2] = dh->pub_key;
```

Use of Zero Initialized Pointer\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=310
Status	New

The variable declared in encodedPoint at minix-2/s3_srvr.c in line 1639 is not initialized when it is used by encodedPoint at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1653	1966
Object	encodedPoint	encodedPoint

Code Snippet

File Name minix-2/s3_srvr.c

Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1653.      unsigned char *encodedPoint = NULL;
....
1966.                        (unsigned char *)encodedPoint, encodedlen);
```

Use of Zero Initialized Pointer\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=311
Status	New

The variable declared in init_buf at minix-2/s3_srvr.c in line 212 is not initialized when it is used by pub at minix-2/s3_srvr.c in line 2182.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	841	2392
Object	init_buf	pub

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_accept(SSL *s)

```
....
841.              s->init_buf = NULL;
```

File Name minix-2/s3_srvr.c
Method int ssl3_get_client_key_exchange(SSL *s)

```
....
2392.          pub = BN_bin2bn(p, i, NULL);
```

Use of Zero Initialized Pointer\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=312
Status	New

The variable declared in BinaryExpr at minix-2/s3_srvr.c in line 975 is not initialized when it is used by pub at minix-2/s3_srvr.c in line 2182.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c

Line	986	2392
Object	BinaryExpr	pub

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_client_hello(SSL *s)

```
....
986.      STACK_OF(SSL_CIPHER) *ciphers = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_get_client_key_exchange(SSL *s)

```
....
2392.          pub = BN_bin2bn(p, i, NULL);
```

Use of Zero Initialized Pointer\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=313
Status	New

The variable declared in ciphers at minix-2/s3_srvr.c in line 975 is not initialized when it is used by pub at minix-2/s3_srvr.c in line 2182.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1438	2392
Object	ciphers	pub

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_client_hello(SSL *s)

```
....
1438.          ciphers = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_get_client_key_exchange(SSL *s)

```
....
2392.          pub = BN_bin2bn(p, i, NULL);
```

Use of Zero Initialized Pointer\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=314
Status	New

The variable declared in ciphers at minix-2/s3_srvr.c in line 975 is not initialized when it is used by pub at minix-2/s3_srvr.c in line 2182.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1312	2392
Object	ciphers	pub

Code Snippet

File Name minix-2/s3_srvr.c

Method int ssl3_get_client_hello(SSL *s)

```
....
1312.              ciphers = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_get_client_key_exchange(SSL *s)

```
....
2392.          pub = BN_bin2bn(p, i, NULL);
```

Use of Zero Initialized Pointer\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=315
Status	New

The variable declared in handshake_buffer at minix-2/s3_srvr.c in line 2961 is not initialized when it is used by pub at minix-2/s3_srvr.c in line 2182.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	3163	2392
Object	handshake_buffer	pub

Code Snippet

File Name minix-2/s3_srvr.c

Method int ssl3_get_cert_verify(SSL *s)

```
....
3163.          s->s3->handshake_buffer = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_get_client_key_exchange(SSL *s)

```
....
2392.          pub = BN_bin2bn(p, i, NULL);
```

Use of Zero Initialized Pointer\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=316
Status	New

The variable declared in rsa at minix-2/s3_srvr.c in line 2182 is not initialized when it is used by pub at minix-2/s3_srvr.c in line 2182.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	2189	2392
Object	rsa	pub

Code Snippet

File Name minix-2/s3_srvr.c

Method int ssl3_get_client_key_exchange(SSL *s)

```
....
2189.      RSA *rsa = NULL;
....
2392.          pub = BN_bin2bn(p, i, NULL);
```

Use of Zero Initialized Pointer\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=317
Status	New

The variable declared in init_buf at minix-2/s3_srvr.c in line 212 is not initialized when it is used by t at minix-2/s3_srvr.c in line 2182.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c

Line	841	2797
Object	init_buf	t

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_accept(SSL *s)

```
....
841.                s->init_buf = NULL;
```

File Name minix-2/s3_srvr.c

Method int ssl3_get_client_key_exchange(SSL *s)

```
....
2797.           t += psk_len;
```

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=1030030&projectid=30025&pathid=32
Status	New

The dangerous function, memcpy, was found in use at line 1298 in minix-2/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	minix-2/print-802_11.c	minix-2/print-802_11.c
Line	1336	1336
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-802_11.c
Method parse_elements(netdissect_options *ndo,

```
....
1336.                   memcpy(&ssid, p + offset, 2);
```

Dangerous Functions\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=33
Status	New

The dangerous function, memcpy, was found in use at line 1298 in minix-2/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	minix-2/print-802_11.c	minix-2/print-802_11.c
Line	1346	1346
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-802_11.c
Method parse_elements(netdissect_options *ndo,

```
....
1346.                          memcpy(&ssid.ssid, p + offset,
ssid.length);
```

Dangerous Functions\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=34
Status	New

The dangerous function, memcpy, was found in use at line 1298 in minix-2/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	minix-2/print-802_11.c	minix-2/print-802_11.c
Line	1364	1364
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-802_11.c
Method parse_elements(netdissect_options *ndo,

```
....
1364.                    memcpy(&challenge, p + offset, 2);
```

Dangerous Functions\Path 4:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=35
Status	New

The dangerous function, memcpy, was found in use at line 1298 in minix-2/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	minix-2/print-802_11.c	minix-2/print-802_11.c
Line	1375	1375
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-802_11.c
Method parse_elements(netdissect_options *ndo,

```
....
1375.                          memcpy(&challenge.text, p + offset,
```

Dangerous Functions\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=36
Status	New

The dangerous function, memcpy, was found in use at line 1298 in minix-2/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	minix-2/print-802_11.c	minix-2/print-802_11.c
Line	1394	1394
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-802_11.c
Method parse_elements(netdissect_options *ndo,

```
....
1394.                    memcpy(&rates, p + offset, 2);
```

Dangerous Functions\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=300

	25&pathid=37
Status	New

The dangerous function, memcpy, was found in use at line 1298 in minix-2/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	minix-2/print-802_11.c	minix-2/print-802_11.c
Line	1404	1404
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-802_11.c
Method parse_elements(netdissect_options *ndo,

```
....
1404.                          memcpy(&rates.rate, p + offset,
rates.length);
```

Dangerous Functions\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=38
Status	New

The dangerous function, memcpy, was found in use at line 1298 in minix-2/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	minix-2/print-802_11.c	minix-2/print-802_11.c
Line	1430	1430
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-802_11.c
Method parse_elements(netdissect_options *ndo,

```
....
1430.                      memcpy(&ds, p + offset, 2);
```

Dangerous Functions\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=39
Status	New

The dangerous function, memcpy, was found in use at line 1298 in minix-2/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	minix-2/print-802_11.c	minix-2/print-802_11.c
Line	1454	1454
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-802_11.c
Method parse_elements(netdissect_options *ndo,

```
....
1454.                    memcpy(&cf, p + offset, 2);
```

Dangerous Functions\Path 9:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=40
Status New

The dangerous function, memcpy, was found in use at line 1298 in minix-2/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	minix-2/print-802_11.c	minix-2/print-802_11.c
Line	1462	1462
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-802_11.c
Method parse_elements(netdissect_options *ndo,

```
....
1462.                    memcpy(&cf.count, p + offset, 6);
```

Dangerous Functions\Path 10:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=41
Status New

The dangerous function, memcpy, was found in use at line 1298 in minix-2/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	minix-2/print-802_11.c	minix-2/print-802_11.c
Line	1478	1478
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-802_11.c
Method parse_elements(netdissect_options *ndo,

```
....
1478.                    memcpy(&tim, p + offset, 2);
```

Dangerous Functions\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=42
Status	New

The dangerous function, memcpy, was found in use at line 1298 in minix-2/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	minix-2/print-802_11.c	minix-2/print-802_11.c
Line	1488	1488
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-802_11.c
Method parse_elements(netdissect_options *ndo,

```
....
1488.                    memcpy(&tim.count, p + offset, 3);
```

Dangerous Functions\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=43
Status	New

The dangerous function, memcpy, was found in use at line 1298 in minix-2/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	minix-2/print-802_11.c	minix-2/print-802_11.c

Line	1492	1492
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-802_11.c
Method parse_elements(netdissect_options *ndo,

```
....
1492.                    memcpy(tim.bitmap, p + (tim.length - 3),
```

Dangerous Functions\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=44
Status	New

The dangerous function, memcpy, was found in use at line 1528 in minix-2/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	minix-2/print-802_11.c	minix-2/print-802_11.c
Line	1543	1543
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-802_11.c
Method handle_beacon(netdissect_options *ndo,

```
....
1543.       memcpy(&pbody.timestamp, p, IEEE802_11_TSTAMP_LEN);
```

Dangerous Functions\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=45
Status	New

The dangerous function, memcpy, was found in use at line 1630 in minix-2/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	minix-2/print-802_11.c	minix-2/print-802_11.c
Line	1651	1651
Object	memcpy	memcpy

Code Snippet	
File Name	minix-2/print-802_11.c
Method	handle_reassoc_request(netdissect_options *ndo,

```
....
1651.       memcpy(&pbody.ap, p+offset, IEEE802_11_AP_LEN);
```

Dangerous Functions\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=46
Status	New

The dangerous function, memcpy, was found in use at line 1690 in minix-2/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	minix-2/print-802_11.c	minix-2/print-802_11.c
Line	1705	1705
Object	memcpy	memcpy

Code Snippet	
File Name	minix-2/print-802_11.c
Method	handle_probe_response(netdissect_options *ndo,

```
....
1705.       memcpy(&pbody.timestamp, p, IEEE802_11_TSTAMP_LEN);
```

Dangerous Functions\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=47
Status	New

The dangerous function, memcpy, was found in use at line 344 in minix-2/print-babel.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-babel.c	minix-2/print-babel.c
Line	517	517
Object	memcpy	memcpy

Code Snippet	
File Name	minix-2/print-babel.c

Method	babel_print_v2(netdissect_options *ndo,

```
....
517.                            memcpy(v4_prefix, prefix, 16);
```

Dangerous Functions\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=48
Status	New

The dangerous function, memcpy, was found in use at line 344 in minix-2/print-babel.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-babel.c	minix-2/print-babel.c
Line	519	519
Object	memcpy	memcpy

Code Snippet	
File Name	minix-2/print-babel.c
Method	babel_print_v2(netdissect_options *ndo,

```
....
519.                            memcpy(v6_prefix, prefix, 16);
```

Dangerous Functions\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=49
Status	New

The dangerous function, memcpy, was found in use at line 176 in minix-2/print-babel.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-babel.c	minix-2/print-babel.c
Line	201	201
Object	memcpy	memcpy

Code Snippet	
File Name	minix-2/print-babel.c
Method	network_prefix(int ae, int plen, unsigned int omitted,

```
....
201.          memcpy(prefix, v4prefix, 12);
```

Dangerous Functions\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=50
Status	New

The dangerous function, memcpy, was found in use at line 176 in minix-2/print-babel.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-babel.c	minix-2/print-babel.c
Line	204	204
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-babel.c
Method network_prefix(int ae, int plen, unsigned int omitted,

```
....
204.              memcpy(prefix, dp, 12 + omitted);
```

Dangerous Functions\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=51
Status	New

The dangerous function, memcpy, was found in use at line 176 in minix-2/print-babel.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-babel.c	minix-2/print-babel.c
Line	207	207
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-babel.c
Method network_prefix(int ae, int plen, unsigned int omitted,

```
....
207.                memcpy(prefix + 12 + omitted, p, pb - omitted);
```

Dangerous Functions\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=52
Status	New

The dangerous function, memcpy, was found in use at line 176 in minix-2/print-babel.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-babel.c	minix-2/print-babel.c
Line	216	216
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-babel.c
Method network_prefix(int ae, int plen, unsigned int omitted,

```
....
216.                memcpy(prefix, dp, omitted);
```

Dangerous Functions\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=53
Status	New

The dangerous function, memcpy, was found in use at line 176 in minix-2/print-babel.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-babel.c	minix-2/print-babel.c
Line	219	219
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-babel.c
Method network_prefix(int ae, int plen, unsigned int omitted,

```
....
219.              memcpy(prefix + omitted, p, pb - omitted);
```

Dangerous Functions\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=54
Status	New

The dangerous function, memcpy, was found in use at line 176 in minix-2/print-babel.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-babel.c	minix-2/print-babel.c
Line	228	228
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-babel.c
Method network_prefix(int ae, int plen, unsigned int omitted,

```
....
228.              memcpy(prefix + 8, p, pb - 8);
```

Dangerous Functions\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=55
Status	New

The dangerous function, memcpy, was found in use at line 176 in minix-2/print-babel.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-babel.c	minix-2/print-babel.c
Line	236	236
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-babel.c
Method network_prefix(int ae, int plen, unsigned int omitted,

```
....
236.      memcpy(p_r, prefix, 16);
```

Dangerous Functions\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=56
Status	New

The dangerous function, memcpy, was found in use at line 2708 in minix-2/print-bgp.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	2750	2750
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-bgp.c
Method bgp_print(netdissect_options *ndo,

```
....
2750.              memcpy(&bgp, p, BGP_SIZE);
```

Dangerous Functions\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=57
Status	New

The dangerous function, memcpy, was found in use at line 489 in minix-2/print-bgp.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	506	506
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-bgp.c
Method decode_prefix4(netdissect_options *ndo,

```
....
506.        memcpy(&addr, &pptr[1], plenbytes);
```

Dangerous Functions\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=58
Status	New

The dangerous function, memcpy, was found in use at line 522 in minix-2/print-bgp.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	554	554
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-bgp.c
Method decode_labeled_prefix4(netdissect_options *ndo,

```
....
554.        memcpy(&addr, &pptr[4], plenbytes);
```

Dangerous Functions\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=59
Status	New

The dangerous function, memcpy, was found in use at line 716 in minix-2/print-bgp.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	740	740
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-bgp.c
Method decode_rt_routing_info(netdissect_options *ndo,

```
....
740.        memcpy(&route_target, &pptr[1], (plen + 7) / 8);
```

Dangerous Functions\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=60
Status	New

The dangerous function, memcpy, was found in use at line 756 in minix-2/print-bgp.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	775	775
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-bgp.c
Method decode_labeled_vpn_prefix4(netdissect_options *ndo,

```
....
775.        memcpy(&addr, &pptr[12], (plen + 7) / 8);
```

Dangerous Functions\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=61
Status	New

The dangerous function, memcpy, was found in use at line 1064 in minix-2/print-bgp.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	1081	1081
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-bgp.c
Method decode_prefix6(netdissect_options *ndo,

```
....
1081.       memcpy(&addr, &pd[1], plenbytes);
```

Dangerous Functions\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=62
Status	New

The dangerous function, memcpy, was found in use at line 1097 in minix-2/print-bgp.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	1120	1120
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-bgp.c
Method decode_labeled_prefix6(netdissect_options *ndo,

```
....
1120.       memcpy(&addr, &pptr[4], plenbytes);
```

Dangerous Functions\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=63
Status	New

The dangerous function, memcpy, was found in use at line 1142 in minix-2/print-bgp.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	1161	1161
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-bgp.c
Method decode_labeled_vpn_prefix6(netdissect_options *ndo,

```
....
1161.       memcpy(&addr, &pptr[12], (plen + 7) / 8);
```

Dangerous Functions\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=64
Status	New

The dangerous function, memcpy, was found in use at line 1182 in minix-2/print-bgp.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	1196	1196
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-bgp.c
Method decode_clnp_prefix(netdissect_options *ndo,

```
....
1196.       memcpy(&addr, &pptr[4], (plen + 7) / 8);
```

Dangerous Functions\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=65
Status	New

The dangerous function, memcpy, was found in use at line 1212 in minix-2/print-bgp.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	1231	1231
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-bgp.c
Method decode_labeled_vpn_clnp_prefix(netdissect_options *ndo,

```
....
1231.       memcpy(&addr, &pptr[12], (plen + 7) / 8);
```

Dangerous Functions\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=66
Status	New

The dangerous function, memcpy, was found in use at line 2315 in minix-2/print-bgp.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	2325	2325
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-bgp.c
Method bgp_open_print(netdissect_options *ndo,

```
....
2325.       memcpy(&bgpo, dat, BGP_OPEN_SIZE);
```

Dangerous Functions\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=67
Status	New

The dangerous function, memcpy, was found in use at line 2315 in minix-2/print-bgp.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	2345	2345
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-bgp.c
Method bgp_open_print(netdissect_options *ndo,

```
....
2345.             memcpy(&bgpopt, &opt[i], BGP_OPT_SIZE);
```

Dangerous Functions\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=68
Status	New

The dangerous function, memcpy, was found in use at line 2380 in minix-2/print-bgp.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	2397	2397
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-bgp.c
Method bgp_update_print(netdissect_options *ndo,

```
....
2397.       memcpy(&bgp, dat, BGP_SIZE);
```

Dangerous Functions\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=69
Status	New

The dangerous function, memcpy, was found in use at line 2550 in minix-2/print-bgp.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	2559	2559
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-bgp.c
Method bgp_notification_print(netdissect_options *ndo,

```
....
2559.       memcpy(&bgpn, dat, BGP_NOTIFICATION_SIZE);
```

Dangerous Functions\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=70
Status	New

The dangerous function, memcpy, was found in use at line 2665 in minix-2/print-bgp.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	2672	2672
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-bgp.c
Method bgp_header_print(netdissect_options *ndo,

```
....
2672.       memcpy(&bgp, dat, BGP_SIZE);
```

Dangerous Functions\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=71
Status	New

The dangerous function, memcpy, was found in use at line 336 in minix-2/print-gre.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-gre.c	minix-2/print-gre.c
Line	359	359
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-gre.c
Method gre_sre_ip_print(netdissect_options *ndo, uint8_t sreoff, uint8_t srelen,

```
....
359.              memcpy(&a, bp, sizeof(a));
```

Dangerous Functions\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=72
Status	New

The dangerous function, memcpy, was found in use at line 413 in minix-2/print-ospf6.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/print-ospf6.c	minix-2/print-ospf6.c
Line	434	434
Object	memcpy	memcpy

Code Snippet
File Name minix-2/print-ospf6.c
Method ospf6_print_lsaprefix(netdissect_options *ndo,

```
....
434.        memcpy(&prefix, lsapp->lsa_p_prefix, wordlen * 4);
```

Dangerous Functions\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=73
Status	New

The dangerous function, memcpy, was found in use at line 334 in minix-2/s2_clnt.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/s2_clnt.c	minix-2/s2_clnt.c
Line	531	531
Object	memcpy	memcpy

Code Snippet
File Name minix-2/s2_clnt.c
Method static int get_server_hello(SSL *s)

```
....
531.      memcpy(s->s2->conn_id, p, s->s2->tmp.conn_id_length);
```

Dangerous Functions\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=74
Status	New

The dangerous function, memcpy, was found in use at line 535 in minix-2/s2_clnt.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/s2_clnt.c	minix-2/s2_clnt.c
Line	573	573
Object	memcpy	memcpy

Code Snippet
File Name minix-2/s2_clnt.c
Method static int client_hello(SSL *s)

```
....
573.              memcpy(d, s->session->session_id, (unsigned int)i);
```

Dangerous Functions\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=75
Status	New

The dangerous function, memcpy, was found in use at line 535 in minix-2/s2_clnt.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/s2_clnt.c	minix-2/s2_clnt.c
Line	586	586
Object	memcpy	memcpy

Code Snippet
File Name minix-2/s2_clnt.c
Method static int client_hello(SSL *s)

```
....
586.          memcpy(d, s->s2->challenge, SSL2_CHALLENGE_LENGTH);
```

Dangerous Functions\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=76
Status	New

The dangerous function, memcpy, was found in use at line 597 in minix-2/s2_clnt.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/s2_clnt.c	minix-2/s2_clnt.c
Line	664	664
Object	memcpy	memcpy

Code Snippet
File Name minix-2/s2_clnt.c
Method static int client_master_key(SSL *s)

```
....
664.          memcpy(d, sess->master_key, (unsigned int)clear);
```

Dangerous Functions\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=77
Status	New

The dangerous function, memcpy, was found in use at line 597 in minix-2/s2_clnt.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/s2_clnt.c	minix-2/s2_clnt.c
Line	692	692
Object	memcpy	memcpy

Code Snippet
File Name minix-2/s2_clnt.c
Method static int client_master_key(SSL *s)

```
....
692.          memcpy(d, sess->key_arg, (unsigned int)karg);
```

Dangerous Functions\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=78
Status	New

The dangerous function, memcpy, was found in use at line 704 in minix-2/s2_clnt.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/s2_clnt.c	minix-2/s2_clnt.c
Line	715	715
Object	memcpy	memcpy

Code Snippet
File Name minix-2/s2_clnt.c
Method static int client_finished(SSL *s)

```
....
715.          memcpy(p, s->s2->conn_id, (unsigned int)s->s2-
>conn_id_length);
```

Dangerous Functions\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=79
Status	New

The dangerous function, memcpy, was found in use at line 922 in minix-2/s2_clnt.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/s2_clnt.c	minix-2/s2_clnt.c
Line	980	980
Object	memcpy	memcpy

Code Snippet
File Name minix-2/s2_clnt.c
Method static int get_server_finished(SSL *s)

```
....
980.          memcpy(s->session->session_id, p + 1,
SSL2_SSL_SESSION_ID_LENGTH);
```

Dangerous Functions\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=80
Status	New

The dangerous function, memcpy, was found in use at line 672 in minix-2/s3_clnt.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	750	750
Object	memcpy	memcpy

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_client_hello(SSL *s)

```
....
750.          memcpy(p, s->s3->client_random, SSL3_RANDOM_SIZE);
```

Dangerous Functions\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=81
Status	New

The dangerous function, memcpy, was found in use at line 672 in minix-2/s3_clnt.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	764	764
Object	memcpy	memcpy

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_client_hello(SSL *s)

```
....
764.              memcpy(p, s->session->session_id, i);
```

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=1030030&projectid=30025&pathid=3
Status	New

The size of the buffer used by mib_create in b, at line 487 of minix-2/tree.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that mib_create passes to b, at line 487 of minix-2/tree.c, to overwrite the target buffer.

	Source	Destination
File	minix-2/tree.c	minix-2/tree.c
Line	747	747
Object	b	b

Code Snippet
File Name minix-2/tree.c
Method mib_create(struct mib_call * call, struct mib_node * parent,

```
....
747.                    memcpy(node->node_data, &b, sizeof(b));
```

Buffer Overflow boundcpy WrongSizeParam\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=4
Status	New

The size of the buffer used by mib_write in b, at line 1160 of minix-2/tree.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that mib_write passes to b, at line 1160 of minix-2/tree.c, to overwrite the target buffer.

	Source	Destination
File	minix-2/tree.c	minix-2/tree.c

Line	1266	1266
Object	b	b

Code Snippet
File Name minix-2/tree.c
Method mib_write(struct mib_call * call, struct mib_node * node,

```
....
1266.                   memcpy(dst, &b[0], sizeof(b[0]));
```

Buffer Overflow boundcpy WrongSizeParam\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=5
Status	New

The size of the buffer used by archive_write_set_format_mtree in ->, at line 1013 of minix-2/archive_write_set_format_mtree.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that archive_write_set_format_mtree passes to ->, at line 1013 of minix-2/archive_write_set_format_mtree.c, to overwrite the target buffer.

	Source	Destination
File	minix-2/archive_write_set_format_mtree.c	minix-2/archive_write_set_format_mtree.c
Line	1029	1029
Object	->	->

Code Snippet
File Name minix-2/archive_write_set_format_mtree.c
Method archive_write_set_format_mtree(struct archive *_a)

```
....
1029.       memset(&(mtree->set), 0, sizeof(mtree->set));
```

Buffer Overflow boundcpy WrongSizeParam\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=6
Status	New

The size of the buffer used by ssl3_send_server_key_exchange in s, at line 1639 of minix-2/s3_srvr.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ssl3_send_server_key_exchange passes to s, at line 1639 of minix-2/s3_srvr.c, to overwrite the target buffer.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c

Line	1978	1978
Object	s	s

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1978.                        strlen(s->ctx->psk_identity_hint));
```

Buffer Overflow boundcpy WrongSizeParam\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=7
Status	New

The size of the buffer used by decode_labeled_prefix4 in plenbytes, at line 522 of minix-2/print-bgp.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that decode_labeled_prefix4 passes to plenbytes, at line 522 of minix-2/print-bgp.c, to overwrite the target buffer.

	Source	Destination
File	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	554	554
Object	plenbytes	plenbytes

Code Snippet
File Name minix-2/print-bgp.c
Method decode_labeled_prefix4(netdissect_options *ndo,

```
....
554.         memcpy(&addr, &pptr[4], plenbytes);
```

Buffer Overflow boundcpy WrongSizeParam\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=8
Status	New

The size of the buffer used by decode_prefix6 in plenbytes, at line 1064 of minix-2/print-bgp.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that decode_prefix6 passes to plenbytes, at line 1064 of minix-2/print-bgp.c, to overwrite the target buffer.

	Source	Destination
File	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	1081	1081

Object	plenbytes	plenbytes

Code Snippet
File Name minix-2/print-bgp.c
Method decode_prefix6(netdissect_options *ndo,

```
....
1081.       memcpy(&addr, &pd[1], plenbytes);
```

Buffer Overflow boundcpy WrongSizeParam\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=9
Status	New

The size of the buffer used by decode_labeled_prefix6 in plenbytes, at line 1097 of minix-2/print-bgp.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that decode_labeled_prefix6 passes to plenbytes, at line 1097 of minix-2/print-bgp.c, to overwrite the target buffer.

	Source	Destination
File	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	1120	1120
Object	plenbytes	plenbytes

Code Snippet
File Name minix-2/print-bgp.c
Method decode_labeled_prefix6(netdissect_options *ndo,

```
....
1120.       memcpy(&addr, &pptr[4], plenbytes);
```

Buffer Overflow boundcpy WrongSizeParam\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=10
Status	New

The size of the buffer used by get_server_hello in s, at line 334 of minix-2/s2_clnt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that get_server_hello passes to s, at line 334 of minix-2/s2_clnt.c, to overwrite the target buffer.

	Source	Destination
File	minix-2/s2_clnt.c	minix-2/s2_clnt.c
Line	531	531
Object	s	s

Code Snippet
File Name minix-2/s2_clnt.c
Method static int get_server_hello(SSL *s)

```
....
531.      memcpy(s->s2->conn_id, p, s->s2->tmp.conn_id_length);
```

Buffer Overflow boundcpy WrongSizeParam\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=11
Status	New

The size of the buffer used by client_hello in SSL2_CHALLENGE_LENGTH, at line 535 of minix-2/s2_clnt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that client_hello passes to SSL2_CHALLENGE_LENGTH, at line 535 of minix-2/s2_clnt.c, to overwrite the target buffer.

	Source	Destination
File	minix-2/s2_clnt.c	minix-2/s2_clnt.c
Line	586	586
Object	SSL2_CHALLENGE_LENGTH	SSL2_CHALLENGE_LENGTH

Code Snippet
File Name minix-2/s2_clnt.c
Method static int client_hello(SSL *s)

```
....
586.          memcpy(d, s->s2->challenge, SSL2_CHALLENGE_LENGTH);
```

Buffer Overflow boundcpy WrongSizeParam\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=12
Status	New

The size of the buffer used by get_server_finished in SSL2_SSL_SESSION_ID_LENGTH, at line 922 of minix-2/s2_clnt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that get_server_finished passes to SSL2_SSL_SESSION_ID_LENGTH, at line 922 of minix-2/s2_clnt.c, to overwrite the target buffer.

	Source	Destination
File	minix-2/s2_clnt.c	minix-2/s2_clnt.c
Line	980	980
Object	SSL2_SSL_SESSION_ID_LENGTH	SSL2_SSL_SESSION_ID_LENGTH

Code Snippet
File Name minix-2/s2_clnt.c
Method static int get_server_finished(SSL *s)

```
....
980.          memcpy(s->session->session_id, p + 1,
SSL2_SSL_SESSION_ID_LENGTH);
```

Buffer Overflow boundcpy WrongSizeParam\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=13
Status	New

The size of the buffer used by ssl3_send_client_key_exchange in enc_ticket, at line 2302 of minix-2/s3_clnt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ssl3_send_client_key_exchange passes to enc_ticket, at line 2302 of minix-2/s3_clnt.c, to overwrite the target buffer.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	2471	2471
Object	enc_ticket	enc_ticket

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_send_client_key_exchange(SSL *s)

```
....
2471.              memcpy(p, enc_ticket->data, enc_ticket->length);
```

Buffer Overflow boundcpy WrongSizeParam\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=14
Status	New

The size of the buffer used by ssl3_send_client_key_exchange in authp, at line 2302 of minix-2/s3_clnt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ssl3_send_client_key_exchange passes to authp, at line 2302 of minix-2/s3_clnt.c, to overwrite the target buffer.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	2478	2478
Object	authp	authp

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_send_client_key_exchange(SSL *s)

```
....
2478.                   memcpy(p, authp->data, authp->length);
```

Buffer Overflow boundcpy WrongSizeParam\Path 13:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=15
Status New

The size of the buffer used by ssl3_send_cert_status in s, at line 3520 of minix-2/s3_srvr.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ssl3_send_cert_status passes to s, at line 3520 of minix-2/s3_srvr.c, to overwrite the target buffer.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	3546	3546
Object	s	s

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_send_cert_status(SSL *s)

```
....
3546.           memcpy(p, s->tlsext_ocsp_resp, s->tlsext_ocsp_resplen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 14:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=16
Status New

The size of the buffer used by str_concat in len1, at line 95 of minix-2/str.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that str_concat passes to len1, at line 95 of minix-2/str.c, to overwrite the target buffer.

	Source	Destination
File	minix-2/str.c	minix-2/str.c
Line	108	108
Object	len1	len1

Code Snippet
File Name minix-2/str.c

Method	str_concat(const char *s1, const char *s2, int flags)

```
....
108.        memcpy(result, s1, len1);
```

Buffer Overflow boundcpy WrongSizeParam\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=17
Status	New

The size of the buffer used by mib_write in node, at line 1160 of minix-2/tree.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that mib_write passes to node, at line 1160 of minix-2/tree.c, to overwrite the target buffer.

	Source	Destination
File	minix-2/tree.c	minix-2/tree.c
Line	1271	1271
Object	node	node

Code Snippet	
File Name	minix-2/tree.c
Method	mib_write(struct mib_call * call, struct mib_node * node,

```
....
1271.                   memcpy(dst, src, node->node_size);
```

Buffer Overflow boundcpy WrongSizeParam\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=18
Status	New

The size of the buffer used by dns_rdata_towire in rdata, at line 565 of minix-2/rdata.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that dns_rdata_towire passes to rdata, at line 565 of minix-2/rdata.c, to overwrite the target buffer.

	Source	Destination
File	minix-2/rdata.c	minix-2/rdata.c
Line	592	592
Object	rdata	rdata

Code Snippet	
File Name	minix-2/rdata.c
Method	dns_rdata_towire(dns_rdata_t *rdata, dns_compress_t *cctx,

```
....
592.              memmove(tr.base, rdata->data, rdata->length);
```

Buffer Overflow boundcpy WrongSizeParam\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=19
Status	New

The size of the buffer used by str_totext in l, at line 1442 of minix-2/rdata.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that str_totext passes to l, at line 1442 of minix-2/rdata.c, to overwrite the target buffer.

	Source	Destination
File	minix-2/rdata.c	minix-2/rdata.c
Line	1452	1452
Object	l	l

Code Snippet
File Name minix-2/rdata.c
Method str_totext(const char *source, isc_buffer_t *target) {

```
....
1452.       memmove(region.base, source, l);
```

Buffer Overflow boundcpy WrongSizeParam\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=20
Status	New

The size of the buffer used by *createOurException in size, at line 359 of minix-2/ExceptionDemo.cpp, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *createOurException passes to size, at line 359 of minix-2/ExceptionDemo.cpp, to overwrite the target buffer.

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	361	361
Object	size	size

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method OurUnwindException *createOurException(int type) {

```
....
361.    OurException *ret = (OurException*) memset(malloc(size), 0,
size);
```

Buffer Overflow boundcpy WrongSizeParam\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=21
Status	New

The size of the buffer used by get_server_hello in SSL_MAX_SSL_SESSION_ID_LENGTH_IN_BYTES, at line 334 of minix-2/s2_clnt.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that get_server_hello passes to SSL_MAX_SSL_SESSION_ID_LENGTH_IN_BYTES, at line 334 of minix-2/s2_clnt.c, to overwrite the target buffer.

	Source	Destination
File	minix-2/s2_clnt.c	minix-2/s2_clnt.c
Line	424	424
Object	SSL_MAX_SSL_SESSION_ID_LENGTH_IN_BYTES	SSL_MAX_SSL_SESSION_ID_LENGTH_IN_BYTES

Code Snippet
File Name minix-2/s2_clnt.c
Method static int get_server_hello(SSL *s)

```
....
424.                SSL_MAX_SSL_SESSION_ID_LENGTH_IN_BYTES);
```

Buffer Overflow boundcpy WrongSizeParam\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=22
Status	New

The size of the buffer used by mib_create in scn, at line 487 of minix-2/tree.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that mib_create passes to scn, at line 487 of minix-2/tree.c, to overwrite the target buffer.

	Source	Destination
File	minix-2/tree.c	minix-2/tree.c
Line	738	738
Object	scn	scn

Code Snippet
File Name minix-2/tree.c

Method	mib_create(struct mib_call * call, struct mib_node * parent,

```
....
738.                    memset(node->node_data, 0, scn.sysctl_size);
```

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=1030030&projectid=30025&pathid=258
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	1733	1733
Object	ObjectCreateExpr	ObjectCreateExpr

Code Snippet	
File Name	minix-2/ExceptionDemo.cpp
Method	static void createStandardUtilityFunctions(unsigned numTypeInfos,

```
....
1733.       new llvm::GlobalVariable(module,
```

Memory Leak\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=259
Status	New

	Source	Destination
File	minix-2/rdata.c	minix-2/rdata.c
Line	319	319
Object	neW	neW

Code Snippet	
File Name	minix-2/rdata.c
Method	mem_maybedup(isc_mem_t *mctx, void *source, size_t length) {

```
....
319.         void *new;
```

Memory Leak\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=260
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	914	914
Object	stringVar	stringVar

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method void generateStringPrint(llvm::LLVMContext &context,

```
....
914.       stringVar =
```

Memory Leak\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=261
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	958	958
Object	stringVar	stringVar

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method void generateIntegerPrint(llvm::LLVMContext &context,

```
....
958.       stringVar =
```

Memory Leak\Path 5:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=262
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	361	361
Object	ret	ret

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method OurUnwindException *createOurException(int type) {

```
....
361.    OurException *ret = (OurException*) memset(malloc(size), 0,
size);
```

Memory Leak\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=263
Status	New

	Source	Destination
File	minix-2/archive_write_set_format_mtree.c	minix-2/archive_write_set_format_mtree.c
Line	1021	1021
Object	mtree	mtree

Code Snippet
File Name minix-2/archive_write_set_format_mtree.c
Method archive_write_set_format_mtree(struct archive *_a)

```
....
1021.       if ((mtree = malloc(sizeof(*mtree))) == NULL) {
```

Memory Leak\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=264
Status	New

	Source	Destination
File	minix-2/tree.c	minix-2/tree.c

Line	1043	1043
Object	node_desc	node_desc

Code Snippet
File Name minix-2/tree.c
Method mib_describe(struct mib_call * call, struct mib_node * parent,

```
....
1043.                   if ((node->node_desc = strdup(scratch)) == NULL)
{
```

Memory Leak\Path 8:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=265
Status New

	Source	Destination
File	minix-2/tree.c	minix-2/tree.c
Line	1741	1741
Object	dynode	dynode

Code Snippet
File Name minix-2/tree.c
Method mib_mount(const int * mib, unsigned int miblen, unsigned int eid, uint32_t rid,

```
....
1741.             if ((dynode = malloc(size)) == 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=1030030&projectid=30025&pathid=27
Status New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 334 of minix-2/s2_clnt.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	minix-2/s2_clnt.c	minix-2/s2_clnt.c
Line	391	391
Object	AssignExpr	AssignExpr

Code Snippet
File Name minix-2/s2_clnt.c
Method static int get_server_hello(SSL *s)

```
....
391.      j = (int)len - s->init_num;
```

Integer Overflow\Path 2:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=28
Status New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 2302 of minix-2/s3_clnt.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	2867	2867
Object	AssignExpr	AssignExpr

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_send_client_key_exchange(SSL *s)

```
....
2867.                  n = msglen + 3;
```

Integer Overflow\Path 3:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=29
Status New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 2302 of minix-2/s3_clnt.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination

File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	2870	2870
Object	AssignExpr	AssignExpr

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_send_client_key_exchange(SSL *s)

```
....
2870.                    n = msglen + 2;
```

Integer Overflow\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=30
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 2302 of minix-2/s3_clnt.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	3000	3000
Object	AssignExpr	AssignExpr

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_send_client_key_exchange(SSL *s)

```
....
3000.               n = 2 + identity_len;
```

Integer Overflow\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=31
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 2182 of minix-2/s3_srvr.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	2373	2373
Object	AssignExpr	AssignExpr

Code Snippet	
File Name	minix-2/s3_srvr.c
Method	int ssl3_get_client_key_exchange(SSL *s)

```
....
2373.                   i = (int)n;
```

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=1030030&projectid=30025&pathid=194
Status	New

Calling free() (line 891) on a variable that was not dynamically allocated (line 891) in file minix-2/archive_write_set_format_mtree.c may result with a crash.

	Source	Destination
File	minix-2/archive_write_set_format_mtree.c	minix-2/archive_write_set_format_mtree.c
Line	902	902
Object	mtree	mtree

Code Snippet	
File Name	minix-2/archive_write_set_format_mtree.c
Method	archive_write_mtree_destroy(struct archive_write *a)

```
....
902.        free(mtree);
```

MemoryFree on StackVariable\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=195
Status	New

Calling free() (line 141) on a variable that was not dynamically allocated (line 141) in file minix-2/str.c may result with a crash.

	Source	Destination
File	minix-2/str.c	minix-2/str.c
Line	226	226

Object	argv	argv

Code Snippet
File Name minix-2/str.c
Method brk_string(const char *str, int *store_argc, Boolean expand, char **buffer)

```
....
226.                                    free(argv);
```

MemoryFree on StackVariable\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=196
Status	New

Calling free() (line 164) on a variable that was not dynamically allocated (line 164) in file minix-2/test_pac.c may result with a crash.

	Source	Destination
File	minix-2/test_pac.c	minix-2/test_pac.c
Line	246	246
Object	list	list

Code Snippet
File Name minix-2/test_pac.c
Method main(int argc, char **argv)

```
....
246.        free(list);
```

MemoryFree on StackVariable\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=197
Status	New

Calling free() (line 164) on a variable that was not dynamically allocated (line 164) in file minix-2/test_pac.c may result with a crash.

	Source	Destination
File	minix-2/test_pac.c	minix-2/test_pac.c
Line	374	374
Object	list	list

Code Snippet
File Name minix-2/test_pac.c
Method main(int argc, char **argv)

```
....
374.        free(list);
```

MemoryFree on StackVariable\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=198
Status	New

Calling free() (line 785) on a variable that was not dynamically allocated (line 785) in file minix-2/tree.c may result with a crash.

	Source	Destination
File	minix-2/tree.c	minix-2/tree.c
Line	820	820
Object	dynode	dynode

Code Snippet
File Name minix-2/tree.c
Method mib_remove(struct mib_node * node, struct mib_dynode ** prevp)

```
....
820.              free(dynode);
```

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=1030030&projectid=30025&pathid=23
Status	New

The function size in minix-2/ExceptionDemo.cpp at line 359 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	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	361	361
Object	size	size

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method OurUnwindException *createOurException(int type) {

```
....
361.    OurException *ret = (OurException*) memset(malloc(size), 0,
size);
```

Wrong Size t Allocation\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=24
Status	New

The function size in minix-2/tree.c at line 487 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	minix-2/tree.c	minix-2/tree.c
Line	683	683
Object	size	size

Code Snippet
File Name minix-2/tree.c
Method mib_create(struct mib_call * call, struct mib_node * parent,

```
....
683.        if ((dynode = malloc(size)) == NULL)
```

Wrong Size t Allocation\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=25
Status	New

The function size in minix-2/tree.c at line 1544 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	minix-2/tree.c	minix-2/tree.c
Line	1741	1741
Object	size	size

Code Snippet

File Name	minix-2/tree.c
Method	mib_mount(const int * mib, unsigned int miblen, unsigned int eid, uint32_t rid,

```
....
1741.             if ((dynode = malloc(size)) == NULL) {
```

Wrong Size t Allocation\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=26
Status	New

The function newlen in minix-2/tree.c at line 1160 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	minix-2/tree.c	minix-2/tree.c
Line	1235	1235
Object	newlen	newlen

Code Snippet

File Name	minix-2/tree.c
Method	mib_write(struct mib_call * call, struct mib_node * node,

```
....
1235.             if ((src = malloc(newlen + 1)) == 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=1030030&projectid=30025&pathid=266
Status	New

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	2878	2904
Object	Ttag	Ttag

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_client_key_exchange(SSL *s)

```
....
2878.          int Ttag, Tclass;
....
2904.               n) != V_ASN1_CONSTRUCTED || Ttag != V_ASN1_SEQUENCE
```

Use of Uninitialized Variable\Path 2:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=267
Status New

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	2878	2905
Object	Tclass	Tclass

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_client_key_exchange(SSL *s)

```
....
2878.          int Ttag, Tclass;
....
2905.               || Tclass != V_ASN1_UNIVERSAL) {
```

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=1030030&projectid=30025&pathid=145
Status New

Method SRP_VBASE_init at line 358 of minix-2/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	minix-2/srp_vfy.c	minix-2/srp_vfy.c
Line	367	367
Object	user_pwd	user_pwd

Code Snippet
File Name minix-2/srp_vfy.c
Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
367.      SRP_user_pwd *user_pwd = NULL;
```

NULL Pointer Dereference

Query Path:
CPP\Cx\CPP Low Visibility\NULL Pointer Dereference Version:1

Categories

NIST SP 800-53: SC-5 Denial of Service Protection (P1)
OWASP Top 10 2017: A1-Injection

Description
NULL Pointer Dereference\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=199
Status	New

The variable declared in null at minix-2/s3_clnt.c in line 838 is not initialized when it is used by cipher at minix-2/s3_clnt.c in line 838.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	917	923
Object	null	cipher

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_hello(SSL *s)

```
....
917.          SSL_CIPHER *pref_cipher = NULL;
....
923.              s->session->cipher = pref_cipher ?
```

NULL Pointer Dereference\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=200
Status	New

The variable declared in null at minix-2/s3_clnt.c in line 838 is not initialized when it is used by session at minix-2/s3_clnt.c in line 838.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	917	1037
Object	null	session

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_hello(SSL *s)

```
....
917.          SSL_CIPHER *pref_cipher = NULL;
....
1037.      if (s->session->compress_meth != 0) {
```

NULL Pointer Dereference\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=201
Status	New

The variable declared in null at minix-2/s3_clnt.c in line 838 is not initialized when it is used by session at minix-2/s3_clnt.c in line 838.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	917	1002
Object	null	session

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_hello(SSL *s)

```
....
917.          SSL_CIPHER *pref_cipher = NULL;
....
1002.      if (s->hit && (s->session->cipher_id != c->id)) {
```

NULL Pointer Dereference\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=202
Status	New

The variable declared in null at minix-2/s3_clnt.c in line 838 is not initialized when it is used by cipher at minix-2/s3_clnt.c in line 838.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	917	1001
Object	null	cipher

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_hello(SSL *s)

```
....
917.          SSL_CIPHER *pref_cipher = NULL;
....
1001.          s->session->cipher_id = s->session->cipher->id;
```

NULL Pointer Dereference\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=203
Status	New

The variable declared in null at minix-2/s3_clnt.c in line 838 is not initialized when it is used by session at minix-2/s3_clnt.c in line 838.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	917	1001
Object	null	session

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_hello(SSL *s)

```
....
917.          SSL_CIPHER *pref_cipher = NULL;
....
1001.          s->session->cipher_id = s->session->cipher->id;
```

NULL Pointer Dereference\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=204
Status	New

The variable declared in null at minix-2/s3_clnt.c in line 838 is not initialized when it is used by session at minix-2/s3_clnt.c in line 838.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	917	1000
Object	null	session

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_hello(SSL *s)

```
....
917.          SSL_CIPHER *pref_cipher = NULL;
....
1000.      if (s->session->cipher)
```

NULL Pointer Dereference\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=205
Status	New

The variable declared in null at minix-2/s3_clnt.c in line 838 is not initialized when it is used by session at minix-2/s3_clnt.c in line 838.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	917	935
Object	null	session

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_hello(SSL *s)

```
....
917.          SSL_CIPHER *pref_cipher = NULL;
....
935.          if (s->sid_ctx_length != s->session->sid_ctx_length
```

NULL Pointer Dereference\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=206
Status	New

The variable declared in null at minix-2/s3_clnt.c in line 838 is not initialized when it is used by session at minix-2/s3_clnt.c in line 838.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	917	936
Object	null	session

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_hello(SSL *s)

```
....
917.          SSL_CIPHER *pref_cipher = NULL;
....
936.              || memcmp(s->session->sid_ctx, s->sid_ctx, s-
>sid_ctx_length)) {
```

NULL Pointer Dereference\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=207
Status	New

The variable declared in null at minix-2/s3_clnt.c in line 838 is not initialized when it is used by session at minix-2/s3_clnt.c in line 838.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	917	959
Object	null	session

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_hello(SSL *s)

```
....
917.          SSL_CIPHER *pref_cipher = NULL;
....
959.          memcpy(s->session->session_id, p, j); /* j could be 0 */
```

NULL Pointer Dereference\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=208
Status	New

The variable declared in null at minix-2/s3_clnt.c in line 838 is not initialized when it is used by session at minix-2/s3_clnt.c in line 838.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	917	952
Object	null	session

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_hello(SSL *s)

```
....
917.          SSL_CIPHER *pref_cipher = NULL;
....
952.          if (s->session->session_id_length > 0) {
```

NULL Pointer Dereference\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=209
Status	New

The variable declared in null at minix-2/s3_clnt.c in line 838 is not initialized when it is used by session at minix-2/s3_clnt.c in line 838.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	917	933
Object	null	session

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_hello(SSL *s)

```
....
917.          SSL_CIPHER *pref_cipher = NULL;
....
933.      if (j != 0 && j == s->session->session_id_length
```

NULL Pointer Dereference\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=210
Status	New

The variable declared in null at minix-2/s3_clnt.c in line 838 is not initialized when it is used by session at minix-2/s3_clnt.c in line 838.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	917	934
Object	null	session

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_hello(SSL *s)

```
....
917.          SSL_CIPHER *pref_cipher = NULL;
....
934.          && memcmp(p, s->session->session_id, j) == 0) {
```

NULL Pointer Dereference\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=211
Status	New

The variable declared in null at minix-2/s3_clnt.c in line 1099 is not initialized when it is used by references at minix-2/s3_clnt.c in line 1099.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	1235	1256
Object	null	references

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_certificate(SSL *s)

```
....
1235.          x = NULL;
....
1256.          CRYPTO_add(&x->references, 1, CRYPTO_LOCK_X509);
```

NULL Pointer Dereference\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=212
Status	New

The variable declared in null at minix-2/s3_clnt.c in line 1099 is not initialized when it is used by references at minix-2/s3_clnt.c in line 1099.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	1235	1244
Object	null	references

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_certificate(SSL *s)

```
....
1235.           x = NULL;
....
1244.           CRYPTO_add(&x->references, 1, CRYPTO_LOCK_X509);
```

NULL Pointer Dereference\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=213
Status	New

The variable declared in null at minix-2/s3_srvr.c in line 1639 is not initialized when it is used by pkey at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1926	2009
Object	null	pkey

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1926.               pkey = NULL;
....
2009.                                       &(p[2]), &u, pkey->pkey.rsa) <= 0) {
```

NULL Pointer Dereference\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=214
Status	New

The variable declared in null at minix-2/srp_vfy.c in line 358 is not initialized when it is used by info at minix-2/srp_vfy.c in line 188.

	Source	Destination
File	minix-2/srp_vfy.c	minix-2/srp_vfy.c
Line	429	195
Object	null	info

Code Snippet
File Name minix-2/srp_vfy.c
Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
429.                    user_pwd = NULL; /* abandon responsability */
```

File Name minix-2/srp_vfy.c
Method static void SRP_user_pwd_free(SRP_user_pwd *user_pwd)

```
....
195.        OPENSSL_free(user_pwd->info);
```

NULL Pointer Dereference\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=215
Status	New

The variable declared in null at minix-2/srp_vfy.c in line 358 is not initialized when it is used by info at minix-2/srp_vfy.c in line 188.

	Source	Destination
File	minix-2/srp_vfy.c	minix-2/srp_vfy.c
Line	367	195
Object	null	info

Code Snippet
File Name minix-2/srp_vfy.c

Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
367.      SRP_user_pwd *user_pwd = NULL;
```

File Name minix-2/srp_vfy.c

Method static void SRP_user_pwd_free(SRP_user_pwd *user_pwd)

```
....
195.      OPENSSL_free(user_pwd->info);
```

NULL Pointer Dereference\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=216
Status	New

The variable declared in null at minix-2/srp_vfy.c in line 358 is not initialized when it is used by id at minix-2/srp_vfy.c in line 188.

	Source	Destination
File	minix-2/srp_vfy.c	minix-2/srp_vfy.c
Line	429	194
Object	null	id

Code Snippet

File Name minix-2/srp_vfy.c

Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
429.                  user_pwd = NULL; /* abandon responsability */
```

File Name minix-2/srp_vfy.c

Method static void SRP_user_pwd_free(SRP_user_pwd *user_pwd)

```
....
194.      OPENSSL_free(user_pwd->id);
```

NULL Pointer Dereference\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=217
Status	New

The variable declared in null at minix-2/srp_vfy.c in line 358 is not initialized when it is used by id at minix-2/srp_vfy.c in line 188.

	Source	Destination
File	minix-2/srp_vfy.c	minix-2/srp_vfy.c
Line	367	194
Object	null	id

Code Snippet
File Name minix-2/srp_vfy.c
Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
367.      SRP_user_pwd *user_pwd = NULL;
```

File Name minix-2/srp_vfy.c
Method static void SRP_user_pwd_free(SRP_user_pwd *user_pwd)

```
....
194.      OPENSSL_free(user_pwd->id);
```

NULL Pointer Dereference\Path 20:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=218
Status New

The variable declared in null at minix-2/srp_vfy.c in line 358 is not initialized when it is used by v at minix-2/srp_vfy.c in line 188.

	Source	Destination
File	minix-2/srp_vfy.c	minix-2/srp_vfy.c
Line	429	193
Object	null	v

Code Snippet
File Name minix-2/srp_vfy.c
Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
429.                  user_pwd = NULL; /* abandon responsability */
```

File Name minix-2/srp_vfy.c

Method static void SRP_user_pwd_free(SRP_user_pwd *user_pwd)

```
....
193.      BN_clear_free(user_pwd->v);
```

NULL Pointer Dereference\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=219
Status	New

The variable declared in null at minix-2/srp_vfy.c in line 358 is not initialized when it is used by v at minix-2/srp_vfy.c in line 188.

	Source	Destination
File	minix-2/srp_vfy.c	minix-2/srp_vfy.c
Line	367	193
Object	null	v

Code Snippet
File Name minix-2/srp_vfy.c
Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
367.      SRP_user_pwd *user_pwd = NULL;
```

File Name minix-2/srp_vfy.c

Method static void SRP_user_pwd_free(SRP_user_pwd *user_pwd)

```
....
193.      BN_clear_free(user_pwd->v);
```

NULL Pointer Dereference\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=220
Status	New

The variable declared in null at minix-2/srp_vfy.c in line 471 is not initialized when it is used by v at minix-2/srp_vfy.c in line 188.

	Source	Destination
File	minix-2/srp_vfy.c	minix-2/srp_vfy.c

Line	509	193
Object	null	v

Code Snippet
File Name minix-2/srp_vfy.c
Method SRP_user_pwd *SRP_VBASE_get_by_user(SRP_VBASE *vb, char *username)

```
....
509.          (user, BN_bin2bn(digs, SHA_DIGEST_LENGTH, NULL),
```

File Name minix-2/srp_vfy.c

Method static void SRP_user_pwd_free(SRP_user_pwd *user_pwd)

```
....
193.      BN_clear_free(user_pwd->v);
```

NULL Pointer Dereference\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=221
Status	New

The variable declared in null at minix-2/srp_vfy.c in line 358 is not initialized when it is used by s at minix-2/srp_vfy.c in line 188.

	Source	Destination
File	minix-2/srp_vfy.c	minix-2/srp_vfy.c
Line	429	192
Object	null	s

Code Snippet
File Name minix-2/srp_vfy.c
Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
429.                  user_pwd = NULL; /* abandon responsability */
```

File Name minix-2/srp_vfy.c

Method static void SRP_user_pwd_free(SRP_user_pwd *user_pwd)

```
....
192.      BN_free(user_pwd->s);
```

NULL Pointer Dereference\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=222
Status	New

The variable declared in null at minix-2/srp_vfy.c in line 358 is not initialized when it is used by s at minix-2/srp_vfy.c in line 188.

	Source	Destination
File	minix-2/srp_vfy.c	minix-2/srp_vfy.c
Line	367	192
Object	null	s

Code Snippet

File Name minix-2/srp_vfy.c

Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
367.      SRP_user_pwd *user_pwd = NULL;
```

File Name minix-2/srp_vfy.c

Method static void SRP_user_pwd_free(SRP_user_pwd *user_pwd)

```
....
192.      BN_free(user_pwd->s);
```

NULL Pointer Dereference\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=223
Status	New

The variable declared in null at minix-2/srp_vfy.c in line 471 is not initialized when it is used by s at minix-2/srp_vfy.c in line 188.

	Source	Destination
File	minix-2/srp_vfy.c	minix-2/srp_vfy.c
Line	510	192
Object	null	s

Code Snippet

File Name minix-2/srp_vfy.c

Method SRP_user_pwd *SRP_VBASE_get_by_user(SRP_VBASE *vb, char *username)

```
....
510.            BN_bin2bn(digv, SHA_DIGEST_LENGTH, NULL)))
```

File Name minix-2/srp_vfy.c

Method static void SRP_user_pwd_free(SRP_user_pwd *user_pwd)

```
....
192.        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=1030030&projectid=30025&pathid=224
Status	New

The variable declared in null at minix-2/srp_vfy.c in line 471 is not initialized when it is used by s at minix-2/srp_vfy.c in line 243.

	Source	Destination
File	minix-2/srp_vfy.c	minix-2/srp_vfy.c
Line	509	247
Object	null	s

Code Snippet

File Name minix-2/srp_vfy.c

Method SRP_user_pwd *SRP_VBASE_get_by_user(SRP_VBASE *vb, char *username)

```
....
509.            (user, BN_bin2bn(digs, SHA_DIGEST_LENGTH, NULL),
```

File Name minix-2/srp_vfy.c

Method static int SRP_user_pwd_set_sv_BN(SRP_user_pwd *vinfo, BIGNUM *s, BIGNUM *v)

```
....
247.        return (vinfo->s != NULL && vinfo->v != NULL);
```

NULL Pointer Dereference\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=225
Status	New

The variable declared in null at minix-2/srp_vfy.c in line 471 is not initialized when it is used by v at minix-2/srp_vfy.c in line 243.

	Source	Destination
File	minix-2/srp_vfy.c	minix-2/srp_vfy.c
Line	510	247
Object	null	v

Code Snippet
File Name minix-2/srp_vfy.c
Method SRP_user_pwd *SRP_VBASE_get_by_user(SRP_VBASE *vb, char *username)

```
....
510.            BN_bin2bn(digv, SHA_DIGEST_LENGTH, NULL)))
```

File Name minix-2/srp_vfy.c
Method static int SRP_user_pwd_set_sv_BN(SRP_user_pwd *vinfo, BIGNUM *s, BIGNUM *v)

```
....
247.      return (vinfo->s != NULL && vinfo->v != NULL);
```

NULL Pointer Dereference\Path 28:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=226
Status New

The variable declared in 0 at minix-2/archive_write_set_format_mtree.c in line 908 is not initialized when it is used by dironly at minix-2/archive_write_set_format_mtree.c in line 908.

	Source	Destination
File	minix-2/archive_write_set_format_mtree.c	minix-2/archive_write_set_format_mtree.c
Line	927	927
Object	0	dironly

Code Snippet
File Name minix-2/archive_write_set_format_mtree.c
Method archive_write_mtree_options(struct archive_write *a, const char *key,

```
....
927.                    mtree->dironly = (value != NULL)? 1: 0;
```

NULL Pointer Dereference\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=227
Status	New

The variable declared in 0 at minix-2/archive_write_set_format_mtree.c in line 908 is not initialized when it is used by indent at minix-2/archive_write_set_format_mtree.c in line 908.

	Source	Destination
File	minix-2/archive_write_set_format_mtree.c	minix-2/archive_write_set_format_mtree.c
Line	943	943
Object	0	indent

Code Snippet
File Name minix-2/archive_write_set_format_mtree.c
Method archive_write_mtree_options(struct archive_write *a, const char *key,

```
....
943.                    mtree->indent = (value != NULL)? 1: 0;
```

NULL Pointer Dereference\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=228
Status	New

The variable declared in 0 at minix-2/archive_write_set_format_mtree.c in line 908 is not initialized when it is used by output at minix-2/archive_write_set_format_mtree.c in line 908.

	Source	Destination
File	minix-2/archive_write_set_format_mtree.c	minix-2/archive_write_set_format_mtree.c
Line	996	996
Object	0	output

Code Snippet
File Name minix-2/archive_write_set_format_mtree.c
Method archive_write_mtree_options(struct archive_write *a, const char *key,

```
....
996.                    mtree->set.output = (value != NULL)? 1: 0;
```

NULL Pointer Dereference\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=229
Status	New

The variable declared in 0 at minix-2/ExceptionDemo.cpp in line 452 is not initialized when it is used by result at minix-2/ExceptionDemo.cpp in line 452.

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	453	522
Object	0	result

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method static uintptr_t readEncodedPointer(const uint8_t **data, uint8_t encoding) {

```
....
453.    uintptr_t result = 0;
....
522.      result = *((uintptr_t*)result);
```

NULL Pointer Dereference\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=230
Status	New

The variable declared in sc at minix-2/s2_clnt.c in line 1000 is not initialized when it is used by peer_key at minix-2/s2_clnt.c in line 1000.

	Source	Destination
File	minix-2/s2_clnt.c	minix-2/s2_clnt.c
Line	1004	1040
Object	sc	peer_key

Code Snippet
File Name minix-2/s2_clnt.c
Method int ssl2_set_certificate(SSL *s, int type, int len, const unsigned char *data)

```
....
1004.      SESS_CERT *sc = NULL;
....
1040.      sc->peer_key = &(sc->peer_pkeys[SSL_PKEY_RSA_ENC]);
```

NULL Pointer Dereference\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=231
Status	New

The variable declared in sc at minix-2/s2_clnt.c in line 1000 is not initialized when it is used by x509 at minix-2/s2_clnt.c in line 1000.

	Source	Destination
File	minix-2/s2_clnt.c	minix-2/s2_clnt.c
Line	1004	1039
Object	sc	x509

Code Snippet
File Name minix-2/s2_clnt.c
Method int ssl2_set_certificate(SSL *s, int type, int len, const unsigned char *data)

```
....
1004.       SESS_CERT *sc = NULL;
....
1039.       sc->peer_pkeys[SSL_PKEY_RSA_ENC].x509 = x509;
```

NULL Pointer Dereference\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=232
Status	New

The variable declared in sc at minix-2/s2_clnt.c in line 1000 is not initialized when it is used by peer_pkeys at minix-2/s2_clnt.c in line 1000.

	Source	Destination
File	minix-2/s2_clnt.c	minix-2/s2_clnt.c
Line	1004	1040
Object	sc	peer_pkeys

Code Snippet
File Name minix-2/s2_clnt.c
Method int ssl2_set_certificate(SSL *s, int type, int len, const unsigned char *data)

```
....
1004.       SESS_CERT *sc = NULL;
....
1040.       sc->peer_key = &(sc->peer_pkeys[SSL_PKEY_RSA_ENC]);
```

NULL Pointer Dereference\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=233
Status	New

The variable declared in pkey at minix-2/s3_clnt.c in line 2302 is not initialized when it is used by pkey at minix-2/s3_clnt.c in line 2302.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	2309	2354
Object	pkey	pkey

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_send_client_key_exchange(SSL *s)

```
....
2309.       EVP_PKEY *pkey = NULL;
....
2354.                       || (pkey->pkey.rsa == NULL)) {
```

NULL Pointer Dereference\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=234
Status	New

The variable declared in pkey at minix-2/s3_clnt.c in line 2302 is not initialized when it is used by pkey at minix-2/s3_clnt.c in line 2302.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	2309	2359
Object	pkey	pkey

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_send_client_key_exchange(SSL *s)

```
....
2309.       EVP_PKEY *pkey = NULL;
....
2359.                   rsa = pkey->pkey.rsa;
```

NULL Pointer Dereference\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=235
Status	New

The variable declared in pkey at minix-2/s3_clnt.c in line 2302 is not initialized when it is used by type at minix-2/s3_clnt.c in line 2302.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	2309	2353
Object	pkey	type

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_send_client_key_exchange(SSL *s)

```
....
2309.      EVP_PKEY *pkey = NULL;
....
2353.                  if ((pkey == NULL) || (pkey->type !=
EVP_PKEY_RSA)
```

NULL Pointer Dereference\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=236
Status	New

The variable declared in srvr_pub_pkey at minix-2/s3_clnt.c in line 2302 is not initialized when it is used by pkey at minix-2/s3_clnt.c in line 2302.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	2317	2648
Object	srvr_pub_pkey	pkey

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_send_client_key_exchange(SSL *s)

```
....
2317.      EVP_PKEY *srvr_pub_pkey = NULL;
....
2648.                      || (srvr_pub_pkey->pkey.ec == NULL)) {
```

NULL Pointer Dereference\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=237
Status	New

The variable declared in srvr_pub_pkey at minix-2/s3_clnt.c in line 2302 is not initialized when it is used by pkey at minix-2/s3_clnt.c in line 2302.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	2317	2654
Object	srvr_pub_pkey	pkey

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_send_client_key_exchange(SSL *s)

```
....
2317.       EVP_PKEY *srvr_pub_pkey = NULL;
....
2654.                   tkey = srvr_pub_pkey->pkey.ec;
```

NULL Pointer Dereference\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=238
Status	New

The variable declared in srvr_pub_pkey at minix-2/s3_clnt.c in line 2302 is not initialized when it is used by type at minix-2/s3_clnt.c in line 2302.

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	2317	2647
Object	srvr_pub_pkey	type

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_send_client_key_exchange(SSL *s)

```
....
2317.       EVP_PKEY *srvr_pub_pkey = NULL;
....
2647.                       || (srvr_pub_pkey->type != EVP_PKEY_EC)
```

NULL Pointer Dereference\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=239
Status	New

The variable declared in dh at minix-2/s3_srvr.c in line 1639 is not initialized when it is used by pub_key at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1649	1744
Object	dh	pub_key

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1649.       DH *dh = NULL, *dhp;
....
1744.                   dh->pub_key = BN_dup(dhp->pub_key);
```

NULL Pointer Dereference\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=240
Status	New

The variable declared in dh at minix-2/s3_srvr.c in line 1639 is not initialized when it is used by priv_key at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1649	1745
Object	dh	priv_key

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1649.       DH *dh = NULL, *dhp;
....
1745.                   dh->priv_key = BN_dup(dhp->priv_key);
```

NULL Pointer Dereference\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=241
Status	New

The variable declared in dh at minix-2/s3_srvr.c in line 1639 is not initialized when it is used by g at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1649	1752
Object	dh	g

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1649.       DH *dh = NULL, *dhp;
....
1752.               r[1] = dh->g;
```

NULL Pointer Dereference\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=242
Status	New

The variable declared in dh at minix-2/s3_srvr.c in line 1639 is not initialized when it is used by p at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1649	1751
Object	dh	p

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1649.       DH *dh = NULL, *dhp;
....
1751.               r[0] = dh->p;
```

NULL Pointer Dereference\Path 45:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=243
Status	New

The variable declared in dh at minix-2/s3_srvr.c in line 1639 is not initialized when it is used by priv_key at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1649	1746
Object	dh	priv_key

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1649.       DH *dh = NULL, *dhp;
....
1746.                   if ((dh->pub_key == NULL) || (dh->priv_key ==
NULL)) {
```

NULL Pointer Dereference\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=244
Status	New

The variable declared in dh at minix-2/s3_srvr.c in line 1639 is not initialized when it is used by pub_key at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1649	1753
Object	dh	pub_key

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1649.       DH *dh = NULL, *dhp;
....
1753.               r[2] = dh->pub_key;
```

NULL Pointer Dereference\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=245
Status	New

The variable declared in dh at minix-2/s3_srvr.c in line 1639 is not initialized when it is used by pub_key at minix-2/s3_srvr.c in line 1639.

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1649	1746
Object	dh	pub_key

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_send_server_key_exchange(SSL *s)

```
....
1649.       DH *dh = NULL, *dhp;
....
1746.                   if ((dh->pub_key == NULL) || (dh->priv_key ==
NULL)) {
```

NULL Pointer Dereference\Path 48:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=246
Status	New

The variable declared in gN at minix-2/srp_vfy.c in line 358 is not initialized when it is used by N at minix-2/srp_vfy.c in line 358.

	Source	Destination
File	minix-2/srp_vfy.c	minix-2/srp_vfy.c
Line	366	398
Object	gN	N

Code Snippet
File Name minix-2/srp_vfy.c
Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
366.      SRP_gN *gN = NULL;
....
398.                    || !(gN->N =
```

NULL Pointer Dereference\Path 49:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=247
Status	New

The variable declared in gN at minix-2/srp_vfy.c in line 358 is not initialized when it is used by id at minix-2/srp_vfy.c in line 358.

	Source	Destination
File	minix-2/srp_vfy.c	minix-2/srp_vfy.c
Line	366	397
Object	gN	id

Code Snippet
File Name minix-2/srp_vfy.c
Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
366.      SRP_gN *gN = NULL;
....
397.              if (!(gN->id = BUF_strdup(pp[DB_srpid]))
```

NULL Pointer Dereference\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=248
Status	New

The variable declared in gN at minix-2/srp_vfy.c in line 358 is not initialized when it is used by g at minix-2/srp_vfy.c in line 358.

	Source	Destination
File	minix-2/srp_vfy.c	minix-2/srp_vfy.c
Line	366	400
Object	gN	g

Code Snippet
File Name minix-2/srp_vfy.c
Method int SRP_VBASE_init(SRP_VBASE *vb, char *verifier_file)

```
....
366.      SRP_gN *gN = NULL;
....
400.                  || !(gN->g = SRP_gN_place_bn(vb->gN_cache,
pp[DB_srpsalt]))
```

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=1030030&projectid=30025&pathid=413
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	1938	1938
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method int main(int argc, char *argv[]) {

```
....
1938.      fprintf(stderr,
```

Improper Resource Access Authorization\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=414
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	2017	2017
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method int main(int argc, char *argv[]) {

```
....
2017.       fprintf(stderr, "\nBegin module dump:\n\n");
```

Improper Resource Access Authorization\Path 3:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=415
Status New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	2021	2021
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method int main(int argc, char *argv[]) {

```
....
2021.       fprintf(stderr, "\nEnd module dump:\n");
```

Improper Resource Access Authorization\Path 4:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=416
Status New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	2023	2023
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method int main(int argc, char *argv[]) {

```
....
2023.       fprintf(stderr, "\n\nBegin Test:\n");
```

Improper Resource Access Authorization\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=417
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	2033	2033
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method int main(int argc, char *argv[]) {

```
....
2033.       fprintf(stderr, "\nEnd Test:\n\n");
```

Improper Resource Access Authorization\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=418
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	283	283
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method void print32Int(int intToPrint, const char *format) {

```
....
283.       fprintf(stderr, format, intToPrint);
```

Improper Resource Access Authorization\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=419
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	287	287
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method void print32Int(int intToPrint, const char *format) {

```
....
287.      fprintf(stderr, "::print32Int(...):NULL arg.\n");
```

Improper Resource Access Authorization\Path 8:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=420
Status New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	300	300
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method void print64Int(long int intToPrint, const char *format) {

```
....
300.      fprintf(stderr, format, intToPrint);
```

Improper Resource Access Authorization\Path 9:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=421
Status New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	304	304
Object	fprintf	fprintf

Code Snippet

File Name	minix-2/ExceptionDemo.cpp
Method	void print64Int(long int intToPrint, const char *format) {

```
....
304.      fprintf(stderr, "::print64Int(...):NULL arg.\n");
```

Improper Resource Access Authorization\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=422
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	313	313
Object	fprintf	fprintf

Code Snippet

File Name	minix-2/ExceptionDemo.cpp
Method	void printStr(char *toPrint) {

```
....
313.      fprintf(stderr, "%s", toPrint);
```

Improper Resource Access Authorization\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=423
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	316	316
Object	fprintf	fprintf

Code Snippet

File Name	minix-2/ExceptionDemo.cpp
Method	void printStr(char *toPrint) {

```
....
316.      fprintf(stderr, "::printStr(...):NULL arg.\n");
```

Improper Resource Access Authorization\Path 12:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=424
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	327	327
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method void deleteOurException(OurUnwindException *expToDelete) {

```
....
327.    fprintf(stderr,
```

Improper Resource Access Authorization\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=425
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	348	348
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method void deleteFromUnwindOurException(_Unwind_Reason_Code reason,

```
....
348.    fprintf(stderr,
```

Improper Resource Access Authorization\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=426
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp

Line	568	568
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method static bool handleActionValue(int64_t *resultAction,

```
....
568.    fprintf(stderr,
```

Improper Resource Access Authorization\Path 15:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=427
Status New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	589	589
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method static bool handleActionValue(int64_t *resultAction,

```
....
589.        fprintf(stderr,
```

Improper Resource Access Authorization\Path 16:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=428
Status New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	602	602
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method static bool handleActionValue(int64_t *resultAction,

```
....
602.        fprintf(stderr,
```

Improper Resource Access Authorization\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=429
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	619	619
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method static bool handleActionValue(int64_t *resultAction,

```
....
619.      fprintf(stderr,
```

Improper Resource Access Authorization\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=430
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	655	655
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method static _Unwind_Reason_Code handleLsda(int version,

```
....
655.    fprintf(stderr,
```

Improper Resource Access Authorization\Path 19:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=431
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	720	720
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method static _Unwind_Reason_Code handleLsda(int version,

```
....
720.        fprintf(stderr,
```

Improper Resource Access Authorization\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=432
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	732	732
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method static _Unwind_Reason_Code handleLsda(int version,

```
....
732.        fprintf(stderr,
```

Improper Resource Access Authorization\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=433
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	741	741

Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method static _Unwind_Reason_Code handleLsda(int version,

```
....
741.        fprintf(stderr,
```

Improper Resource Access Authorization\Path 22:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=434
Status New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	757	757
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method static _Unwind_Reason_Code handleLsda(int version,

```
....
757.          fprintf(stderr,
```

Improper Resource Access Authorization\Path 23:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=435
Status New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	792	792
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method static _Unwind_Reason_Code handleLsda(int version,

```
....
792.          fprintf(stderr,
```

Improper Resource Access Authorization\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=436
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	803	803
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method static _Unwind_Reason_Code handleLsda(int version,

```
....
803.          fprintf(stderr,
```

Improper Resource Access Authorization\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=437
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	833	833
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method _Unwind_Reason_Code ourPersonality(int version,

```
....
833.    fprintf(stderr,
```

Improper Resource Access Authorization\Path 26:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=438
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	838	838
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method _Unwind_Reason_Code ourPersonality(int version,

```
....
838.      fprintf(stderr, "ourPersonality(...):In search phase.\n");
```

Improper Resource Access Authorization\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=439
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	841	841
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method _Unwind_Reason_Code ourPersonality(int version,

```
....
841.      fprintf(stderr, "ourPersonality(...):In non-search phase.\n");
```

Improper Resource Access Authorization\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=440
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	848	848

Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method _Unwind_Reason_Code ourPersonality(int version,

```
....
848.    fprintf(stderr,
```

Improper Resource Access Authorization\Path 29:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=441
Status New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	1619	1619
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method void runExceptionThrow(llvm::ExecutionEngine *engine,

```
....
1619.       fprintf(stderr,
```

Improper Resource Access Authorization\Path 30:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=442
Status New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	1631	1631
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method void runExceptionThrow(llvm::ExecutionEngine *engine,

```
....
1631.       fprintf(stderr,
```

Improper Resource Access Authorization\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=443
Status	New

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	1698	1698
Object	fprintf	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method static void createStandardUtilityFunctions(unsigned numTypeInfos,

```
....
1698.     fprintf(stderr,
```

Improper Resource Access Authorization\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=444
Status	New

	Source	Destination
File	minix-2/rdata.c	minix-2/rdata.c
Line	1893	1893
Object	fprintf	fprintf

Code Snippet
File Name minix-2/rdata.c
Method default_fromtext_callback(dns_rdatacallbacks_t *callbacks, const char *fmt,

```
....
1893.       fprintf(stderr, "\n");
```

Improper Resource Access Authorization\Path 33:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=445
Status	New

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	1217	1217
Object	fprintf	fprintf

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_certificate(SSL *s)

```
....
1217.       fprintf(stderr, "pkey,x = %p, %p\n", pkey, x);
```

Improper Resource Access Authorization\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=446
Status	New

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	1218	1218
Object	fprintf	fprintf

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_certificate(SSL *s)

```
....
1218.       fprintf(stderr, "ssl_cert_type(x,pkey) = %d\n",
ssl_cert_type(x, pkey));
```

Improper Resource Access Authorization\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=447
Status	New

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c

Line	1219	1219
Object	fprintf	fprintf

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_server_certificate(SSL *s)

```
....
1219.       fprintf(stderr, "cipher, alg, nc = %s, %lx, %lx, %d\n",
```

Improper Resource Access Authorization\Path 36:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=448
Status New

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	1837	1837
Object	fprintf	fprintf

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_get_key_exchange(SSL *s)

```
....
1837.               fprintf(stderr, "USING TLSv1.2 HASH %s\n",
EVP_MD_name(md));
```

Improper Resource Access Authorization\Path 37:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=449
Status New

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	2420	2420
Object	fprintf	fprintf

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_send_client_key_exchange(SSL *s)

```
....
2420.                fprintf(stderr, "ssl3_send_client_key_exchange(%lx &
%lx)\n",
```

Improper Resource Access Authorization\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=450
Status	New

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	2436	2436
Object	fprintf	fprintf

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_send_client_key_exchange(SSL *s)

```
....
2436.                    fprintf(stderr, "kssl_cget_tkt rtn %d\n",
krb5rc);
```

Improper Resource Access Authorization\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=451
Status	New

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	2438	2438
Object	fprintf	fprintf

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_send_client_key_exchange(SSL *s)

```
....
2438.                        fprintf(stderr, "kssl_cget_tkt
kssl_err=%s\n",
```

Improper Resource Access Authorization\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=452
Status	New

	Source	Destination
File	minix-2/s3_clnt.c	minix-2/s3_clnt.c
Line	3085	3085
Object	fprintf	fprintf

Code Snippet
File Name minix-2/s3_clnt.c
Method int ssl3_send_client_verify(SSL *s)

```
....
3085.              fprintf(stderr, "Using TLS 1.2 with client alg %s\n",
```

Improper Resource Access Authorization\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=453
Status	New

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1203	1203
Object	fprintf	fprintf

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_client_hello(SSL *s)

```
....
1203.          fprintf(stderr, "client sent %d ciphers\n",
```

Improper Resource Access Authorization\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=454
Status	New

	Source	Destination

File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	1209	1209
Object	fprintf	fprintf

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_client_hello(SSL *s)

```
....
1209.               fprintf(stderr, "client [%2d of %2d]:%s\n",
```

Improper Resource Access Authorization\Path 43:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=455
Status New

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	2032	2032
Object	fprintf	fprintf

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_send_server_key_exchange(SSL *s)

```
....
2032.                   fprintf(stderr, "Using hash %s\n",
EVP_MD_name(md));
```

Improper Resource Access Authorization\Path 44:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=456
Status New

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	2488	2488
Object	fprintf	fprintf

Code Snippet
File Name minix-2/s3_srvr.c

Method	int ssl3_get_client_key_exchange(SSL *s)

```
....
2488.              fprintf(stderr, "kssl_sget_tkt rtn %d [%d]\n",
```

Improper Resource Access Authorization\Path 45:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=457
Status	New

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	2491	2491
Object	fprintf	fprintf

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_client_key_exchange(SSL *s)

```
....
2491.                  fprintf(stderr, "kssl_err text= %s\n",
kssl_err.text);
```

Improper Resource Access Authorization\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=458
Status	New

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	2504	2504
Object	fprintf	fprintf

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_client_key_exchange(SSL *s)

```
....
2504.              fprintf(stderr, "kssl_check_authent rtn %d [%d]\n",
```

Improper Resource Access Authorization\Path 47:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=459
Status	New

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	2507	2507
Object	fprintf	fprintf

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_client_key_exchange(SSL *s)

```
....
2507.                   fprintf(stderr, "kssl_err text= %s\n",
kssl_err.text);
```

Improper Resource Access Authorization\Path 48:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=460
Status	New

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	3039	3039
Object	fprintf	fprintf

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_cert_verify(SSL *s)

```
....
3039.               fprintf(stderr, "USING TLSv1.2 HASH %s\n",
EVP_MD_name(md));
```

Improper Resource Access Authorization\Path 49:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=461
Status	New

	Source	Destination

File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	3069	3069
Object	fprintf	fprintf

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_cert_verify(SSL *s)

```
....
3069.          fprintf(stderr, "Using TLS 1.2 with client verify alg
%s\n",
```

Improper Resource Access Authorization\Path 50:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=462
Status New

	Source	Destination
File	minix-2/s3_srvr.c	minix-2/s3_srvr.c
Line	3135	3135
Object	fprintf	fprintf

Code Snippet
File Name minix-2/s3_srvr.c
Method int ssl3_get_cert_verify(SSL *s)

```
....
3135.              fprintf(stderr, "GOST signature length is %d", i);
```

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=1030030&projectid=30025&pathid=146
Status New

The format_id method calls the snprintf function, at line 107 of minix-2/print-babel.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	minix-2/print-babel.c	minix-2/print-babel.c
Line	110	110
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-babel.c
Method format_id(const u_char *id)

```
....
110.      snprintf(buf, 25, "%02x:%02x:%02x:%02x:%02x:%02x:%02x:%02x",
```

Unchecked Return Value\Path 2:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=147
Status New

The format_prefix method calls the snprintf function, at line 120 of minix-2/print-babel.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	minix-2/print-babel.c	minix-2/print-babel.c
Line	124	124
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-babel.c
Method format_prefix(netdissect_options *ndo, const u_char *prefix, unsigned char plen)

```
....
124.          snprintf(buf, 50, "%s/%u", ipaddr_string(ndo, prefix +
12), plen - 96);
```

Unchecked Return Value\Path 3:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=148
Status New

The format_prefix method calls the snprintf function, at line 120 of minix-2/print-babel.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	minix-2/print-babel.c	minix-2/print-babel.c
Line	127	127
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-babel.c
Method format_prefix(netdissect_options *ndo, const u_char *prefix, unsigned char plen)

```
....
127.          snprintf(buf, 50, "%s/%u", ip6addr_string(ndo, prefix),
plen);
```

Unchecked Return Value\Path 4:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=149
Status New

The format_interval method calls the snprintf function, at line 149 of minix-2/print-babel.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	minix-2/print-babel.c	minix-2/print-babel.c
Line	155	155
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-babel.c
Method format_interval(const uint16_t i)

```
....
155.      snprintf(buf, sizeof(buf), "%u.%02us", i / 100, i % 100);
```

Unchecked Return Value\Path 5:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=150
Status New

The format_timestamp method calls the snprintf function, at line 166 of minix-2/print-babel.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	minix-2/print-babel.c	minix-2/print-babel.c
Line	169	169
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-babel.c
Method format_timestamp(const uint32_t i)

```
....
169.      snprintf(buf, sizeof(buf), "%u.%06us", i / 1000000, i %
1000000);
```

Unchecked Return Value\Path 6:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=151
Status New

The as_printf method calls the snprintf function, at line 475 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	479	479
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method as_printf(netdissect_options *ndo,

```
....
479.              snprintf(str, size, "%u", asnum);
```

Unchecked Return Value\Path 7:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=152
Status New

The as_printf method calls the snprintf function, at line 475 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	481	481
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method as_printf(netdissect_options *ndo,

```
....
481.                snprintf(str, size, "%u.%u", asnum >> 16, asnum &
0xFFFF);
```

Unchecked Return Value\Path 8:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=153
Status New

The decode_prefix4 method calls the snprintf function, at line 489 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	511	511
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method decode_prefix4(netdissect_options *ndo,

```
....
511.          snprintf(buf, buflen, "%s/%d", getname(ndo, (u_char
*)&addr), plen);
```

Unchecked Return Value\Path 9:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=154
Status New

The decode_labeled_prefix4 method calls the snprintf function, at line 522 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	560	560
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method decode_labeled_prefix4(netdissect_options *ndo,

```
....
560.         snprintf(buf, buflen, "%s/%d, label:%u %s",
```

Unchecked Return Value\Path 10:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=155
Status New

The bgp_vpn_ip_print method calls the snprintf function, at line 581 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	592	592
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method bgp_vpn_ip_print(netdissect_options *ndo,

```
....
592.           snprintf(pos, sizeof(addr), "%s", ipaddr_string(ndo,
pptr));
```

Unchecked Return Value\Path 11:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=156
Status New

The bgp_vpn_ip_print method calls the snprintf function, at line 581 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	597	597
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method bgp_vpn_ip_print(netdissect_options *ndo,

```
....
597.            snprintf(pos, sizeof(addr), "%s", ip6addr_string(ndo,
pptr));
```

Unchecked Return Value\Path 12:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=157
Status New

The bgp_vpn_ip_print method calls the snprintf function, at line 581 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	601	601
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method bgp_vpn_ip_print(netdissect_options *ndo,

```
....
601.            snprintf(pos, sizeof(addr), "bogus address length %u",
addr_length);
```

Unchecked Return Value\Path 13:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=158

Status	New

The bgp_vpn_sg_print method calls the snprintf function, at line 630 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	647	647
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method bgp_vpn_sg_print(netdissect_options *ndo,

```
....
647.          snprintf(buf + offset, buflen - offset, ", Source %s",
```

Unchecked Return Value\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=159
Status	New

The bgp_vpn_sg_print method calls the snprintf function, at line 630 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	661	661
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method bgp_vpn_sg_print(netdissect_options *ndo,

```
....
661.          snprintf(buf + offset, buflen - offset, ", Group %s",
```

Unchecked Return Value\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=160
Status	New

The bgp_vpn_rd_print method calls the snprintf function, at line 675 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	687	687
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method bgp_vpn_rd_print(netdissect_options *ndo,

```
....
687.            snprintf(pos, sizeof(rd) - (pos - rd), "%u:%u (=
%u.%u.%u.%u)",
```

Unchecked Return Value\Path 16:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=161
Status New

The bgp_vpn_rd_print method calls the snprintf function, at line 675 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	695	695
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method bgp_vpn_rd_print(netdissect_options *ndo,

```
....
695.            snprintf(pos, sizeof(rd) - (pos - rd), "%u.%u.%u.%u:%u",
```

Unchecked Return Value\Path 17:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=162
Status New

The bgp_vpn_rd_print method calls the snprintf function, at line 675 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	701	701
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method bgp_vpn_rd_print(netdissect_options *ndo,

```
....
701.         snprintf(pos, sizeof(rd) - (pos - rd), "%s:%u
(%u.%u.%u.%u:%u)",
```

Unchecked Return Value\Path 18:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=163
Status New

The bgp_vpn_rd_print method calls the snprintf function, at line 675 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	707	707
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method bgp_vpn_rd_print(netdissect_options *ndo,

```
....
707.           snprintf(pos, sizeof(rd) - (pos - rd), "unknown RD
format");
```

Unchecked Return Value\Path 19:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=164

Status	New

The decode_rt_routing_info method calls the snprintf function, at line 716 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	726	726
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method decode_rt_routing_info(netdissect_options *ndo,

```
....
726.              snprintf(buf, buflen, "default route target");
```

Unchecked Return Value\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=165
Status	New

The decode_rt_routing_info method calls the snprintf function, at line 716 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	745	745
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method decode_rt_routing_info(netdissect_options *ndo,

```
....
745.        snprintf(buf, buflen, "origin AS: %s, route target %s",
```

Unchecked Return Value\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=166
Status	New

The decode_labeled_vpn_prefix4 method calls the snprintf function, at line 756 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	781	781
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method decode_labeled_vpn_prefix4(netdissect_options *ndo,

```
....
781.        snprintf(buf, buflen, "RD: %s, %s/%d, label:%u %s",
```

Unchecked Return Value\Path 22:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=167
Status New

The decode_mdt_vpn_nlri method calls the snprintf function, at line 807 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	834	834
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method decode_mdt_vpn_nlri(netdissect_options *ndo,

```
....
834.      snprintf(buf, buflen, "RD: %s, VPN IP Address: %s, MC Group
Address: %s",
```

Unchecked Return Value\Path 23:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=168
Status New

The decode_multicast_vpn method calls the snprintf function, at line 863 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	873	873
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method decode_multicast_vpn(netdissect_options *ndo,

```
....
873.          snprintf(buf, buflen, "Route-Type: %s (%u), length: %u",
```

Unchecked Return Value\Path 24:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=169
Status New

The decode_multicast_vpn method calls the snprintf function, at line 863 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	882	882
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method decode_multicast_vpn(netdissect_options *ndo,

```
....
882.              snprintf(buf + offset, buflen - offset, ", RD: %s,
Originator %s",
```

Unchecked Return Value\Path 25:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=170
Status New

The decode_multicast_vpn method calls the snprintf function, at line 863 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	890	890
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method decode_multicast_vpn(netdissect_options *ndo,

```
....
890.            snprintf(buf + offset, buflen - offset, ", RD: %s,
Source-AS %s",
```

Unchecked Return Value\Path 26:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=171
Status New

The decode_multicast_vpn method calls the snprintf function, at line 863 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	899	899
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method decode_multicast_vpn(netdissect_options *ndo,

```
....
899.              snprintf(buf + offset, buflen - offset, ", RD: %s",
```

Unchecked Return Value\Path 27:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=172
Status New

The decode_multicast_vpn method calls the snprintf function, at line 863 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	908	908
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method decode_multicast_vpn(netdissect_options *ndo,

```
....
908.                snprintf(buf + offset, buflen - offset, ", Originator
%s",
```

Unchecked Return Value\Path 28:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=173
Status New

The decode_multicast_vpn method calls the snprintf function, at line 863 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	915	915
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method decode_multicast_vpn(netdissect_options *ndo,

```
....
915.                snprintf(buf + offset, buflen - offset, ", RD: %s",
```

Unchecked Return Value\Path 29:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=174
Status New

The decode_multicast_vpn method calls the snprintf function, at line 863 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	926	926
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method decode_multicast_vpn(netdissect_options *ndo,

```
....
926.            snprintf(buf + offset, buflen - offset, ", RD: %s,
Source-AS %s",
```

Unchecked Return Value\Path 30:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=175
Status New

The decode_prefix6 method calls the snprintf function, at line 1064 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	1086	1086
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method decode_prefix6(netdissect_options *ndo,

```
....
1086.       snprintf(buf, buflen, "%s/%d", getname6(ndo, (u_char
*)&addr), plen);
```

Unchecked Return Value\Path 31:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=176

Status	New

The decode_labeled_prefix6 method calls the snprintf function, at line 1097 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	1126	1126
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method decode_labeled_prefix6(netdissect_options *ndo,

```
....
1126.       snprintf(buf, buflen, "%s/%d, label:%u %s",
```

Unchecked Return Value\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=177
Status	New

The decode_labeled_vpn_prefix6 method calls the snprintf function, at line 1142 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	1167	1167
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method decode_labeled_vpn_prefix6(netdissect_options *ndo,

```
....
1167.       snprintf(buf, buflen, "RD: %s, %s/%d, label:%u %s",
```

Unchecked Return Value\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=178
Status	New

The decode_clnp_prefix method calls the snprintf function, at line 1182 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	1201	1201
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method decode_clnp_prefix(netdissect_options *ndo,

```
....
1201.       snprintf(buf, buflen, "%s/%d",
```

Unchecked Return Value\Path 34:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=179
Status New

The decode_labeled_vpn_clnp_prefix method calls the snprintf function, at line 1212 of minix-2/print-bgp.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	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	1237	1237
Object	snprintf	snprintf

Code Snippet
File Name minix-2/print-bgp.c
Method decode_labeled_vpn_clnp_prefix(netdissect_options *ndo,

```
....
1237.       snprintf(buf, buflen, "RD: %s, %s/%d, label:%u %s",
```

Unchecked Return Value\Path 35:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=180
Status New

The unknown_totext method calls the snprintf function, at line 795 of minix-2/rdata.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	minix-2/rdata.c	minix-2/rdata.c
Line	809	809
Object	snprintf	snprintf

Code Snippet
File Name minix-2/rdata.c
Method unknown_totext(dns_rdata_t *rdata, dns_rdata_textctx_t *tctx,

```
....
809.        snprintf(buf, sizeof(buf), "%u", sr.length);
```

Unchecked Return Value\Path 36:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=181
Status New

The btoa_totext method calls the snprintf function, at line 1861 of minix-2/rdata.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	minix-2/rdata.c	minix-2/rdata.c
Line	1877	1877
Object	snprintf	snprintf

Code Snippet
File Name minix-2/rdata.c
Method btoa_totext(unsigned char *inbuf, int inbuflen, isc_buffer_t *target) {

```
....
1877.       snprintf(buf, sizeof(buf), "x %d %x %x %x", inbuflen, Ceor,
Csum, Crot);
```

Unchecked Return Value\Path 37:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=182
Status New

The *createOurException method calls the ret function, at line 359 of minix-2/ExceptionDemo.cpp. 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	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	361	361
Object	ret	ret

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method OurUnwindException *createOurException(int type) {

```
....
361.    OurException *ret = (OurException*) memset(malloc(size), 0,
size);
```

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=1030030&projectid=30025&pathid=183
Status New

	Source	Destination
File	minix-2/print-ldp.c	minix-2/print-ldp.c
Line	570	622
Object	ldp_msg_header	sizeof

Code Snippet
File Name minix-2/print-ldp.c
Method ldp_pdu_print(netdissect_options *ndo,

```
....
570.      const struct ldp_msg_header *ldp_msg_header;
....
622.          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=1030030&projectid=30025&pathid=184

Status	New

	Source	Destination
File	minix-2/print-ldp.c	minix-2/print-ldp.c
Line	570	616
Object	ldp_msg_header	sizeof

Code Snippet
File Name minix-2/print-ldp.c
Method ldp_pdu_print(netdissect_options *ndo,

```
....
570.      const struct ldp_msg_header *ldp_msg_header;
....
616.          ND_TCHECK2(*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=1030030&projectid=30025&pathid=185
Status	New

	Source	Destination
File	minix-2/print-ldp.c	minix-2/print-ldp.c
Line	570	631
Object	ldp_msg_header	sizeof

Code Snippet
File Name minix-2/print-ldp.c
Method ldp_pdu_print(netdissect_options *ndo,

```
....
570.      const struct ldp_msg_header *ldp_msg_header;
....
631.                      (u_int)(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=1030030&projectid=30025&pathid=186
Status	New

	Source	Destination
File	minix-2/print-ldp.c	minix-2/print-ldp.c

Line	570	645
Object	ldp_msg_header	sizeof

Code Snippet
File Name minix-2/print-ldp.c
Method ldp_pdu_print(netdissect_options *ndo,

```
....
570.      const struct ldp_msg_header *ldp_msg_header;
....
645.          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=1030030&projectid=30025&pathid=187
Status New

	Source	Destination
File	minix-2/print-ldp.c	minix-2/print-ldp.c
Line	570	646
Object	ldp_msg_header	sizeof

Code Snippet
File Name minix-2/print-ldp.c
Method ldp_pdu_print(netdissect_options *ndo,

```
....
570.      const struct ldp_msg_header *ldp_msg_header;
....
646.          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=1030030&projectid=30025&pathid=188
Status New

	Source	Destination
File	minix-2/print-ldp.c	minix-2/print-ldp.c
Line	570	687
Object	ldp_msg_header	sizeof

Code Snippet	
File Name	minix-2/print-ldp.c
Method	ldp_pdu_print(netdissect_options *ndo,

```
....
570.      const struct ldp_msg_header *ldp_msg_header;
....
687.              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=1030030&projectid=30025&pathid=189
Status	New

	Source	Destination
File	minix-2/print-ldp.c	minix-2/print-ldp.c
Line	249	276
Object	ldp_tlv_header	sizeof

Code Snippet	
File Name	minix-2/print-ldp.c
Method	ldp_tlv_print(netdissect_options *ndo,

```
....
249.      const struct ldp_tlv_header *ldp_tlv_header;
....
276.      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=1030030&projectid=30025&pathid=190
Status	New

	Source	Destination
File	minix-2/str.c	minix-2/str.c
Line	162	162
Object	sizeof	sizeof

Code Snippet	
File Name	minix-2/str.c
Method	brk_string(const char *str, int *store_argc, Boolean expand, char **buffer)

```
....
162.        argv = bmake_malloc((argmax + 1) * sizeof(char *));
```

Use of Sizeof On a Pointer Type\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=191
Status	New

	Source	Destination
File	minix-2/str.c	minix-2/str.c
Line	220	220
Object	sizeof	sizeof

Code Snippet
File Name minix-2/str.c
Method brk_string(const char *str, int *store_argc, Boolean expand, char **buffer)

```
....
220.                                (argmax + 1) * sizeof(char *));
```

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=1030030&projectid=30025&pathid=252
Status	New

	Source	Destination
File	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	1194	1194
Object	addr	sizeof

Code Snippet
File Name minix-2/print-bgp.c
Method decode_clnp_prefix(netdissect_options *ndo,

```
....
1194.       memset(&addr, 0, sizeof(addr));
```

Sizeof Pointer Argument\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=253
Status	New

	Source	Destination
File	minix-2/print-bgp.c	minix-2/print-bgp.c
Line	1229	1229
Object	addr	sizeof

Code Snippet
File Name minix-2/print-bgp.c
Method decode_labeled_vpn_clnp_prefix(netdissect_options *ndo,

```
....
1229.       memset(&addr, 0, sizeof(addr));
```

Sizeof Pointer Argument\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=254
Status	New

	Source	Destination
File	minix-2/tree.c	minix-2/tree.c
Line	1702	1702
Object	name	sizeof

Code Snippet
File Name minix-2/tree.c
Method mib_mount(const int * mib, unsigned int miblen, unsigned int eid, uint32_t rid,

```
....
1702.             if ((r = mib_remote_info(eid, rid, name, sizeof(name),
scratch,
```

Sizeof Pointer Argument\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=255
Status	New

	Source	Destination
File	minix-2/tree.c	minix-2/tree.c
Line	1711	1711
Object	name	sizeof

Code Snippet
File Name minix-2/tree.c
Method mib_mount(const int * mib, unsigned int miblen, unsigned int eid, uint32_t rid,

```
....
1711.             if ((namelen = mib_check_name(name, sizeof(name))) ==
0) {
```

Potential Off by One Error in Loops

Query Path:
CPP\Cx\CPP Heuristic\Potential Off by One Error in Loops Version:1

Categories

PCI DSS v3.2: PCI DSS (3.2) - 6.5.1 - Injection flaws - particularly SQL injection
NIST SP 800-53: SI-16 Memory Protection (P1)
OWASP Top 10 2017: A1-Injection

Description
Potential Off by One Error in Loops\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=192
Status	New

The buffer allocated by <= in minix-2/ExceptionDemo.cpp at line 1648 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	1723	1723
Object	<=	<=

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method static void createStandardUtilityFunctions(unsigned numTypeInfos,

```
....
1723.     for (unsigned i = 0; i <= numTypeInfos; ++i) {
```

Potential Off by One Error in Loops\Path 2:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=193
Status	New

The buffer allocated by <= in minix-2/print-802_11.c at line 2824 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	minix-2/print-802_11.c	minix-2/print-802_11.c
Line	2882	2882
Object	<=	<=

Code Snippet
File Name minix-2/print-802_11.c
Method ieee802_11_radio_print(netdissect_options *ndo,

```
....
2882.       for (bit0 = 0, presentp = &hdr->it_present; presentp <=
last_presentp;
```

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=1030030&projectid=30025&pathid=256
Status	New

	Source	Destination
File	minix-2/str.c	minix-2/str.c
Line	112	112
Object	len1	len1

Code Snippet
File Name minix-2/str.c
Method str_concat(const char *s1, const char *s2, int flags)

```
....
112.              result[len1] = ' ';
```

Unchecked Array Index\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=257
Status	New

	Source	Destination
File	minix-2/str.c	minix-2/str.c
Line	115	115
Object	len1	len1

Code Snippet
File Name minix-2/str.c
Method str_concat(const char *s1, const char *s2, int flags)

```
....
115.              result[len1] = '/';
```

Exposure of System Data to Unauthorized Control Sphere

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

Categories

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

Description

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

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1030030&projectid=30025&pathid=463
Status	New

The system data read by runExceptionThrow in the file minix-2/ExceptionDemo.cpp at line 1604 is potentially exposed by runExceptionThrow found in minix-2/ExceptionDemo.cpp at line 1604.

	Source	Destination
File	minix-2/ExceptionDemo.cpp	minix-2/ExceptionDemo.cpp
Line	1617	1619
Object	exc	fprintf

Code Snippet
File Name minix-2/ExceptionDemo.cpp
Method void runExceptionThrow(llvm::ExecutionEngine *engine,

```
....
1617.    catch (OurCppRunException exc) {
....
1619.      fprintf(stderr,
```

Buffer Copy without Checking Size of Input ('Classic Buffer Overflow')

Weakness ID: 120 *(Weakness Base)* **Status:** Incomplete

Description

Description Summary

The program copies an input buffer to an output buffer without verifying that the size of the input buffer is less than the size of the output buffer, leading to a buffer overflow.

Extended Description

A buffer overflow condition exists when a program attempts to put more data in a buffer than it can hold, or when a program attempts to put data in a memory area outside of the boundaries of a buffer. The simplest type of error, and the most common cause of buffer overflows, is the "classic" case in which the program copies the buffer without checking its length at all. Other variants exist, but the existence of a classic overflow strongly suggests that the programmer is not considering even the most basic of security protections.

Alternate Terms

buffer overrun:	Some prominent vendors and researchers use the term "buffer overrun," but most people use "buffer overflow."
Unbounded Transfer	

Terminology Notes

Many issues that are now called "buffer overflows" are substantively different than the "classic" overflow, including entirely different bug types that rely on overflow exploit techniques, such as integer signedness errors, integer overflows, and format string bugs. This imprecise terminology can make it difficult to determine which variant is being reported.

Time of Introduction

- Implementation

Applicable Platforms

Languages

C

C++

Assembly

Common Consequences

Scope	Effect
Integrity	**Technical Impact:** *Execute unauthorized code or commands* Buffer overflows often can be used to execute arbitrary code, which is usually outside the scope of a program's implicit security policy. This can often be used to subvert any other security service.
Availability	Buffer overflows generally lead to crashes. Other attacks leading to lack of availability are possible, including putting the program into an infinite loop.

Likelihood of Exploit

High to Very 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 buffer overflows that originate from command line arguments in a program that is not expected to run with setuid or other special privileges.

Effectiveness: High

Detection techniques for buffer-related errors are more mature than for most other weakness types.

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.

Manual Analysis

Manual analysis can be useful for finding this weakness, but it might not achieve desired code coverage within limited time constraints. This becomes difficult for weaknesses that must be considered for all inputs, since the attack surface can be too large.

Demonstrative Examples

Example 1

The following code asks the user to enter their last name and then attempts to store the value entered in the last_name array.

(Bad Code)

Example Language: **C**

```
char last_name[20];
printf ("Enter your last name: ");
scanf ("%s", last_name);
```

The problem with the code above is that it does not check the size of the name entered by the user. If the user enters "Very_very_long_last_name" which is 24 characters long, then a buffer overflow will occur since the array can only hold 20 characters total.

Example 2

The following code attempts to create a local copy of a buffer to perform some manipulations to the data.

(Bad Code)

Example Language: **C**

```
void manipulate_string(char* string){
char buf[24];
strcpy(buf, string);
...
}
```

However, the programmer does not ensure that the size of the data pointed to by string will fit in the local buffer and blindly copies the data with the potentially dangerous strcpy() function. This may result in a buffer overflow condition if an attacker can influence the contents of the string parameter.

Example 3

The excerpt below calls the gets() function in C, which is inherently unsafe.

(Bad Code)

Example Language: **C**

```
char buf[24];
printf("Please enter your name and press <Enter>\n");
gets(buf);
...
}
```

However, the programmer uses the function gets() which is inherently unsafe because it blindly copies all input from STDIN to the buffer without checking size. This allows the user to provide a string that is larger than the buffer size, resulting in an overflow

condition.

Example 4

In the following example, a server accepts connections from a client and processes the client request. After accepting a client connection, the program will obtain client information using the gethostbyaddr method, copy the hostname of the client that connected to a local variable and output the hostname of the client to a log file.

(Bad Code)

Example Languages: **C and C++**

```
...
struct hostent *clienthp;
char hostname[MAX_LEN];

// create server socket, bind to server address and listen on socket
...

// accept client connections and process requests
int count = 0;
for (count = 0; count < MAX_CONNECTIONS; count++) {

int clientlen = sizeof(struct sockaddr_in);
int clientsocket = accept(serversocket, (struct sockaddr *)&clientaddr, &clientlen);

if (clientsocket >= 0) {
clienthp = gethostbyaddr((char *)&clientaddr.sin_addr.s_addr,
sizeof(clientaddr.sin_addr.s_addr), AF_INET);
strcpy(hostname, clienthp->h_name);
logOutput("Accepted client connection from host ", hostname);

// process client request
...
close(clientsocket);
}
}
close(serversocket);
...
```

However, the hostname of the client that connected may be longer than the allocated size for the local hostname variable. This will result in a buffer overflow when copying the client hostname to the local variable using the strcpy method.

Observed Examples

Reference	Description
CVE-2000-1094	buffer overflow using command with long argument
CVE-1999-0046	buffer overflow in local program using long environment variable
CVE-2002-1337	buffer overflow in comment characters, when product increments a counter for a ">" but does not decrement for "<"
CVE-2003-0595	By replacing a valid cookie value with an extremely long string of characters, an attacker may overflow the application's buffers.
CVE-2001-0191	By replacing a valid cookie value with an extremely long string of characters, an attacker may overflow the application's buffers.

Potential Mitigations

Phase: Requirements

Strategy: Language Selection

Use a language with features that can automatically mitigate or eliminate buffer overflows.

For example, many languages that perform their own memory management, such as Java and Perl, are not subject to buffer overflows. Other languages, such as Ada and C#, typically provide overflow protection, but the protection can be disabled by the

programmer.

Be wary that a language's interface to native code may still be subject to overflows, even if the language itself is theoretically safe.

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.

Examples include the Safe C String Library (SafeStr) by Messier and Viega, and the Strsafe.h library from Microsoft. These libraries provide safer versions of overflow-prone string-handling functions. This is not a complete solution, since many buffer overflows are not related to strings.

Phase: Build and Compilation

Run or compile your software using features or extensions that automatically provide a protection mechanism that mitigates or eliminates buffer overflows.

For example, certain compilers and extensions provide automatic buffer overflow detection mechanisms that are built into the compiled code. Examples include the Microsoft Visual Studio /GS flag, Fedora/Red Hat FORTIFY_SOURCE GCC flag, StackGuard, and ProPolice.

This is not necessarily a complete solution, since these mechanisms can only detect certain types of overflows. In addition, a buffer overflow attack can still cause a denial of service, since the typical response is to exit the application.

Phase: Implementation

Programmers should adhere to the following rules when allocating and managing their applications memory:

- Double check that your buffer is as large as you specify.
- When using functions that accept a number of bytes to copy, such as strncpy(), be aware that if the destination buffer size is equal to the source buffer size, it may not NULL-terminate the string.
- Check buffer boundaries if calling this function in a loop and make sure you are not in danger of writing past the allocated space.
- If necessary, truncate all input strings to a reasonable length before passing them to the copy and concatenation functions.

Phase: Operation

Use a feature like Address Space Layout Randomization (ASLR). This is not a complete solution. However, it forces the attacker to guess an unknown value that changes every program execution.

Phase: Operation

Use a CPU and operating system that offers Data Execution Protection (NX) or its equivalent. This is not a complete solution, since buffer overflows could be used to overwrite nearby variables to modify the software's state in dangerous ways. In addition, it cannot be used in cases in which self-modifying code is required.

Phases: Build and Compilation; Operation

Most mitigating technologies at the compiler or OS level to date address only a subset of buffer overflow problems and rarely provide complete protection against even that subset. It is good practice to implement strategies to increase the workload of an attacker, such as leaving the attacker to guess an unknown value that changes every program execution.

Phase: Implementation

Replace unbounded copy functions with analogous functions that support length arguments, such as strcpy with strncpy. Create these if they are not available.

Effectiveness: Moderate

This approach is still susceptible to calculation errors, including issues such as off-by-one errors (CWE-193) and incorrectly calculating buffer lengths (CWE-131).

Weakness Ordinalities

Ordinality	Description
Resultant	*(where the weakness is typically related to the presence of some other weaknesses)*
Primary	*(where the weakness exists independent of other weaknesses)*

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	20	Improper Input	**Seven Pernicious**

			Validation	**Kingdoms (primary)700**
ChildOf	Weakness Class	119	Failure to Constrain Operations within the Bounds of a Memory Buffer	**Development Concepts (primary)699 Research Concepts (primary)1000**
ChildOf	Category	633	Weaknesses that Affect Memory	**Resource-specific Weaknesses (primary)631**
ChildOf	Category	722	OWASP Top Ten 2004 Category A1 - Unvalidated Input	Weaknesses in OWASP Top Ten (2004)711
ChildOf	Category	726	OWASP Top Ten 2004 Category A5 - Buffer Overflows	**Weaknesses in OWASP Top Ten (2004) (primary)711**
ChildOf	Category	741	CERT C Secure Coding Section 07 - Characters and Strings (STR)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
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 Base	123	Write-what-where Condition	Research Concepts1000
ParentOf	Weakness Variant	785	Use of Path Manipulation Function without Maximum-sized Buffer	**Development Concepts (primary)699** Research Concepts1000
CanFollow	Weakness Base	170	Improper Null Termination	Research Concepts1000
CanFollow	Weakness Base	231	Improper Handling of Extra Values	Research Concepts1000
CanFollow	Weakness Base	242	Use of Inherently Dangerous Function	Research Concepts1000
CanFollow	Weakness Base	416	Use After Free	Research Concepts1000
CanFollow	Weakness Base	456	Missing Initialization	Research Concepts1000
PeerOf	Weakness Base	124	Buffer Underwrite ('Buffer Underflow')	Research Concepts1000
CanAlsoBe	Weakness Variant	196	Unsigned to Signed Conversion Error	Research Concepts1000

Relationship Notes

At the code level, stack-based and heap-based overflows do not differ significantly, so there usually is not a need to distinguish them. From the attacker perspective, they can be quite different, since different techniques are required to exploit them.

Affected Resources

- Memory

Functional Areas

- Memory Management

f Causal Nature

Explicit

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
PLOVER			Unbounded Transfer ('classic overflow')
7 Pernicious Kingdoms			Buffer Overflow
CLASP			Buffer overflow
OWASP Top Ten 2004	A1	CWE More Specific	Unvalidated Input

OWASP Top Ten 2004	A5	CWE More Specific	Buffer Overflows
CERT C Secure Coding	STR35-C		Do not copy data from an unbounded source to a fixed-length array
WASC	7		Buffer Overflow

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
8	Buffer Overflow in an API Call	
9	Buffer Overflow in Local Command-Line Utilities	
10	Buffer Overflow via Environment Variables	
14	Client-side Injection-induced Buffer Overflow	
24	Filter Failure through Buffer Overflow	
92	Forced Integer Overflow	
42	MIME Conversion	
44	Overflow Binary Resource File	
45	Buffer Overflow via Symbolic Links	
100	Overflow Buffers	
46	Overflow Variables and Tags	
47	Buffer Overflow via Parameter Expansion	
67	String Format Overflow in syslog()	

White Box Definitions

A weakness where the code path includes a Buffer Write Operation such that:

1. the expected size of the buffer is greater than the actual size of the buffer where expected size is equal to the sum of the size of the data item and the position in the buffer

Where Buffer Write Operation is a statement that writes a data item of a certain size into a buffer at a certain position and at a certain index

References

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 5, "Public Enemy #1: The Buffer Overrun" Page 127. 2nd Edition. Microsoft. 2002.

[REF-17] Michael Howard, David LeBlanc and John Viega. "24 Deadly Sins of Software Security". "Sin 5: Buffer Overruns." Page 89. McGraw-Hill. 2010.

Microsoft. "Using the Strsafe.h Functions". <http://msdn.microsoft.com/en-us/library/ms647466.aspx>.

Matt Messier and John Viega. "Safe C String Library v1.0.3". <http://www.zork.org/safestr/>.

Michael Howard. "Address Space Layout Randomization in Windows Vista". <http://blogs.msdn.com/michael_howard/archive/2006/05/26/address-space-layout-randomization-in-windows-vista.aspx>.

Arjan van de Ven. "Limiting buffer overflows with ExecShield". <http://www.redhat.com/magazine/009jul05/features/execshield/>.

"PaX". <http://en.wikipedia.org/wiki/PaX>.

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 Alternate Terms, Applicable Platforms, Common Consequences, Relationships, Observed Example, Other Notes, Taxonomy Mappings, Weakness Ordinalities		
2008-10-10	CWE Content Team	MITRE	Internal
	Changed name and description to more clearly emphasize the "classic" nature of the overflow.		
2008-10-14	CWE Content Team	MITRE	Internal
	updated Alternate Terms, Description, Name, Other Notes, Terminology Notes		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Other Notes, Relationships, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences, Other Notes, Potential Mitigations, References, Relationship Notes, Relationships		
2009-07-27	CWE Content Team	MITRE	Internal
	updated Other Notes, Potential Mitigations, Relationships		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Common Consequences, Relationships		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Demonstrative Examples, Detection Factors, Potential Mitigations, References, Related Attack Patterns, Relationships, Taxonomy Mappings, Time of Introduction, Type		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples, Related Attack Patterns		

Previous Entry Names

Change Date	Previous Entry Name
2008-10-14	Unbounded Transfer ('Classic Buffer Overflow')

BACK TO TOP

Buffer Overflow boundcpy WrongSizeParam

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.
Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

CPP

Overflowing Buffers

```
const int BUFFER_SIZE = 10;
char buffer[BUFFER_SIZE];

void copyStringToBuffer(char* inputString)
{
    strcpy(buffer, inputString);
}
```

Checked Buffers

```
const int BUFFER_SIZE = 10;
const int MAX_INPUT_SIZE = 256;
char buffer[BUFFER_SIZE];

void copyStringToBuffer(char* inputString)
```

```
{
    if (strnlen(inputString, MAX_INPUT_SIZE) < sizeof(buffer))
    {
        strncpy(buffer, inputString, sizeof(buffer));
    }
}
```

Wrong Size t Allocation

Risk

What might happen

Incorrect allocation of memory may result in unexpected behavior by either overwriting sections of memory with unexpected values. Under certain conditions where both an incorrect allocation of memory and the values being written can be controlled by an attacker, such an issue may result in execution of malicious code.

Cause

How does it happen

Some memory allocation functions require a size value to be provided as a parameter. The allocated size should be derived from the provided value, by providing the length value of the intended source, multiplied by the size of that length. Failure to perform the correct arithmetic to obtain the exact size of the value will likely result in the source overflowing its destination.

General Recommendations

How to avoid it

- Always perform the correct arithmetic to determine size.
- Specifically for memory allocation, calculate the allocation size from the allocation source:
    - Derive the size value from the length of intended source to determine the amount of units to be processed.
    - Always programmatically consider the size of the each unit and their conversion to memory units - for example, by using sizeof() on the unit's type.
    - Memory allocation should be a multiplication of the amount of units being written, times the size of each unit.

Source Code Examples

CPP

Allocating and Assigning Memory without Sizeof Arithmetic

```
int *ptr;
ptr = (int*)malloc(5);
for (int i = 0; i < 5; i++)
{
      ptr[i] = i * 2 + 1;
}
```

Allocating and Assigning Memory with Sizeof Arithmetic

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
for (int i = 0; i < 5; i++)
{
      ptr[i] = i * 2 + 1;
```

```
}
```

Incorrect Arithmetic of Multi-Byte String Allocation

```
wchar_t * dest;
dest = (wchar_t *)malloc(wcslen(source) + 1); // Would not crash for a short "source"
wcscpy((wchar_t *)dest, source);
wprintf(L"Dest: %s\r\n", dest);
```

Correct Arithmetic of Multi-Byte String Allocation

```
wchar_t * dest;
dest = (wchar_t *)malloc((wcslen(source) + 1) * sizeof(wchar_t));
wcscpy((wchar_t *)dest, source);
wprintf(L"Dest: %s\r\n", dest);
```

Integer Overflow

Risk

What might happen

Assigning large data types into smaller data types, without proper checks and explicit casting, will lead to undefined behavior and unintentional effects, such as data corruption (e.g. value wraparound, wherein maximum values become minimum values); system crashes; infinite loops; logic errors, such as bypassing of security mechanisms; or even buffer overflows leading to arbitrary code execution.

Cause

How does it happen

This flaw can occur when implicitly casting numerical data types of a larger size, into a variable with a data type of a smaller size. This forces the program to discard some bits of information from the number. Depending on how the numerical data types are stored in memory, this is often the bits with the highest value, causing substantial corruption of the stored number. Alternatively, the sign bit of a signed integer could be lost, completely reversing the intention of the number.

General Recommendations

How to avoid it

- Avoid casting larger data types to smaller types.
- Prefer promoting the target variable to a large enough data type.
- If downcasting is necessary, always check that values are valid and in range of the target type, before casting

---

Source Code Examples

CPP

Unsafe Downsize Casting

```
int unsafe_addition(short op1, int op2) {

    // op2 gets forced from int into a short
    short total = op1 + op2;

    return total;
}
```

Safer Use of Proper Data Types

```
int safe_addition(short op1, int op2) {

    // total variable is of type int, the largest type that is needed
    int total = 0;

    // check if total will overflow available integer size
    if (INT_MAX - abs(op2) > op1)
    {
        total = op1 + op2;
    }
    else
```

```
    {
        // instead of overflow, saturate (but this is not always a good thing)
        total = INT_MAX
    }

    return total;
}
```

Dangerous Functions

Risk

What might happen

Use of dangerous functions may expose varying risks associated with each particular function, with potential impact of improper usage of these functions varying significantly. The presence of such functions indicates a flaw in code maintenance policies and adherence to secure coding practices, in a way that has allowed introducing known dangerous code into the application.

Cause

How does it happen

A dangerous function has been identified within the code. Functions are often deemed dangerous to use for numerous reasons, as there are different sets of vulnerabilities associated with usage of such functions. For example, some string copy and concatenation functions are vulnerable to Buffer Overflow, Memory Disclosure, Denial of Service and more. Use of these functions is not recommended.

General Recommendations

How to avoid it

- Deploy a secure and recommended alternative to any functions that were identified as dangerous.
    - If no secure alternative is found, conduct further researching and testing to identify whether current usage successfully sanitizes and verifies values, and thus successfully avoids the use-cases for whom the function is indeed dangerous
- Conduct a periodical review of methods that are in use, to ensure that all external libraries and built-in functions are up-to-date and whose use has not been excluded from best secure coding practices.

Source Code Examples

CPP

Buffer Overflow in gets()

```
int main()
{
	char buf[10];

	printf("Please enter your name: ");
	gets(buf); // veryveryverylongname
	if (buf == ACCEPTED_NAME)
	{
		// Do something
	}
	return 0;
}
```

Safe reading from user

```
int main()

{

	char buf[10];

	printf("Please enter your name: ");
	fgets(buf, sizeof(buf), stdin); //setting the amount of bytes to read
	if (buf == ACCEPTED_NAME)
	{
		//Do something
	}
	return 0;
}
```

Unsafe function for string copy

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

{

	char buf[10];
	strcpy(buf, argv[1]); // overflow occurs when len(argv[1]) > 10 bytes

	return 0;
}
```

Safe string copy

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

{

	char buf[10];
	strncpy(buf, argv[1], sizeof(buf));
	buf[9]= '\0'; //strncpy doesn't NULL terminates

	return 0;
}
```

Unsafe format string

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

{

	printf(argv[1]); // If argv[1] contains a format token, such as %s,%x or %d, will cause
an access violation
	return 0;
}
```

Safe format string

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

{

	printf("%s", argv[1]); // Second parameter is not a formattable string

	return 0;
}
```

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;
	}
}
```

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;
}
```

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 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());
```

Unchecked Return Value

Risk

What might happen

A program that does not check function return values could cause the application to enter an undefined state. This could lead to unexpected behavior and unintended consequences, including inconsistent data, system crashes or other error-based exploits.

Cause

How does it happen

The application calls a system function, but does not receive or check the result of this funciton. These functions often return error codes in the result, or share other status codes with it's caller. The application simply ignores this result value, losing this vital information.

General Recommendations

How to avoid it

- Always check the result of any called function that returns a value, and verify the result is an expected value.

- Ensure the calling function responds to all possible return values.

- Expect runtime errors and handle them gracefully. Explicitly define a mechanism for handling unexpected errors.

Source Code Examples

CPP

Unchecked Memory Allocation

```
buff = (char*) malloc(size);
strncpy(buff, source, size);
```

Safer Memory Allocation

```
buff = (char*) malloc(size+1);
if (buff==NULL) exit(1);

strncpy(buff, source, size);
buff[size] = '\0';
```

Use of sizeof() on a Pointer Type

Weakness ID: 467 *(Weakness Variant)* **Status:** Draft

Description

Description Summary

The code calls sizeof() on a malloced pointer type, which always returns the wordsize/8. This can produce an unexpected result if the programmer intended to determine how much memory has been allocated.

Time of Introduction

- Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

Scope	Effect
Integrity	This error can often cause one to allocate a buffer that is much smaller than what is needed, leading to resultant weaknesses such as buffer overflows.

Likelihood of Exploit

High

Demonstrative Examples

Example 1

Care should be taken to ensure sizeof returns the size of the data structure itself, and not the size of the pointer to the data structure.

In this example, sizeof(foo) returns the size of the pointer.

(Bad Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(foo));
```

In this example, sizeof(*foo) returns the size of the data structure and not the size of the pointer.

(Good Code)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(*foo));
```

Example 2

This example defines a fixed username and password. The AuthenticateUser() function is intended to accept a username and a password from an untrusted user, and check to ensure that it matches the username and password. If the username and password match, AuthenticateUser() is intended to indicate that authentication succeeded.

(Bad Code)

```
/* Ignore CWE-259 (hard-coded password) and CWE-309 (use of password system for authentication) for this example. */

char *username = "admin";
char *pass = "password";

int AuthenticateUser(char *inUser, char *inPass) {
```

```
printf("Sizeof username = %d\n", sizeof(username));
printf("Sizeof pass = %d\n", sizeof(pass));

if (strncmp(username, inUser, sizeof(username))) {
printf("Auth failure of username using sizeof\n");
return(AUTH_FAIL);
}
/* Because of CWE-467, the sizeof returns 4 on many platforms and architectures. */
if (! strncmp(pass, inPass, sizeof(pass))) {
printf("Auth success of password using sizeof\n");
return(AUTH_SUCCESS);
}
else {
printf("Auth fail of password using sizeof\n");
return(AUTH_FAIL);
}
}

int main (int argc, char **argv)
{
int authResult;

if (argc < 3) {
ExitError("Usage: Provide a username and password");
}
authResult = AuthenticateUser(argv[1], argv[2]);
if (authResult != AUTH_SUCCESS) {
ExitError("Authentication failed");
}
else {
DoAuthenticatedTask(argv[1]);
}
}
```

In AuthenticateUser(), because sizeof() is applied to a parameter with an array type, the sizeof() call might return 4 on many modern architectures. As a result, the strncmp() call only checks the first four characters of the input password, resulting in a partial comparison (CWE-187), leading to improper authentication (CWE-287).

Because of the partial comparison, any of these passwords would still cause authentication to succeed for the "admin" user:

(Attack)

```
pass5
passABCDEFGH
passWORD
```

Because only 4 characters are checked, this significantly reduces the search space for an attacker, making brute force attacks more feasible.

The same problem also applies to the username, so values such as "adminXYZ" and "administrator" will succeed for the username.

Potential Mitigations

Phase: Implementation

Use expressions such as "sizeof(*pointer)" instead of "sizeof(pointer)", unless you intend to run sizeof() on a pointer type to gain some platform independence or if you are allocating a variable on the stack.

Other Notes

The use of sizeof() on a pointer can sometimes generate useful information. An obvious case is to find out the wordsize on a platform. More often than not, the appearance of sizeof(pointer) indicates a bug.

Weakness Ordinalities

Ordinality	Description
Primary	*(where the weakness exists independent of other weaknesses)*

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	465	Pointer Issues	**Development Concepts (primary)699**
ChildOf	Weakness Class	682	Incorrect Calculation	**Research Concepts (primary)1000**
ChildOf	Category	737	CERT C Secure Coding Section 03 - Expressions (EXP)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
ChildOf	Category	740	CERT C Secure Coding Section 06 - Arrays (ARR)	Weaknesses Addressed by the CERT C Secure Coding Standard734
CanPrecede	Weakness Base	131	Incorrect Calculation of Buffer Size	Research Concepts1000

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Use of sizeof() on a pointer type
CERT C Secure Coding	ARR01-C		Do not apply the sizeof operator to a pointer when taking the size of an array
CERT C Secure Coding	EXP01-C		Do not take the size of a pointer to determine the size of the pointed-to type

White Box Definitions

A weakness where code path has:

1. end statement that passes an identity of a dynamically allocated memory resource to a sizeof operator
2. start statement that allocates the dynamically allocated memory resource

References

Robert Seacord. "EXP01-A. Do not take the sizeof a pointer to determine the size of a type". <https://www.securecoding.cert.org/confluence/display/seccode/EXP01-A.+Do+not+take+the+sizeof+a+pointer+to+determine+the+size+of+a+type>.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	CLASP		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		

BACK TO TOP

Potential Off by One Error in Loops

Risk

What might happen

An off by one error may result in overwriting or over-reading of unintended memory; in most cases, this can result in unexpected behavior and even application crashes. In other cases, where allocation can be controlled by an attacker, a combination of variable assignment and an off by one error can result in execution of malicious code.

Cause

How does it happen

Often when designating variables to memory, a calculation error may occur when determining size or length that is off by one.

For example in loops, when allocating an array of size 2, its cells are counted as 0,1 - therefore, if a For loop iterator on the array is incorrectly set with the start condition i=0 and the continuation condition i<=2, three cells will be accessed instead of 2, and an attempt will be made to write or read cell [2], which was not originally allocated, resulting in potential corruption of memory outside the bounds of the originally assigned array.

Another example occurs when a null-byte terminated string, in the form of a character array, is copied without its terminating null-byte. Without the null-byte, the string representation is unterminated, resulting in certain functions to over-read memory as they expect the missing null terminator.

General Recommendations

How to avoid it

- Always ensure that a given iteration boundary is correct:
  - With array iterations, consider that arrays begin with cell 0 and end with cell n-1, for a size n array.
  - With character arrays and null-byte terminated string representations, consider that the null byte is required and should not be overwritten or ignored; ensure functions in use are not vulnerable to off-by-one, specifically for instances where null-bytes are automatically appended after the buffer, instead of in place of its last character.
- Where possible, use safe functions that manage memory and are not prone to off-by-one errors.

Source Code Examples

CPP

Off-By-One in For Loop

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
for (int i = 0; i <= 5; i++)
{
      ptr[i] = i * 2 + 1; // ptr[5] will be set, but is out of bounds
```

```
}
```

Proper Iteration in For Loop

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
for (int i = 0; i < 5; i++)
{
      ptr[i] = i * 2 + 1; // ptr[0-4] are well defined
}
```

Off-By-One in strncat

```
strncat(buf, input, sizeof(buf) - strlen(buf)); // actual value should be sizeof(buf)-
strlen(buf)-1 - this form will overwrite the terminating nullbyte
```

NULL Pointer Dereference

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
- Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
- Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.

Source Code Examples

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

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 {
    //[..]
};
```

Scanned Languages

Language	Hash Number	Change Date
CPP	4541647240435660	6/19/2024
Common	0105849645654507	6/19/2024