ESP8266_RTOS_SDK Scan Report

Project Name	ESP8266_RTOS_SDK
Scan Start	Friday, June 21, 2024 11:19:07 PM
Preset	Checkmarx Default
Scan Time	00h:07m:18s
Lines Of Code Scanned	18337
Files Scanned	12
Report Creation Time	Friday, June 21, 2024 11:27:21 PM
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063
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

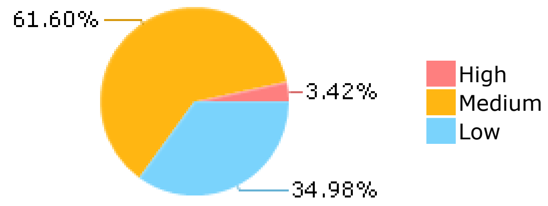
Results Limit

Results limit per query was set to 50

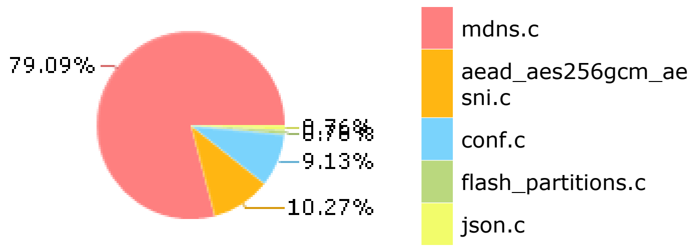
Selected Queries

Selected queries are listed in Result Summary

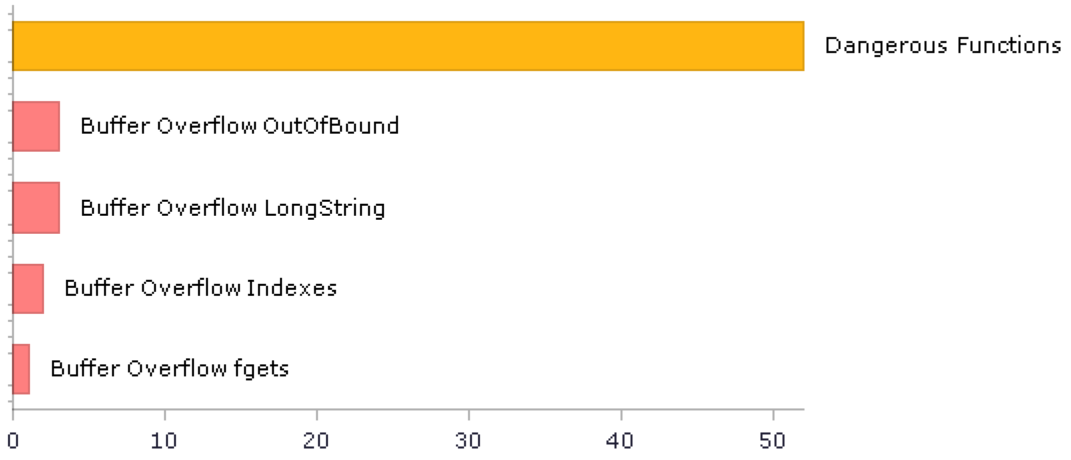
Result Summary



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	100	41
A2-Broken Authentication	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	12	12
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	0	0
A7-Cross-Site Scripting (XSS)	App. Specific	EASY	WIDESPREAD	EASY	MODERATE	App. Specific	0	0
A8-Insecure Deserialization	App. Specific	DIFFICULT	COMMON	AVERAGE	SEVERE	App. Specific	0	0
A9-Using Components with Known Vulnerabilities*	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	MODERATE	App. Specific	52	52
A10-Insufficient Logging & Monitoring	App. Specific	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	App. Specific	0	0

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

Scan Summary - OWASP Top 10 2013

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

Category	Threat Agent	Attack Vectors	Weakness Prevalence	Weakness Detectability	Technical Impact	Business Impact	Issues Found	Best Fix Locations
A1-Injection	EXTERNAL, INTERNAL, ADMIN USERS	EASY	COMMON	AVERAGE	SEVERE	ALL DATA	0	0
A2-Broken Authentication and Session Management	EXTERNAL, INTERNAL USERS	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	AFFECTED DATA AND FUNCTIONS	0	0
A3-Cross-Site Scripting (XSS)	EXTERNAL, INTERNAL, ADMIN USERS	AVERAGE	VERY WIDESPREAD	EASY	MODERATE	AFFECTED DATA AND SYSTEM	0	0
A4-Insecure Direct Object References	SYSTEM USERS	EASY	COMMON	EASY	MODERATE	EXPOSED DATA	0	0
A5-Security Misconfiguration	EXTERNAL, INTERNAL, ADMIN USERS	EASY	COMMON	EASY	MODERATE	ALL DATA AND SYSTEM	0	0
A6-Sensitive Data Exposure	EXTERNAL, INTERNAL, ADMIN USERS, USERS BROWSERS	DIFFICULT	UNCOMMON	AVERAGE	SEVERE	EXPOSED DATA	0	0
A7-Missing Function Level Access Control*	EXTERNAL, INTERNAL USERS	EASY	COMMON	AVERAGE	MODERATE	EXPOSED DATA AND FUNCTIONS	0	0
A8-Cross-Site Request Forgery (CSRF)	USERS BROWSERS	AVERAGE	COMMON	EASY	MODERATE	AFFECTED DATA AND FUNCTIONS	0	0
A9-Using Components with Known Vulnerabilities*	EXTERNAL USERS, AUTOMATED TOOLS	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	AFFECTED DATA AND FUNCTIONS	52	52
A10-Unvalidated Redirects and Forwards	USERS BROWSERS	AVERAGE	WIDESPREAD	DIFFICULT	MODERATE	AFFECTED DATA AND FUNCTIONS	0	0

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

Scan Summary - PCI DSS v3.2

Category	Issues Found	Best Fix Locations
PCI DSS (3.2) - 6.5.1 - Injection flaws - particularly SQL injection	0	0
PCI DSS (3.2) - 6.5.2 - Buffer overflows	43	41
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.	2	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.	12	12
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.	4	4

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

Scan Summary - NIST SP 800-53

Category	Issues Found	Best Fix Locations
AC-12 Session Termination (P2)	0	0
AC-3 Access Enforcement (P1)	14	13
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)	1	1
SC-4 Information in Shared Resources (P1)	0	0
SC-5 Denial of Service Protection (P1)*****	82	20
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*****	26	24
SI-11 Error Handling (P2)*****	5	5
SI-15 Information Output Filtering (P0)	0	0
SI-16 Memory Protection (P1)	30	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

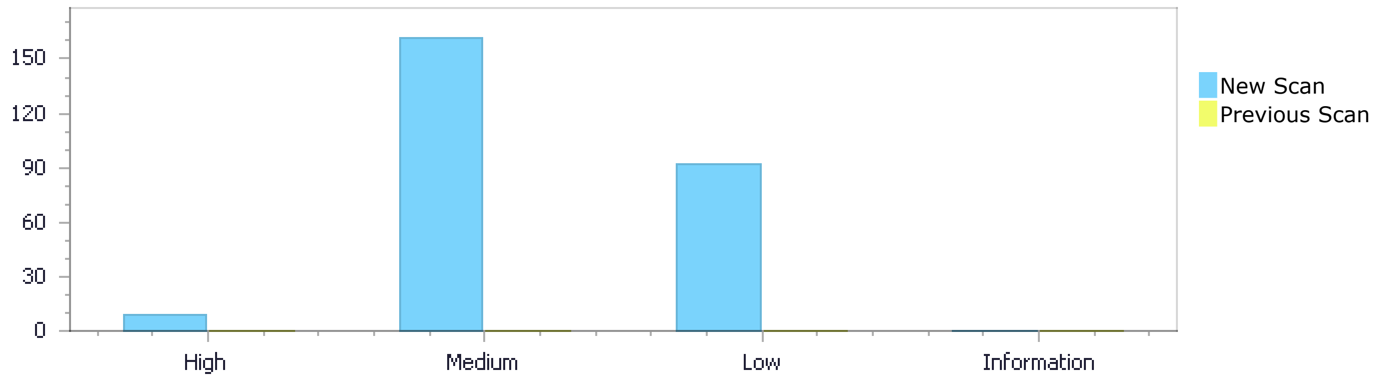
Category	Issues Found	Best Fix Locations
Must audit	0	0
Check	0	0
Optional	0	0

Results Distribution By Status

First scan of the project

	High	Medium	Low	Information	Total
New Issues	9	162	92	0	263
Recurrent Issues	0	0	0	0	0
Total	9	162	92	0	263

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	9	162	92	0	263
Urgent	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0
Total	9	162	92	0	263

Result Summary

Vulnerability Type	Occurrences	Severity
Buffer Overflow LongString	3	High
Buffer Overflow OutOfBound	3	High
Buffer Overflow Indexes	2	High
Buffer Overflow fgets	1	High
Dangerous Functions	52	Medium

Double Free	30	Medium
Buffer Overflow boundcpy WrongSizeParam	25	Medium
MemoryFree on StackVariable	24	Medium
Memory Leak	10	Medium
Use of Zero Initialized Pointer	10	Medium
Char Overflow	5	Medium
Integer Overflow	4	Medium
Stored Buffer Overflow fgets	1	Medium
Wrong Size t Allocation	1	Medium
NULL Pointer Dereference	62	Low
Improper Resource Access Authorization	12	Low
Unchecked Return Value	5	Low
Unchecked Array Index	4	Low
Potential Precision Problem	3	Low
Exposure of System Data to Unauthorized Control Sphere	2	Low
Use of Sizeof On a Pointer Type	2	Low
Inconsistent Implementations	1	Low
Use of Insufficiently Random Values	1	Low

10 Most Vulnerable Files

High and Medium Vulnerabilities

File Name	Issues Found
ESP8266_RTOS_SDK/mdns.c	136
ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	24
ESP8266_RTOS_SDK/conf.c	8
ESP8266_RTOS_SDK/flash_partitions.c	2
ESP8266_RTOS_SDK/json.c	1

Scan Results Details

Buffer Overflow LongString

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

Categories

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

Description

Buffer Overflow LongString\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=1
Status	New

The size of the buffer used by _mdns_append_sdptr_record in sd_str, at line 489 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that _mdns_append_sdptr_record passes to "_services", at line 489 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	500	500
Object	"_services"	sd_str

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static uint16_t _mdns_append_sdptr_record(uint8_t * packet, uint16_t * index, mdns_service_t * service, bool flush, bool bye)

```
....
500.      sd_str[0] = (char*)"_services";
```

Buffer Overflow LongString\Path 2:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=2
Status	New

The size of the buffer used by _mdns_append_sdptr_record in sd_str, at line 489 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that _mdns_append_sdptr_record passes to "_dns-sd", at line 489 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	501	501
Object	"_dns-sd"	sd_str

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static uint16_t _mdns_append_sdptr_record(uint8_t * packet, uint16_t * index, mdns_service_t * service, bool flush, bool bye)

```
....
501.        sd_str[1] = (char*)"_dns-sd";
```

Buffer Overflow LongString\Path 3:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=3
Status	New

The size of the buffer used by _mdns_append_sdptr_record in sd_str, at line 489 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that _mdns_append_sdptr_record passes to "_udp", at line 489 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	502	502
Object	"_udp"	sd_str

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static uint16_t _mdns_append_sdptr_record(uint8_t * packet, uint16_t * index, mdns_service_t * service, bool flush, bool bye)

```
....
502.        sd_str[2] = (char*)"_udp";
```

Buffer Overflow OutOfBound

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

Categories

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

Description

Buffer Overflow OutOfBound\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=4
Status	New

The size of the buffer used by crypto_aead_aes256gcm_encrypt_detached_afternm in i, at line 503 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that crypto_aead_aes256gcm_encrypt_detached_afternm passes to T, at line 503 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	520	614
Object	T	i

Code Snippet

File Name	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method	crypto_aead_aes256gcm_encrypt_detached_afternm(unsigned char *c,

```
....
520.      CRYPTO_ALIGN(16) unsigned char T[16];
....
614.          mac[i] = T[i] ^ accum[15 - i];
```

Buffer Overflow OutOfBound\Path 2:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=5
Status	New

The size of the buffer used by crypto_aead_aes256gcm_encrypt_detached_afternm in i, at line 503 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that crypto_aead_aes256gcm_encrypt_detached_afternm passes to T, at line 503 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	520	614
Object	T	i

Code Snippet

File Name	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method	crypto_aead_aes256gcm_encrypt_detached_afternm(unsigned char *c,

```
....
520.      CRYPTO_ALIGN(16) unsigned char T[16];
....
614.          mac[i] = T[i] ^ accum[15 - i];
```

Buffer Overflow OutOfBound\Path 3:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=6
Status	New

The size of the buffer used by crypto_aead_aes256gcm_decrypt_detached_afternm in i, at line 642 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that crypto_aead_aes256gcm_decrypt_detached_afternm passes to T, at line 642 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	659	778
Object	T	i

Code Snippet

File Name ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c

Method crypto_aead_aes256gcm_decrypt_detached_afternm(unsigned char *m, unsigned char *nsec,

```
....
659.      CRYPTO_ALIGN(16) unsigned char T[16];
....
778.              d |= (mac[i] ^ (T[i] ^ accum[15 - i]));
```

Buffer Overflow Indexes

Query Path:

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

Categories

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

Description

Buffer Overflow Indexes\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=16
Status	New

The size of the buffer used by conf_string in line, at line 134 of ESP8266_RTOS_SDK/conf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that conf_askvalue passes to stdin, at line 85 of ESP8266_RTOS_SDK/conf.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	113	159
Object	stdin	line

Code Snippet
File Name ESP8266_RTOS_SDK/conf.c
Method static int conf_askvalue(struct symbol *sym, const char *def)

```
....
113.            xfgets(line, sizeof(line), stdin);
```

File Name ESP8266_RTOS_SDK/conf.c
Method static int conf_string(struct menu *menu)

```
....
159.                    line[strlen(line)-1] = 0;
```

Buffer Overflow Indexes\Path 2:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=17
Status	New

The size of the buffer used by conf_string in strlen, at line 134 of ESP8266_RTOS_SDK/conf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that conf_askvalue passes to stdin, at line 85 of ESP8266_RTOS_SDK/conf.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	113	159
Object	stdin	strlen

Code Snippet
File Name ESP8266_RTOS_SDK/conf.c
Method static int conf_askvalue(struct symbol *sym, const char *def)

```
....
113.            xfgets(line, sizeof(line), stdin);
```

File Name	ESP8266_RTOS_SDK/conf.c
Method	static int conf_string(struct menu *menu)

```
....
159.                    line[strlen(line)-1] = 0;
```

Buffer Overflow fgets

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

Categories

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

Description

Buffer Overflow fgets\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=67
Status	New

The size of the buffer used by xfgets in size, at line 719 of ESP8266_RTOS_SDK/conf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that conf_choice passes to stdin, at line 236 of ESP8266_RTOS_SDK/conf.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	315	721
Object	stdin	size

Code Snippet

File Name	ESP8266_RTOS_SDK/conf.c
Method	static int conf_choice(struct menu *menu)

```
....
315.                    xfgets(line, sizeof(line), stdin);
```

▼

File Name	ESP8266_RTOS_SDK/conf.c
Method	void xfgets(char *str, int size, FILE *in)

```
....
721.          if (fgets(str, size, in) == NULL)
```

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=1050073&projectid=50063&pathid=147
Status	New

The dangerous function, memcpy, was found in use at line 503 in ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	525	525
Object	memcpy	memcpy

Code Snippet
File Name ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method crypto_aead_aes256gcm_encrypt_detached_afternm(unsigned char *c,

```
....
525.        memcpy(H, ctx->H, sizeof H);
```

Dangerous Functions\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=148
Status	New

The dangerous function, memcpy, was found in use at line 503 in ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	529	529
Object	memcpy	memcpy

Code Snippet
File Name ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c

Method crypto_aead_aes256gcm_encrypt_detached_afternm(unsigned char *c,

```
....
529.        memcpy(&n2[0], npub, 3 * 4);
```

Dangerous Functions\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=149
Status	New

The dangerous function, memcpy, was found in use at line 503 in ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	535	535
Object	memcpy	memcpy

Code Snippet

File Name ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c

Method crypto_aead_aes256gcm_encrypt_detached_afternm(unsigned char *c,

```
....
535.            memcpy(&fb[0], &x, sizeof x);
```

Dangerous Functions\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=150
Status	New

The dangerous function, memcpy, was found in use at line 503 in ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	537	537
Object	memcpy	memcpy

Code Snippet

File Name ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method crypto_aead_aes256gcm_encrypt_detached_afternm(unsigned char *c,

```
....
537.          memcpy(&fb[8], &x, sizeof x);
```

Dangerous Functions\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=151
Status	New

The dangerous function, memcpy, was found in use at line 642 in ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	669	669
Object	memcpy	memcpy

Code Snippet
File Name ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method crypto_aead_aes256gcm_decrypt_detached_afternm(unsigned char *m, unsigned char *nsec,

```
....
669.      memcpy(&n2[0], npub, 3 * 4);
```

Dangerous Functions\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=152
Status	New

The dangerous function, memcpy, was found in use at line 642 in ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	676	676
Object	memcpy	memcpy

Code Snippet

File Name	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method	crypto_aead_aes256gcm_decrypt_detached_afternm(unsigned char *m, unsigned char *nsec,

```
....
676.           memcpy(&fb[0], &x, sizeof x);
```

Dangerous Functions\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=153
Status	New

The dangerous function, memcpy, was found in use at line 642 in ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	678	678
Object	memcpy	memcpy

Code Snippet

File Name	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method	crypto_aead_aes256gcm_decrypt_detached_afternm(unsigned char *m, unsigned char *nsec,

```
....
678.           memcpy(&fb[8], &x, sizeof x);
```

Dangerous Functions\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=154
Status	New

The dangerous function, memcpy, was found in use at line 642 in ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c

Line	681	681
Object	memcpy	memcpy

Code Snippet

File Name	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method	crypto_aead_aes256gcm_decrypt_detached_afternm(unsigned char *m, unsigned char *nsec,

```
....
681.        memcpy(H, ctx->H, sizeof H);
```

Dangerous Functions\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=155
Status	New

The dangerous function, memcpy, was found in use at line 108 in ESP8266_RTOS_SDK/flash_partitions.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/flash_partitions.c	ESP8266_RTOS_SDK/flash_partitions.c
Line	120	120
Object	memcpy	memcpy

Code Snippet

File Name	ESP8266_RTOS_SDK/flash_partitions.c
Method	esp_err_t esp_partition_table_basic_verify(const esp_partition_info_t *partition_table, bool log_errors, int *num_partitions)

```
....
120.            memcpy(&part_local, (void *)((intptr_t)partition_table + 
num_parts * sizeof(esp_partition_info_t)), 
sizeof(esp_partition_info_t));
```

Dangerous Functions\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=156
Status	New

The dangerous function, memcpy, was found in use at line 4754 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination

File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4940	4940
Object	memcpy	memcpy

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method void mdns_debug_packet(const uint8_t * data, size_t len)

```
....
4940.                   memcpy(&ip6, data_ptr, MDNS_ANSWER_AAAA_SIZE);
```

Dangerous Functions\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=157
Status	New

The dangerous function, memcpy, was found in use at line 4754 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4944	4944
Object	memcpy	memcpy

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method void mdns_debug_packet(const uint8_t * data, size_t len)

```
....
4944.                   memcpy(&ip, data_ptr, sizeof(ip4_addr_t));
```

Dangerous Functions\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=158
Status	New

The dangerous function, memcpy, was found in use at line 176 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	204	204

Object	memcpy	memcpy

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static const uint8_t * _mdns_read_fqdn(const uint8_t * packet, const uint8_t * start, mdns_name_t * name, char * buf)

```
....
204.                 memcpy(mdns_name_ptrs[name->parts++], buf, len+1);
```

Dangerous Functions\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=159
Status	New

The dangerous function, memcpy, was found in use at line 347 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	354	354
Object	memcpy	memcpy

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static inline uint8_t _mdns_append_string(uint8_t * packet, uint16_t * index, const char * string)

```
....
354.        memcpy(packet + *index, string, len);
```

Dangerous Functions\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=160
Status	New

The dangerous function, memcpy, was found in use at line 729 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	762	762

Object	memcpy	memcpy

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static uint16_t _mdns_append_aaaa_record(uint8_t * packet, uint16_t * index, uint8_t * ipv6, bool flush, bool bye)

```
....
762.        memcpy(packet + *index, ipv6, part_length);
```

Dangerous Functions\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=161
Status	New

The dangerous function, memcpy, was found in use at line 1181 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1196	1196
Object	memcpy	memcpy

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_tx_packet_t * _mdns_alloc_packet_default(tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol)

```
....
1196.          memcpy(&packet->dst, &addr, sizeof(ip_addr_t));
```

Dangerous Functions\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=162
Status	New

The dangerous function, memcpy, was found in use at line 1204 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1278	1278

Object	memcpy	memcpy

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_create_answer_from_parsed_packet(mdns_parsed_packet_t * parsed_packet)

```
....
1278.           memcpy(&packet->dst, &parsed_packet->src,
sizeof(ip_addr_t));
```

Dangerous Functions\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=163
Status	New

The dangerous function, memcpy, was found in use at line 2038 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2063	2063
Object	memcpy	memcpy

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static int _mdns_check_srv_collision(mdns_service_t * service, uint16_t priority, uint16_t weight, uint16_t port, const char * host, const char * domain)

```
....
2063.       memcpy(our_data + our_index, _mdns_server->hostname,
our_host_len);
```

Dangerous Functions\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=164
Status	New

The dangerous function, memcpy, was found in use at line 2038 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c

Line	2066	2066
Object	memcpy	memcpy

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static int _mdns_check_srv_collision(mdns_service_t * service, uint16_t priority, uint16_t weight, uint16_t port, const char * host, const char * domain)

```
....
2066.       memcpy(our_data + our_index, MDNS_DEFAULT_DOMAIN, 5);
```

Dangerous Functions\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=165
Status	New

The dangerous function, memcpy, was found in use at line 2038 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2076	2076
Object	memcpy	memcpy

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static int _mdns_check_srv_collision(mdns_service_t * service, uint16_t priority, uint16_t weight, uint16_t port, const char * host, const char * domain)

```
....
2076.       memcpy(their_data + their_index, host, their_host_len);
```

Dangerous Functions\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=166
Status	New

The dangerous function, memcpy, was found in use at line 2038 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c

Line	2079	2079
Object	memcpy	memcpy

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static int _mdns_check_srv_collision(mdns_service_t * service, uint16_t priority, uint16_t weight, uint16_t port, const char * host, const char * domain)

```
....
2079.      memcpy(their_data + their_index, domain, their_domain_len);
```

Dangerous Functions\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=167
Status	New

The dangerous function, memcpy, was found in use at line 2473 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2519	2519
Object	memcpy	memcpy

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_result_txt_create(const uint8_t * data, size_t len, mdns_txt_item_t ** out_txt, size_t * out_count)

```
....
2519.          memcpy(key, data + i, name_len);
```

Dangerous Functions\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=168
Status	New

The dangerous function, memcpy, was found in use at line 2473 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c

Line	2531	2531
Object	memcpy	memcpy

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_result_txt_create(const uint8_t * data, size_t len, mdns_txt_item_t ** out_txt, size_t * out_count)

```
....
2531.                memcpy(value, data + i, value_len);
```

Dangerous Functions\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=169
Status	New

The dangerous function, memcpy, was found in use at line 2572 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2926	2926
Object	memcpy	memcpy

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
2926.                    memcpy(ip6.u_addr.ip6.addr, data_ptr, 
MDNS_ANSWER_AAAA_SIZE);
```

Dangerous Functions\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=170
Status	New

The dangerous function, memcpy, was found in use at line 2572 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c

Line	2972	2972
Object	memcpy	memcpy

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
2972.                   memcpy(&(ip.u_addr.ip4.addr), data_ptr, 4);
```

Dangerous Functions\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=171
Status	New

The dangerous function, memcpy, was found in use at line 3230 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	3240	3240
Object	memcpy	memcpy

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static mdns_ip_addr_t * _mdns_result_addr_create_ip(ip_addr_t * ip)

```
....
3240.           memcpy(a->addr.u_addr.ip6.addr, ip->u_addr.ip6.addr, 16);
```

Dangerous Functions\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=172
Status	New

The dangerous function, memcpy, was found in use at line 4719 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4741	4741
Object	memcpy	memcpy

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	esp_err_t mdns_query_aaaa(const char * name, uint32_t timeout, ip6_addr_t * addr)

```
....
4741.                memcpy(addr->addr, a->addr.u_addr.ip6.addr, 16);
```

Dangerous Functions\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=173
Status	New

The dangerous function, sprintf, was found in use at line 53 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	76	76
Object	sprintf	sprintf

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static char * _mdns_mangle_name(char* in) {

```
....
76.            sprintf(ret, "%s-2", in);
```

Dangerous Functions\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=174
Status	New

The dangerous function, sprintf, was found in use at line 53 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	86	86
Object	sprintf	sprintf

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static char * _mdns_mangle_name(char* in) {

```
....
86.         sprintf(ret + baseLen, "-%d", suffix + 1);
```

Dangerous Functions\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=175
Status	New

The dangerous function, sprintf, was found in use at line 539 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	578	578
Object	sprintf	sprintf

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static uint16_t _mdns_append_txt_record(uint8_t * packet, uint16_t * index, mdns_service_t * service, bool flush, bool bye)

```
....
578.              sprintf(tmp, "%s=%s", txt->key, txt->value);
```

Dangerous Functions\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=176
Status	New

The dangerous function, sprintf, was found in use at line 2095 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2126	2126
Object	sprintf	sprintf

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c

Method	static int _mdns_check_txt_collision(mdns_service_t * service, const uint8_t * data, size_t len)

```
....
2126.                 sprintf(tmp, "%s=%s", txt->key, txt->value);
```

Dangerous Functions\Path 31:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=177
Status	New

The dangerous function, strcpy, was found in use at line 53 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	83	83
Object	strcpy	strcpy

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static char * _mdns_mangle_name(char* in) {

```
....
83.             strcpy(ret, in);
```

Dangerous Functions\Path 32:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=178
Status	New

The dangerous function, strlen, was found in use at line 58 in ESP8266_RTOS_SDK/conf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	65	65
Object	strlen	strlen

Code Snippet

File Name	ESP8266_RTOS_SDK/conf.c
Method	static void strip(char *str)

```
....
65.    l = strlen(p);
```

Dangerous Functions\Path 33:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=179
Status	New

The dangerous function, strlen, was found in use at line 134 in ESP8266_RTOS_SDK/conf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	159	159
Object	strlen	strlen

Code Snippet
File Name ESP8266_RTOS_SDK/conf.c
Method static int conf_string(struct menu *menu)

```
....
159.                    line[strlen(line)-1] = 0;
```

Dangerous Functions\Path 34:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=180
Status	New

The dangerous function, strlen, was found in use at line 236 in ESP8266_RTOS_SDK/conf.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	341	341
Object	strlen	strlen

Code Snippet
File Name ESP8266_RTOS_SDK/conf.c
Method static int conf_choice(struct menu *menu)

```
....
341.                if (line[0] && line[strlen(line) - 1] == '?') {
```

Dangerous Functions\Path 35:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=181
Status	New

The dangerous function, strlen, was found in use at line 53 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	71	71
Object	strlen	strlen

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static char * _mdns_mangle_name(char* in) {

```
....
71.             ret = malloc(strlen(in) + 3);
```

Dangerous Functions\Path 36:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=182
Status	New

The dangerous function, strlen, was found in use at line 53 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	78	78
Object	strlen	strlen

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static char * _mdns_mangle_name(char* in) {

```
....
78.          ret = malloc(strlen(in) + 2); //one extra byte in case 9-10
or 99-100 etc
```

Dangerous Functions\Path 37:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=183
Status	New

The dangerous function, strlen, was found in use at line 347 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	349	349
Object	strlen	strlen

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static inline uint8_t _mdns_append_string(uint8_t * packet, uint16_t * index, const char * string)

```
....
349.      uint8_t len = strlen(string);
```

Dangerous Functions\Path 38:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=184
Status	New

The dangerous function, strlen, was found in use at line 370 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	378	378
Object	strlen	strlen

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static uint16_t _mdns_append_fqdn(uint8_t * packet, uint16_t * index, const char * strings[], uint8_t count)

```
....
378.        uint8_t len = strlen(strings[0]);
```

Dangerous Functions\Path 39:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=185
Status	New

The dangerous function, strlen, was found in use at line 539 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	576	576
Object	strlen	strlen

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static uint16_t _mdns_append_txt_record(uint8_t * packet, uint16_t * index, mdns_service_t * service, bool flush, bool bye)

```
....
576.            tmp = (char *)malloc(2 + strlen(txt->key) + strlen(txt-
>value));
```

Dangerous Functions\Path 40:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=186
Status	New

The dangerous function, strlen, was found in use at line 539 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	576	576
Object	strlen	strlen

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static uint16_t _mdns_append_txt_record(uint8_t * packet, uint16_t * index, mdns_service_t * service, bool flush, bool bye)

```
....
576.           tmp = (char *)malloc(2 + strlen(txt->key) + strlen(txt-
>value));
```

Dangerous Functions\Path 41:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=187
Status	New

The dangerous function, strlen, was found in use at line 2038 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2044	2044
Object	strlen	strlen

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static int _mdns_check_srv_collision(mdns_service_t * service, uint16_t priority, uint16_t weight, uint16_t port, const char * host, const char * domain)

```
....
2044.       size_t our_host_len = strlen(_mdns_server->hostname);
```

Dangerous Functions\Path 42:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=188
Status	New

The dangerous function, strlen, was found in use at line 2038 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2047	2047
Object	strlen	strlen

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static int _mdns_check_srv_collision(mdns_service_t * service, uint16_t priority, uint16_t weight, uint16_t port, const char * host, const char * domain)

```
....
2047.      size_t their_host_len = strlen(host);
```

Dangerous Functions\Path 43:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=189
Status	New

The dangerous function, strlen, was found in use at line 2038 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2048	2048
Object	strlen	strlen

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static int _mdns_check_srv_collision(mdns_service_t * service, uint16_t priority, uint16_t weight, uint16_t port, const char * host, const char * domain)

```
....
2048.      size_t their_domain_len = strlen(domain);
```

Dangerous Functions\Path 44:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=190
Status	New

The dangerous function, strlen, was found in use at line 2095 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2108	2108
Object	strlen	strlen

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static int _mdns_check_txt_collision(mdns_service_t * service, const uint8_t * data, size_t len)

```
....
2108.           data_len += 2 + strlen(service->txt->key) +
strlen(service->txt->value);
```

Dangerous Functions\Path 45:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=191
Status	New

The dangerous function, strlen, was found in use at line 2095 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2108	2108
Object	strlen	strlen

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static int _mdns_check_txt_collision(mdns_service_t * service, const uint8_t * data, size_t len)

```
....
2108.           data_len += 2 + strlen(service->txt->key) +
strlen(service->txt->value);
```

Dangerous Functions\Path 46:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=192
Status	New

The dangerous function, strlen, was found in use at line 2095 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2124	2124
Object	strlen	strlen

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static int _mdns_check_txt_collision(mdns_service_t * service, const uint8_t * data, size_t len)

```
....
2124.          tmp = (char *)malloc(2 + strlen(txt->key) + strlen(txt-
>value));
```

Dangerous Functions\Path 47:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=193
Status	New

The dangerous function, strlen, was found in use at line 2095 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2124	2124
Object	strlen	strlen

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static int _mdns_check_txt_collision(mdns_service_t * service, const uint8_t * data, size_t len)

```
....
2124.          tmp = (char *)malloc(2 + strlen(txt->key) + strlen(txt-
>value));
```

Dangerous Functions\Path 48:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=194
Status	New

The dangerous function, strlen, was found in use at line 4267 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4272	4272
Object	strlen	strlen

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	esp_err_t mdns_hostname_set(const char * hostname)

```
....
4272.       if (_str_null_or_empty(hostname) || strlen(hostname) >
(MDNS_NAME_BUF_LEN - 1)) {
```

Dangerous Functions\Path 49:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=195
Status	New

The dangerous function, strlen, was found in use at line 4296 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4301	4301
Object	strlen	strlen

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	esp_err_t mdns_instance_name_set(const char * instance)

```
....
4301.       if (_str_null_or_empty(instance) || strlen(instance) >
(MDNS_NAME_BUF_LEN - 1)) {
```

Dangerous Functions\Path 50:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=196
Status	New

The dangerous function, strlen, was found in use at line 4515 in ESP8266_RTOS_SDK/mdns.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4520	4520
Object	strlen	strlen

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c

Method esp_err_t mdns_service_instance_name_set(const char * service, const char * proto, const char * instance)

```
....
4520.        if (_str_null_or_empty(instance) || strlen(instance) >
(MDNS_NAME_BUF_LEN - 1)) {
```

Double Free

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

Categories

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

Description

Double Free\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=199
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1821	1842
Object	new_item	value

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1821.                     free(new_item);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_linked_txt(mdns_txt_linked_item_t *txt)

```
....
1842.             free((char *)t->value);
```

Double Free\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=200
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1827	1842
Object	new_item	value

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1827.                       free(new_item);
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_linked_txt(mdns_txt_linked_item_t *txt)

```
....
1842.               free((char *)t->value);
```

Double Free\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=201
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1821	1843
Object	new_item	key

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1821.                       free(new_item);
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_linked_txt(mdns_txt_linked_item_t *txt)

```
....
1843.           free((char *)t->key);
```

Double Free\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=202
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1827	1843
Object	new_item	key

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1827.                   free(new_item);
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_linked_txt(mdns_txt_linked_item_t *txt)

```
....
1843.           free((char *)t->key);
```

Double Free\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=203
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1821	1844
Object	new_item	t

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method	static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1821.                   free(new_item);
```

▼

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_free_linked_txt(mdns_txt_linked_item_t *txt)

```
....
1844.           free(t);
```

Double Free\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=204
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1827	1844
Object	new_item	t

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1827.                   free(new_item);
```

▼

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_free_linked_txt(mdns_txt_linked_item_t *txt)

```
....
1844.           free(t);
```

Double Free\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=205
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1821	1881
Object	new_item	s

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1821.                    free(new_item);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_service_t * _mdns_create_service(const char * service, const char * proto, uint16_t port, const char * instance, size_t num_items, mdns_txt_item_t txt[])

```
....
1881.            free(s);
```

Double Free\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=206
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1827	1881
Object	new_item	s

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1827.                    free(new_item);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method	static mdns_service_t * _mdns_create_service(const char * service, const char * proto, uint16_t port, const char * instance, size_t num_items, mdns_txt_item_t txt[])

```
....
1881.           free(s);
```

Double Free\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=207
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1821	1888
Object	new_item	s

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1821.                   free(new_item);
```

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static mdns_service_t * _mdns_create_service(const char * service, const char * proto, uint16_t port, const char * instance, size_t num_items, mdns_txt_item_t txt[])

```
....
1888.           free(s);
```

Double Free\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=208
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1827	1888

Object	new_item	s

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1827.                       free(new_item);
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_service_t * _mdns_create_service(const char * service, const char * proto, uint16_t port, const char * instance, size_t num_items, mdns_txt_item_t txt[])

```
....
1888.               free(s);
```

Double Free\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=209
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1821	2016
Object	new_item	instance

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1821.                       free(new_item);
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_service(mdns_service_t * service)

```
....
2016.         free((char *)service->instance);
```

Double Free\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=210
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1827	2016
Object	new_item	instance

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1827.                       free(new_item);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_service(mdns_service_t * service)

```
....
2016.         free((char *)service->instance);
```

Double Free\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=211
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1821	2017
Object	new_item	service

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1821.                       free(new_item);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_service(mdns_service_t * service)

```
....
2017.       free((char *)service->service);
```

Double Free\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=212
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1827	2017
Object	new_item	service

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1827.                   free(new_item);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_service(mdns_service_t * service)

```
....
2017.       free((char *)service->service);
```

Double Free\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=213
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1821	2018

Object	new_item	proto

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1821.                     free(new_item);
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_service(mdns_service_t * service)

```
....
2018.         free((char *)service->proto);
```

Double Free\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=214
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1827	2018
Object	new_item	proto

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1827.                     free(new_item);
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_service(mdns_service_t * service)

```
....
2018.         free((char *)service->proto);
```

Double Free\Path 17:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=215
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1821	2024
Object	new_item	s

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1821.                       free(new_item);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_service(mdns_service_t * service)

```
....
2024.           free(s);
```

Double Free\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=216
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1827	2024
Object	new_item	s

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1827.                       free(new_item);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_service(mdns_service_t * service)

```
....
2024.           free(s);
```

Double Free\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=217
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2027	3681
Object	service	service

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_free_service(mdns_service_t * service)

```
....
2027.       free(service);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_action(mdns_action_t * action)

```
....
3681.           free(action->data.srv_add.service);
```

Double Free\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=218
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2026	3681
Object	txt	service

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_service(mdns_service_t * service)

```
....
2026.       free(service->txt);
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_action(mdns_action_t * action)

```
....
3681.           free(action->data.srv_add.service);
```

Double Free\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=219
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2027	3844
Object	service	a

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_service(mdns_service_t * service)

```
....
2027.       free(service);
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3844.                       free(a);
```

Double Free\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=220
Status	New

	Source	Destination

File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2026	3844
Object	txt	a

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_free_service(mdns_service_t * service)

```
....
2026.       free(service->txt);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3844.                   free(a);
```

Double Free\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=221
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2027	3855
Object	service	b

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_free_service(mdns_service_t * service)

```
....
2027.       free(service);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3855.                      free(b);
```

Double Free\Path 24:

Severity	Medium

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=222
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2026	3855
Object	txt	b

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_service(mdns_service_t * service)

```
....
2026.       free(service->txt);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3855.                       free(b);
```

Double Free\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=223
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2027	3870
Object	service	s

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_service(mdns_service_t * service)

```
....
2027.       free(service);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3870.                 free(s);
```

Double Free\Path 26:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=224
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2026	3870
Object	txt	s

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_service(mdns_service_t * service)

```
....
2026.         free(service->txt);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3870.                 free(s);
```

Double Free\Path 27:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=225
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1821	4363
Object	new_item	item

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1821.                       free(new_item);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method esp_err_t mdns_service_add(const char * instance, const char * service, const char * proto, uint16_t port, mdns_txt_item_t txt[], size_t num_items)

```
....
4363.              free(item);
```

Double Free\Path 28:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=226
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1827	4363
Object	new_item	item

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1827.                       free(new_item);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method esp_err_t mdns_service_add(const char * instance, const char * service, const char * proto, uint16_t port, mdns_txt_item_t txt[], size_t num_items)

```
....
4363.              free(item);
```

Double Free\Path 29:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=227

Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1821	4370
Object	new_item	item

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1821.                       free(new_item);
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method esp_err_t mdns_service_add(const char * instance, const char * service, const char * proto, uint16_t port, mdns_txt_item_t txt[], size_t num_items)

```
....
4370.               free(item);
```

Double Free\Path 30:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=228
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1827	4370
Object	new_item	item

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1827.                       free(new_item);
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method	esp_err_t mdns_service_add(const char * instance, const char * service, const char * proto, uint16_t port, mdns_txt_item_t txt[], size_t num_items)

```
....
4370.           free(item);
```

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=1050073&projectid=50063&pathid=18
Status	New

The size of the buffer used by crypto_aead_aes256gcm_encrypt_detached_afternm in x, at line 503 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that crypto_aead_aes256gcm_encrypt_detached_afternm passes to x, at line 503 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	535	535
Object	x	x

Code Snippet	
File Name	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method	crypto_aead_aes256gcm_encrypt_detached_afternm(unsigned char *c,

```
....
535.          memcpy(&fb[0], &x, sizeof x);
```

Buffer Overflow boundcpy WrongSizeParam\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=19
Status	New

The size of the buffer used by crypto_aead_aes256gcm_encrypt_detached_afternm in x, at line 503 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c, is not properly verified before writing data to the buffer.

This can enable a buffer overflow attack, using the source buffer that crypto_aead_aes256gcm_encrypt_detached_afternm passes to x, at line 503 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	537	537
Object	x	x

Code Snippet
File Name ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method crypto_aead_aes256gcm_encrypt_detached_afternm(unsigned char *c,

```
....
537.          memcpy(&fb[8], &x, sizeof x);
```

Buffer Overflow boundcpy WrongSizeParam\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=20
Status	New

The size of the buffer used by crypto_aead_aes256gcm_decrypt_detached_afternm in x, at line 642 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that crypto_aead_aes256gcm_decrypt_detached_afternm passes to x, at line 642 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	676	676
Object	x	x

Code Snippet
File Name ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method crypto_aead_aes256gcm_decrypt_detached_afternm(unsigned char *m, unsigned char *nsec,

```
....
676.          memcpy(&fb[0], &x, sizeof x);
```

Buffer Overflow boundcpy WrongSizeParam\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=21

Status	New

The size of the buffer used by crypto_aead_aes256gcm_decrypt_detached_afternm in x, at line 642 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that crypto_aead_aes256gcm_decrypt_detached_afternm passes to x, at line 642 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	678	678
Object	x	x

Code Snippet
File Name ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method crypto_aead_aes256gcm_decrypt_detached_afternm(unsigned char *m, unsigned char *nsec,

```
....
678.            memcpy(&fb[8], &x, sizeof x);
```

Buffer Overflow boundcpy WrongSizeParam\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=22
Status	New

The size of the buffer used by esp_partition_table_basic_verify in esp_partition_info_t, at line 108 of ESP8266_RTOS_SDK/flash_partitions.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that esp_partition_table_basic_verify passes to esp_partition_info_t, at line 108 of ESP8266_RTOS_SDK/flash_partitions.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/flash_partitions.c	ESP8266_RTOS_SDK/flash_partitions.c
Line	120	120
Object	esp_partition_info_t	esp_partition_info_t

Code Snippet
File Name ESP8266_RTOS_SDK/flash_partitions.c
Method esp_err_t esp_partition_table_basic_verify(const esp_partition_info_t *partition_table, bool log_errors, int *num_partitions)

```
....
120.            memcpy(&part_local, (void *)((intptr_t)partition_table + 
num_parts * sizeof(esp_partition_info_t)), 
sizeof(esp_partition_info_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=23
Status	New

The size of the buffer used by mdns_debug_packet in ip4_addr_t, at line 4754 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that mdns_debug_packet passes to ip4_addr_t, at line 4754 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4944	4944
Object	ip4_addr_t	ip4_addr_t

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method void mdns_debug_packet(const uint8_t * data, size_t len)

```
....
4944.                    memcpy(&ip, data_ptr, sizeof(ip4_addr_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=24
Status	New

The size of the buffer used by _mdns_alloc_packet_default in ip_addr_t, at line 1181 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that _mdns_alloc_packet_default passes to ip_addr_t, at line 1181 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1196	1196
Object	ip_addr_t	ip_addr_t

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static mdns_tx_packet_t * _mdns_alloc_packet_default(tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol)

```
....
1196.            memcpy(&packet->dst, &addr, sizeof(ip_addr_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=25
Status	New

The size of the buffer used by _mdns_create_answer_from_parsed_packet in ip_addr_t, at line 1204 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that _mdns_create_answer_from_parsed_packet passes to ip_addr_t, at line 1204 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1278	1278
Object	ip_addr_t	ip_addr_t

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_create_answer_from_parsed_packet(mdns_parsed_packet_t * parsed_packet)

```
....
1278.           memcpy(&packet->dst, &parsed_packet->src,
sizeof(ip_addr_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=26
Status	New

The size of the buffer used by mdns_debug_packet in mdns_name_t, at line 4754 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that mdns_debug_packet passes to mdns_name_t, at line 4754 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4761	4761
Object	mdns_name_t	mdns_name_t

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method void mdns_debug_packet(const uint8_t * data, size_t len)

```
....
4761.       memset(name, 0, sizeof(mdns_name_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=27
Status	New

The size of the buffer used by _mdns_alloc_packet_default in mdns_tx_packet_t, at line 1181 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that _mdns_alloc_packet_default passes to mdns_tx_packet_t, at line 1181 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1188	1188
Object	mdns_tx_packet_t	mdns_tx_packet_t

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static mdns_tx_packet_t * _mdns_alloc_packet_default(tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol)

```
....
1188.      memset((uint8_t*)packet, 0, sizeof(mdns_tx_packet_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=28
Status	New

The size of the buffer used by mdns_parse_packet in mdns_parsed_packet_t, at line 2572 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that mdns_parse_packet passes to mdns_parsed_packet_t, at line 2572 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2597	2597
Object	mdns_parsed_packet_t	mdns_parsed_packet_t

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
2597.      memset(parsed_packet, 0, sizeof(mdns_parsed_packet_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=29
Status	New

The size of the buffer used by mdns_parse_packet in mdns_name_t, at line 2572 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that mdns_parse_packet passes to mdns_name_t, at line 2572 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2600	2600
Object	mdns_name_t	mdns_name_t

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
2600.       memset(name, 0, sizeof(mdns_name_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=30
Status	New

The size of the buffer used by _mdns_search_init in mdns_search_once_t, at line 3140 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that _mdns_search_init passes to mdns_search_once_t, at line 3140 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	3147	3147
Object	mdns_search_once_t	mdns_search_once_t

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static mdns_search_once_t * _mdns_search_init(const char * name, const char * service, const char * proto, uint16_t type, uint32_t timeout, uint8_t max_results)

```
....
3147.       memset(search, 0, sizeof(mdns_search_once_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=31
Status	New

The size of the buffer used by _mdns_result_addr_create_ip in mdns_ip_addr_t, at line 3230 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that _mdns_result_addr_create_ip passes to mdns_ip_addr_t, at line 3230 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	3237	3237
Object	mdns_ip_addr_t	mdns_ip_addr_t

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static mdns_ip_addr_t * _mdns_result_addr_create_ip(ip_addr_t * ip)

```
....
3237.      memset(a, 0 , sizeof(mdns_ip_addr_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=32
Status	New

The size of the buffer used by _mdns_search_result_add_ip in mdns_result_t, at line 3275 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that _mdns_search_result_add_ip passes to mdns_result_t, at line 3275 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	3298	3298
Object	mdns_result_t	mdns_result_t

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_search_result_add_ip(mdns_search_once_t * search, const char * hostname, ip_addr_t * ip, tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol)

```
....
3298.              memset(r, 0 , sizeof(mdns_result_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=33
Status	New

The size of the buffer used by _mdns_search_result_add_srv in mdns_result_t, at line 3364 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that _mdns_search_result_add_srv passes to mdns_result_t, at line 3364 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	3380	3380
Object	mdns_result_t	mdns_result_t

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_search_result_add_srv(mdns_search_once_t * search, const char * hostname, uint16_t port, tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol)

```
....
3380.            memset(r, 0 , sizeof(mdns_result_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=34
Status	New

The size of the buffer used by _mdns_search_result_add_txt in mdns_result_t, at line 3398 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that _mdns_search_result_add_txt passes to mdns_result_t, at line 3398 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	3420	3420
Object	mdns_result_t	mdns_result_t

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_search_result_add_txt(mdns_search_once_t * search, mdns_txt_item_t * txt, size_t txt_count, tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol)

```
....
3420.          memset(r, 0 , sizeof(mdns_result_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=35
Status	New

The size of the buffer used by mdns_init in mdns_server_t, at line 4151 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that mdns_init passes to mdns_server_t, at line 4151 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4164	4164
Object	mdns_server_t	mdns_server_t

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	esp_err_t mdns_init()

```
....
4164.      memset((uint8_t*)_mdns_server, 0, sizeof(mdns_server_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=36
Status	New

The size of the buffer used by _mdns_check_a_collision in ip4_addr_t, at line 2168 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that _mdns_check_a_collision passes to ip4_addr_t, at line 2168 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2179	2179
Object	ip4_addr_t	ip4_addr_t

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static int _mdns_check_a_collision(ip4_addr_t * ip, tcpip_adapter_if_t tcpip_if)

```
....
2179.       int ret = memcmp((uint8_t*)&if_ip_info.ip.addr,
(uint8_t*)&ip->addr, sizeof(ip4_addr_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=37
Status	New

The size of the buffer used by _mdns_check_srv_collision in our_host_len, at line 2038 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that _mdns_check_srv_collision passes to our_host_len, at line 2038 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2063	2063
Object	our_host_len	our_host_len

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static int _mdns_check_srv_collision(mdns_service_t * service, uint16_t priority, uint16_t weight, uint16_t port, const char * host, const char * domain)

```
....
2063.       memcpy(our_data + our_index, _mdns_server->hostname,
our_host_len);
```

Buffer Overflow boundcpy WrongSizeParam\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=38
Status	New

The size of the buffer used by _mdns_check_srv_collision in their_host_len, at line 2038 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that _mdns_check_srv_collision passes to their_host_len, at line 2038 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2076	2076
Object	their_host_len	their_host_len

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static int _mdns_check_srv_collision(mdns_service_t * service, uint16_t priority, uint16_t weight, uint16_t port, const char * host, const char * domain)

```
....
2076.       memcpy(their_data + their_index, host, their_host_len);
```

Buffer Overflow boundcpy WrongSizeParam\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=39
Status	New

The size of the buffer used by _mdns_check_srv_collision in their_domain_len, at line 2038 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that _mdns_check_srv_collision passes to their_domain_len, at line 2038 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2079	2079
Object	their_domain_len	their_domain_len

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static int _mdns_check_srv_collision(mdns_service_t * service, uint16_t priority, uint16_t weight, uint16_t port, const char * host, const char * domain)

```
....
2079.       memcpy(their_data + their_index, domain, their_domain_len);
```

Buffer Overflow boundcpy WrongSizeParam\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=40
Status	New

The size of the buffer used by crypto_aead_aes256gcm_decrypt_detached_afternm in mlen, at line 642 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that crypto_aead_aes256gcm_decrypt_detached_afternm passes to mlen, at line 642 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	782	782

Object	mlen	mlen

Code Snippet
File Name ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method crypto_aead_aes256gcm_decrypt_detached_afternm(unsigned char *m, unsigned char *nsec,

```
....
782.                    memset(m, 0, mlen);
```

Buffer Overflow boundcpy WrongSizeParam\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=41
Status	New

The size of the buffer used by _mdns_check_srv_collision in our_len, at line 2038 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that _mdns_check_srv_collision passes to our_len, at line 2038 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2083	2083
Object	our_len	our_len

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static int _mdns_check_srv_collision(mdns_service_t * service, uint16_t priority, uint16_t weight, uint16_t port, const char * host, const char * domain)

```
....
2083.       int ret = memcmp(our_data, their_data, our_len);
```

Buffer Overflow boundcpy WrongSizeParam\Path 25:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=42
Status	New

The size of the buffer used by _mdns_check_txt_collision in len, at line 2095 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that _mdns_check_txt_collision passes to len, at line 2095 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c

Line	2136	2136
Object	len	len

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static int _mdns_check_txt_collision(mdns_service_t * service, const uint8_t * data, size_t len)

```
....
2136.       int ret = memcmp(ours, data, len);
```

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=1050073&projectid=50063&pathid=43
Status	New

Calling free() (line 1092) on a variable that was not dynamically allocated (line 1092) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1104	1104
Object	a	a

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_remove_scheduled_answer(tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol, uint16_t type, mdns_srv_item_t * service)

```
....
1104.                   free(a);
```

MemoryFree on StackVariable\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=44
Status	New

Calling free() (line 1092) on a variable that was not dynamically allocated (line 1092) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1110	1110
Object	b	b

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_remove_scheduled_answer(tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol, uint16_t type, mdns_srv_item_t * service)

```
....
1110.                              free(b);
```

MemoryFree on StackVariable\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=45
Status	New

Calling free() (line 1124) on a variable that was not dynamically allocated (line 1124) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1136	1136
Object	d	d

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_dealloc_answer(mdns_out_answer_t ** destnation, uint16_t type, mdns_srv_item_t * service)

```
....
1136.          free(d);
```

MemoryFree on StackVariable\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=46
Status	New

Calling free() (line 1124) on a variable that was not dynamically allocated (line 1124) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1143	1143
Object	a	a

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_dealloc_answer(mdns_out_answer_t ** destnation, uint16_t type, mdns_srv_item_t * service)

```
....
1143.                    free(a);
```

MemoryFree on StackVariable\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=47
Status	New

Calling free() (line 1898) on a variable that was not dynamically allocated (line 1898) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1906	1906
Object	d	d

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_dealloc_scheduled_service_answers(mdns_out_answer_t ** destination, mdns_service_t * service)

```
....
1906.                free(d);
```

MemoryFree on StackVariable\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=48
Status	New

Calling free() (line 1898) on a variable that was not dynamically allocated (line 1898) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1913	1913
Object	a	a

Code Snippet	
File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_dealloc_scheduled_service_answers(mdns_out_answer_t ** destination, mdns_service_t * service)

```
....
1913.                 free(a);
```

MemoryFree on StackVariable\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=49
Status	New

Calling free() (line 1923) on a variable that was not dynamically allocated (line 1923) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1974	1974
Object	qs	qs

Code Snippet	
File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_remove_scheduled_service_packets(mdns_service_t * service)

```
....
1974.                                 free(qs);
```

MemoryFree on StackVariable\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=50
Status	New

Calling free() (line 1923) on a variable that was not dynamically allocated (line 1923) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination

File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1982	1982
Object	qsn	qsn

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_remove_scheduled_service_packets(mdns_service_t * service)

```
....
1982.                                       free(qsn);
```

MemoryFree on StackVariable\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=51
Status	New

Calling free() (line 2397) on a variable that was not dynamically allocated (line 2397) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2419	2419
Object	p	p

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_remove_parsed_question(mdns_parsed_packet_t * parsed_packet, uint16_t type, mdns_srv_item_t * service)

```
....
2419.                   free(p);
```

MemoryFree on StackVariable\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=52
Status	New

Calling free() (line 3719) on a variable that was not dynamically allocated (line 3719) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c

Line	3783	3783
Object	key	key

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3783.                       free(key);
```

MemoryFree on StackVariable\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=53
Status	New

Calling free() (line 3719) on a variable that was not dynamically allocated (line 3719) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	3824	3824
Object	t	t

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3824.                           free(t);
```

MemoryFree on StackVariable\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=54
Status	New

Calling free() (line 3719) on a variable that was not dynamically allocated (line 3719) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	3831	3831
Object	key	key

Code Snippet	
File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_execute_action(mdns_action_t * action)

```
....
3831.            free(key);
```

MemoryFree on StackVariable\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=55
Status	New

Calling free() (line 3719) on a variable that was not dynamically allocated (line 3719) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	3844	3844
Object	a	a

Code Snippet	
File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_execute_action(mdns_action_t * action)

```
....
3844.                    free(a);
```

MemoryFree on StackVariable\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=56
Status	New

Calling free() (line 3719) on a variable that was not dynamically allocated (line 3719) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	3855	3855
Object	b	b

Code Snippet	
File Name	ESP8266_RTOS_SDK/mdns.c

Method	static void _mdns_execute_action(mdns_action_t * action)

```
....
3855.                         free(b);
```

MemoryFree on StackVariable\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=57
Status	New

Calling free() (line 3719) on a variable that was not dynamically allocated (line 3719) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	3870	3870
Object	s	s

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_execute_action(mdns_action_t * action)

```
....
3870.                 free(s);
```

MemoryFree on StackVariable\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=58
Status	New

Calling free() (line 4223) on a variable that was not dynamically allocated (line 4223) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4260	4260
Object	h	h

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	void mdns_free()

```
....
4260.           free(h);
```

MemoryFree on StackVariable\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=59
Status	New

Calling free() (line 4267) on a variable that was not dynamically allocated (line 4267) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4283	4283
Object	new_hostname	new_hostname

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	esp_err_t mdns_hostname_set(const char * hostname)

```
....
4283.           free(new_hostname);
```

MemoryFree on StackVariable\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=60
Status	New

Calling free() (line 4267) on a variable that was not dynamically allocated (line 4267) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4289	4289
Object	new_hostname	new_hostname

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	esp_err_t mdns_hostname_set(const char * hostname)

```
....
4289.           free(new_hostname);
```

MemoryFree on StackVariable\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=61
Status	New

Calling free() (line 4296) on a variable that was not dynamically allocated (line 4296) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4312	4312
Object	new_instance	new_instance

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method esp_err_t mdns_instance_name_set(const char * instance)

```
....
4312.           free(new_instance);
```

MemoryFree on StackVariable\Path 20:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=62
Status	New

Calling free() (line 4296) on a variable that was not dynamically allocated (line 4296) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4318	4318
Object	new_instance	new_instance

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method esp_err_t mdns_instance_name_set(const char * instance)

```
....
4318.           free(new_instance);
```

MemoryFree on StackVariable\Path 21:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=63
Status	New

Calling free() (line 4515) on a variable that was not dynamically allocated (line 4515) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4535	4535
Object	new_instance	new_instance

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method esp_err_t mdns_service_instance_name_set(const char * service, const char * proto, const char * instance)

```
....
4535.           free(new_instance);
```

MemoryFree on StackVariable\Path 22:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=64
Status	New

Calling free() (line 4515) on a variable that was not dynamically allocated (line 4515) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4542	4542
Object	new_instance	new_instance

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method esp_err_t mdns_service_instance_name_set(const char * service, const char * proto, const char * instance)

```
....
4542.           free(new_instance);
```

MemoryFree on StackVariable\Path 23:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=65
Status	New

Calling free() (line 4599) on a variable that was not dynamically allocated (line 4599) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4620	4620
Object	a	a

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	void mdns_query_results_free(mdns_result_t * results)

```
....
4620.                   free(a);
```

MemoryFree on StackVariable\Path 24:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=66
Status	New

Calling free() (line 4599) on a variable that was not dynamically allocated (line 4599) in file ESP8266_RTOS_SDK/mdns.c may result with a crash.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4624	4624
Object	r	r

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	void mdns_query_results_free(mdns_result_t * results)

```
....
4624.           free(r);
```

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=1050073&projectid=50063&pathid=229
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1163	1163
Object	a	a

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static bool _mdns_alloc_answer(mdns_out_answer_t ** destnation, uint16_t type, mdns_service_t * service, bool flush, bool bye)

```
....
1163.      mdns_out_answer_t * a = (mdns_out_answer_t
*)malloc(sizeof(mdns_out_answer_t));
```

Memory Leak\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=230
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2511	2511
Object	key	key

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c

Method	static void _mdns_result_txt_create(const uint8_t * data, size_t len, mdns_txt_item_t ** out_txt, size_t * out_count)

```
....
2511.          char * key = (char *)malloc(name_len + 1);
```

Memory Leak\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=231
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2526	2526
Object	value	value

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_result_txt_create(const uint8_t * data, size_t len, mdns_txt_item_t ** out_txt, size_t * out_count)

```
....
2526.              char * value = (char *)malloc(value_len + 1);
```

Memory Leak\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=232
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	3232	3232
Object	a	a

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static mdns_ip_addr_t * _mdns_result_addr_create_ip(ip_addr_t * ip)

```
....
3232.      mdns_ip_addr_t * a = (mdns_ip_addr_t 
*)malloc(sizeof(mdns_ip_addr_t));
```

Memory Leak\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=233
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	3520	3520
Object	q	q

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_tx_packet_t * _mdns_create_search_packet(mdns_search_once_t * search, tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol)

```
....
3520.       mdns_out_question_t * q = (mdns_out_question_t
*)malloc(sizeof(mdns_out_question_t));
```

Memory Leak\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=234
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	3543	3543
Object	a	a

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_tx_packet_t * _mdns_create_search_packet(mdns_search_once_t * search, tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol)

```
....
3543.               mdns_out_answer_t * a = (mdns_out_answer_t
*)malloc(sizeof(mdns_out_answer_t));
```

Memory Leak\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=500

	63&pathid=235
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2808	2808
Object	hostname	hostname

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
2808.                                   result->hostname = strdup(name-
>host);
```

Memory Leak\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=236
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	3345	3345
Object	instance_name	instance_name

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static mdns_result_t * _mdns_search_result_add_ptr(mdns_search_once_t * search, const char * instance, tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol)

```
....
3345.           r->instance_name = strdup(instance);
```

Memory Leak\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=237
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c

Line	3381	3381
Object	hostname	hostname

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_search_result_add_srv(mdns_search_once_t * search, const char * hostname, uint16_t port, tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol)

```
....
3381.          r->hostname = strdup(hostname);
```

Memory Leak\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=238
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	3414	3414
Object	r	r

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_search_result_add_txt(mdns_search_once_t * search, mdns_txt_item_t * txt, size_t txt_count, tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol)

```
....
3414.          r = (mdns_result_t *)malloc(sizeof(mdns_result_t));
```

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=1050073&projectid=50063&pathid=239
Status	New

The variable declared in endp at ESP8266_RTOS_SDK/mdns.c in line 53 is not initialized when it is used by _mdns_server at ESP8266_RTOS_SDK/mdns.c in line 2572.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	60	3003
Object	endp	_mdns_server

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static char * _mdns_mangle_name(char* in) {

```
....
60.             char *endp = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c

Method void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
3003.                                           _mdns_server->hostname =
new_host;
```

Use of Zero Initialized Pointer\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=240
Status	New

The variable declared in endp at ESP8266_RTOS_SDK/mdns.c in line 53 is not initialized when it is used by _mdns_server at ESP8266_RTOS_SDK/mdns.c in line 2572.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	60	2957
Object	endp	_mdns_server

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static char * _mdns_mangle_name(char* in) {

```
....
60.             char *endp = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c

Method	void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
2957.                                    _mdns_server->hostname =
new_host;
```

Use of Zero Initialized Pointer\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=241
Status	New

The variable declared in new_txt at ESP8266_RTOS_SDK/mdns.c in line 1808 is not initialized when it is used by new_txt at ESP8266_RTOS_SDK/mdns.c in line 1808.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1810	1831
Object	new_txt	new_txt

Code Snippet	
File Name	ESP8266_RTOS_SDK/mdns.c
Method	static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1810.      mdns_txt_linked_item_t * new_txt = NULL;
....
1831.              new_txt = new_item;
```

Use of Zero Initialized Pointer\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=242
Status	New

The variable declared in new_txt at ESP8266_RTOS_SDK/mdns.c in line 1808 is not initialized when it is used by new_txt at ESP8266_RTOS_SDK/mdns.c in line 4411.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1810	4423
Object	new_txt	new_txt

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1810.       mdns_txt_linked_item_t * new_txt = NULL;
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method esp_err_t mdns_service_txt_set(const char * service, const char * proto, mdns_txt_item_t txt[], uint8_t num_items)

```
....
4423.           new_txt = _mdns_allocate_txt(num_items, txt);
```

Use of Zero Initialized Pointer\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=243
Status	New

The variable declared in new_txt at ESP8266_RTOS_SDK/mdns.c in line 1808 is not initialized when it is used by txt at ESP8266_RTOS_SDK/mdns.c in line 1859.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1810	1876
Object	new_txt	txt

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_txt_linked_item_t * _mdns_allocate_txt(size_t num_items, mdns_txt_item_t txt[])

```
....
1810.       mdns_txt_linked_item_t * new_txt = NULL;
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static mdns_service_t * _mdns_create_service(const char * service, const char * proto, uint16_t port, const char * instance, size_t num_items, mdns_txt_item_t txt[])

```
....
1876.       s->txt = new_txt;
```

Use of Zero Initialized Pointer\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=244
Status	New

The variable declared in service at ESP8266_RTOS_SDK/mdns.c in line 2572 is not initialized when it is used by service at ESP8266_RTOS_SDK/mdns.c in line 1153.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2728	1169
Object	service	service

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
2728.             mdns_srv_item_t * service = NULL;
```

▼

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static bool _mdns_alloc_answer(mdns_out_answer_t ** destnation, uint16_t type, mdns_service_t * service, bool flush, bool bye)

```
....
1169.       a->service = service;
```

Use of Zero Initialized Pointer\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=245
Status	New

The variable declared in BinaryExpr at ESP8266_RTOS_SDK/mdns.c in line 1479 is not initialized when it is used by service at ESP8266_RTOS_SDK/mdns.c in line 1153.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1487	1169
Object	BinaryExpr	service

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c

Method	static void _mdns_init_pcb_probe_new_service(tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol, mdns_srv_item_t ** services, size_t len, bool probe_ip)

```
....
1487.       mdns_srv_item_t ** _services = NULL;
```

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static bool _mdns_alloc_answer(mdns_out_answer_t ** destnation, uint16_t type, mdns_service_t * service, bool flush, bool bye)

```
....
1169.       a->service = service;
```

Use of Zero Initialized Pointer\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=246
Status	New

The variable declared in BinaryExpr at ESP8266_RTOS_SDK/mdns.c in line 1479 is not initialized when it is used by probe_services at ESP8266_RTOS_SDK/mdns.c in line 1479.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1487	1521
Object	BinaryExpr	probe_services

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_init_pcb_probe_new_service(tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol, mdns_srv_item_t ** services, size_t len, bool probe_ip)

```
....
1487.       mdns_srv_item_t ** _services = NULL;
....
1521.       pcb->probe_services = _services;
```

Use of Zero Initialized Pointer\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=247
Status	New

The variable declared in service at ESP8266_RTOS_SDK/mdns.c in line 2572 is not initialized when it is used by probe_services at ESP8266_RTOS_SDK/mdns.c in line 1479.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2728	1521
Object	service	probe_services

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
2728.               mdns_srv_item_t * service = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_init_pcb_probe_new_service(tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol, mdns_srv_item_t ** services, size_t len, bool probe_ip)

```
....
1521.       pcb->probe_services = _services;
```

Use of Zero Initialized Pointer\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=248
Status	New

The variable declared in _mdns_server at ESP8266_RTOS_SDK/mdns.c in line 4223 is not initialized when it is used by _mdns_server at ESP8266_RTOS_SDK/mdns.c in line 4223.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4264	4262
Object	_mdns_server	_mdns_server

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method void mdns_free()

```
....
4264.       _mdns_server = NULL;
....
4262.       vSemaphoreDelete(_mdns_server->lock);
```

Char Overflow

Query Path:
CPP\Cx\CPP Integer Overflow\Char Overflow Version:1

Categories

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

Description

Char Overflow\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=134
Status	New

A variable of a larger data type, out, is being assigned to a smaller data type, in 503 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	608	608
Object	out	out

Code Snippet
File Name ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method crypto_aead_aes256gcm_encrypt_detached_afternm(unsigned char *c,

```
....
608.        LOOPRND128;
```

Char Overflow\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=135
Status	New

A variable of a larger data type, in, is being assigned to a smaller data type, in 503 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	608	608
Object	in	in

Code Snippet	
File Name	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method	crypto_aead_aes256gcm_encrypt_detached_afternm(unsigned char *c,

```
....
608.        LOOPRND128;
```

Char Overflow\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=136
Status	New

A variable of a larger data type, in, is being assigned to a smaller data type, in 642 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	771	771
Object	in	in

Code Snippet	
File Name	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method	crypto_aead_aes256gcm_decrypt_detached_afternm(unsigned char *m, unsigned char *nsec,

```
....
771.        LOOPACCUMDRND128;
```

Char Overflow\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=137
Status	New

A variable of a larger data type, out, is being assigned to a smaller data type, in 642 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	792	792
Object	out	out

Code Snippet	
File Name	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method	crypto_aead_aes256gcm_decrypt_detached_afternm(unsigned char *m, unsigned char *nsec,

```
....
792.       LOOPDRND128;
```

Char Overflow\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=138
Status	New

A variable of a larger data type, in, is being assigned to a smaller data type, in 642 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	792	792
Object	in	in

Code Snippet	
File Name	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method	crypto_aead_aes256gcm_decrypt_detached_afternm(unsigned char *m, unsigned char *nsec,

```
....
792.       LOOPDRND128;
```

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=1050073&projectid=50063&pathid=139
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 503 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	562	562
Object	AssignExpr	AssignExpr

Code Snippet
File Name ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method crypto_aead_aes256gcm_encrypt_detached_afternm(unsigned char *c,

```
....
562.            blocklen = (unsigned int) (adlen - i);
```

Integer Overflow\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=140
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 642 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	701	701
Object	AssignExpr	AssignExpr

Code Snippet
File Name ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method crypto_aead_aes256gcm_decrypt_detached_afternm(unsigned char *m, unsigned char *nsec,

```
....
701.            blocklen = (unsigned int) (adlen - i);
```

Integer Overflow\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=141
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 642 of ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	772	772
Object	AssignExpr	AssignExpr

Code Snippet

File Name	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method	crypto_aead_aes256gcm_decrypt_detached_afternm(unsigned char *m, unsigned char *nsec,

```
....
772.       LOOPACCUMDRMD128;
```

Integer Overflow\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=142
Status	New

A variable of a larger data type, value_len, is being assigned to a smaller data type, in 2473 of ESP8266_RTOS_SDK/mdns.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2524	2524
Object	value_len	value_len

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_result_txt_create(const uint8_t * data, size_t len, mdns_txt_item_t ** out_txt, size_t * out_count)

```
....
2524.          int value_len = partLen - name_len - 1;
```

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=1050073&projectid=50063&pathid=68
Status	New

The function services_final_len in ESP8266_RTOS_SDK/mdns.c at line 1479 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	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1489	1489
Object	services_final_len	services_final_len

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_init_pcb_probe_new_service(tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol, mdns_srv_item_t ** services, size_t len, bool probe_ip)

```
....
1489.           _services = (mdns_srv_item_t
**)malloc(sizeof(mdns_srv_item_t *) * services_final_len);
```

Stored Buffer Overflow fgets

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

Categories

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

Description

Stored Buffer Overflow fgets\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=249
Status	New

The size of the buffer used by xfgets in size, at line 719 of ESP8266_RTOS_SDK/conf.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that xfgets passes to str, at line 719 of ESP8266_RTOS_SDK/conf.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	721	721
Object	str	size

Code Snippet

File Name	ESP8266_RTOS_SDK/conf.c
Method	void xfgets(char *str, int size, FILE *in)

```
....
721.        if (fgets(str, size, in) == 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=1050073&projectid=50063&pathid=69
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 2572 is not initialized when it is used by service at ESP8266_RTOS_SDK/mdns.c in line 2368.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2981	2376
Object	null	service

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
2981.
_mdns_remove_parsed_question(parsed_packet, type, NULL);
```

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static bool _mdns_question_matches(mdns_parsed_question_t * question, uint16_t type, mdns_srv_item_t * service)

```
....
2376.        if (!strcasecmp(service->service->service, question-
>service)
```

NULL Pointer Dereference\Path 2:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=70
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 2572 is not initialized when it is used by service at ESP8266_RTOS_SDK/mdns.c in line 2368.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2935	2376
Object	null	service

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
2935.
_mdns_remove_parsed_question(parsed_packet, type, NULL);
```

▼

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static bool _mdns_question_matches(mdns_parsed_question_t * question, uint16_t type, mdns_srv_item_t * service)

```
....
2376.          if (!strcasecmp(service->service->service, question-
>service)
```

NULL Pointer Dereference\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=71
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 2572 is not initialized when it is used by service at ESP8266_RTOS_SDK/mdns.c in line 2368.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2728	2376
Object	null	service

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c

Method void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
2728.               mdns_srv_item_t * service = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static bool _mdns_question_matches(mdns_parsed_question_t * question, uint16_t type, mdns_srv_item_t * service)

```
....
2376.           if (!strcasecmp(service->service->service, question-
>service)
```

NULL Pointer Dereference\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=72
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 2572 is not initialized when it is used by service at ESP8266_RTOS_SDK/mdns.c in line 2368.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2935	2377
Object	null	service

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
2935.
_mdns_remove_parsed_question(parsed_packet, type, NULL);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static bool _mdns_question_matches(mdns_parsed_question_t * question, uint16_t type, mdns_srv_item_t * service)

```
....
2377.               && !strcasecmp(service->service->proto, question-
>proto)
```

NULL Pointer Dereference\Path 5:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=73
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 2572 is not initialized when it is used by service at ESP8266_RTOS_SDK/mdns.c in line 2368.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2981	2377
Object	null	service

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
2981.
_mdns_remove_parsed_question(parsed_packet, type, NULL);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static bool _mdns_question_matches(mdns_parsed_question_t * question, uint16_t type, mdns_srv_item_t * service)

```
....
2377.            && !strcasecmp(service->service->proto, question-
>proto)
```

NULL Pointer Dereference\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=74
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 2572 is not initialized when it is used by service at ESP8266_RTOS_SDK/mdns.c in line 2368.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2728	2377
Object	null	service

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
2728.                 mdns_srv_item_t * service = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static bool _mdns_question_matches(mdns_parsed_question_t * question, uint16_t type, mdns_srv_item_t * service)

```
....
2377.                 && !strcasecmp(service->service->proto, question-
>proto)
```

NULL Pointer Dereference\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=75
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 2572 is not initialized when it is used by next at ESP8266_RTOS_SDK/mdns.c in line 1092.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2965	1107
Object	null	next

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
2965.                               _mdns_remove_scheduled_answer(packet-
>tcpip_if, packet->ip_protocol, type, NULL);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_remove_scheduled_answer(tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol, uint16_t type, mdns_srv_item_t * service)

```
....
1107.                         if (a->next->type == type && a->next->service
== service->service) {
```

NULL Pointer Dereference\Path 8:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=76
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 2572 is not initialized when it is used by next at ESP8266_RTOS_SDK/mdns.c in line 1092.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	3011	1107
Object	null	next

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
3011.                               _mdns_remove_scheduled_answer(packet-
>tcpip_if, packet->ip_protocol, type, NULL);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_remove_scheduled_answer(tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol, uint16_t type, mdns_srv_item_t * service)

```
....
1107.                           if (a->next->type == type && a->next->service
== service->service) {
```

NULL Pointer Dereference\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=77
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 2572 is not initialized when it is used by next at ESP8266_RTOS_SDK/mdns.c in line 1092.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2728	1107
Object	null	next

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
2728.                   mdns_srv_item_t * service = NULL;
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_remove_scheduled_answer(tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol, uint16_t type, mdns_srv_item_t * service)

```
....
1107.                           if (a->next->type == type && a->next->service
== service->service) {
```

NULL Pointer Dereference\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=78
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 2572 is not initialized when it is used by next at ESP8266_RTOS_SDK/mdns.c in line 1092.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	3011	1107
Object	null	next

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
3011.                               _mdns_remove_scheduled_answer(packet-
>tcpip_if, packet->ip_protocol, type, NULL);
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_remove_scheduled_answer(tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol, uint16_t type, mdns_srv_item_t * service)

```
....
1107.                           if (a->next->type == type && a->next->service
== service->service) {
```

NULL Pointer Dereference\Path 11:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=79
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 2572 is not initialized when it is used by next at ESP8266_RTOS_SDK/mdns.c in line 1092.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2965	1107
Object	null	next

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
2965.                          _mdns_remove_scheduled_answer(packet-
>tcpip_if, packet->ip_protocol, type, NULL);
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_remove_scheduled_answer(tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol, uint16_t type, mdns_srv_item_t * service)

```
....
1107.                      if (a->next->type == type && a->next->service
== service->service) {
```

NULL Pointer Dereference\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=80
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 2572 is not initialized when it is used by next at ESP8266_RTOS_SDK/mdns.c in line 1092.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2728	1107
Object	null	next

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method	void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
2728.                mdns_srv_item_t * service = NULL;
```

▼

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_remove_scheduled_answer(tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol, uint16_t type, mdns_srv_item_t * service)

```
....
1107.                      if (a->next->type == type && a->next->service
== service->service) {
```

NULL Pointer Dereference\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=81
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 2572 is not initialized when it is used by result at ESP8266_RTOS_SDK/mdns.c in line 2572.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2778	2806
Object	null	result

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
2778.                    mdns_result_t * result = NULL;
....
2806.                            if (!result->hostname) { // assign
host/port for this entry only if not previously set
```

NULL Pointer Dereference\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=82
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 2572 is not initialized when it is used by result at ESP8266_RTOS_SDK/mdns.c in line 2572.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2868	2888
Object	null	result

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method void mdns_parse_packet(mdns_rx_packet_t * packet)

```
....
2868.                           mdns_txt_item_t * txt = NULL;
....
2888.                               if (!result->txt) {
```

NULL Pointer Dereference\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=83
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3670.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3674
Object	null	data

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_action(mdns_action_t * action)

```
....
3674.           free(action->data.hostname);
```

NULL Pointer Dereference\Path 16:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=84
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3670.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3677
Object	null	data

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_free_action(mdns_action_t * action)

```
....
3677.           free(action->data.instance);
```

NULL Pointer Dereference\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=85
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3670.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3681
Object	null	data

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_free_action(mdns_action_t * action)

```
....
3681.          free(action->data.srv_add.service);
```

NULL Pointer Dereference\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=86
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3670.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3680
Object	null	data

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_free_action(mdns_action_t * action)

```
....
3680.          _mdns_free_service(action->data.srv_add.service-
>service);
```

NULL Pointer Dereference\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=87
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3670.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3684
Object	null	data

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_service_task(void *pvParameters)

```
....
4005.      mdns_action_t * a = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_action(mdns_action_t * action)

```
....
3684.          free(action->data.srv_instance.instance);
```

NULL Pointer Dereference\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=88
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3670.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3687
Object	null	data

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_service_task(void *pvParameters)

```
....
4005.      mdns_action_t * a = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_action(mdns_action_t * action)

```
....
3687.           _mdns_free_linked_txt(action->data.srv_txt_replace.txt);
```

NULL Pointer Dereference\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=89
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3670.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3691
Object	null	data

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_free_action(mdns_action_t * action)

```
....
3691.           free(action->data.srv_txt_set.value);
```

NULL Pointer Dereference\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=90
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3670.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3690

Object	null	data

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_action(mdns_action_t * action)

```
....
3690.           free(action->data.srv_txt_set.key);
```

NULL Pointer Dereference\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=91
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3670.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3694
Object	null	data

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_action(mdns_action_t * action)

```
....
3694.           free(action->data.srv_txt_del.key);
```

NULL Pointer Dereference\Path 24:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=92
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3670.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3701
Object	null	data

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_free_action(mdns_action_t * action)

```
....
3701.           _mdns_search_free(action->data.search_add.search);
```

NULL Pointer Dereference\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=93
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3670.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3704
Object	null	data

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_service_task(void *pvParameters)

```
....
4005.      mdns_action_t * a = NULL;
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_action(mdns_action_t * action)

```
....
3704.          _mdns_free_tx_packet(action->data.tx_handle.packet);
```

NULL Pointer Dereference\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=94
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3670.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3708
Object	null	data

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_service_task(void *pvParameters)

```
....
4005.      mdns_action_t * a = NULL;
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_free_action(mdns_action_t * action)

```
....
3708.          free(action->data.rx_handle.packet);
```

NULL Pointer Dereference\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=95
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3670.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3707
Object	null	data

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_free_action(mdns_action_t * action)

```
....
3707.           pbuf_free(action->data.rx_handle.packet->pb);
```

NULL Pointer Dereference\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=96
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3730
Object	null	data

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3730.                action->data.sys_event.event_id, action-
>data.sys_event.interface);
```

NULL Pointer Dereference\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=97
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3730
Object	null	data

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3730.                action->data.sys_event.event_id, action-
>data.sys_event.interface);
```

NULL Pointer Dereference\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=98
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination

File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3729
Object	null	data

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3729.            _mdns_handle_system_event(action-
>data.sys_event.event_base,
```

NULL Pointer Dereference\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=99
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3735
Object	null	data

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3735.           _mdns_server->hostname = action->data.hostname;
```

NULL Pointer Dereference\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=100
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3742
Object	null	data

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3742.           _mdns_server->instance = action->data.instance;
```

NULL Pointer Dereference\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=101
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3749

Object	null	data

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3749.           _mdns_probe_all_pcbs(&action->data.srv_add.service, 1,
false, false);
```

NULL Pointer Dereference\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=102
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3748
Object	null	data

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3748.           _mdns_server->services = action->data.srv_add.service;
```

NULL Pointer Dereference\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=103
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3758
Object	null	data

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3758.           _mdns_probe_all_pcbs(&action->data.srv_instance.service,
1, false, false);
```

NULL Pointer Dereference\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=104
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3757
Object	null	data

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3757.           action->data.srv_instance.service->service->instance =
action->data.srv_instance.instance;
```

NULL Pointer Dereference\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=105
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3755
Object	null	data

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3755.               free((char*)action->data.srv_instance.service-
>service->instance);
```

NULL Pointer Dereference\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=106

Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3754
Object	null	data

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_execute_action(mdns_action_t * action)

```
....
3754.                   _mdns_send_bye(&action->data.srv_instance.service, 1,
false);
```

NULL Pointer Dereference\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=107
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3753
Object	null	data

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_execute_action(mdns_action_t * action)

```
....
3753.           if (action->data.srv_instance.service->service->instance)
{
```

NULL Pointer Dereference\Path 40:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=108
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3763
Object	null	data

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_execute_action(mdns_action_t * action)

```
....
3763.           _mdns_announce_all_pcbs(&action->data.srv_port.service,
1, true);
```

NULL Pointer Dereference\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=109
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3762
Object	null	data

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3762.               action->data.srv_port.service->service->port = action-
>data.srv_port.port;
```

NULL Pointer Dereference\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=110
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3772
Object	null	data

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3772.          _mdns_announce_all_pcbs(&action-
>data.srv_txt_replace.service, 1, false);
```

NULL Pointer Dereference\Path 43:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=111
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3771
Object	null	data

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_service_task(void *pvParameters)

```
....
4005.      mdns_action_t * a = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3771.          service->txt = action->data.srv_txt_replace.txt;
```

NULL Pointer Dereference\Path 44:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=112
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3767

Object	null	data

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3767.           service = action->data.srv_txt_replace.service->service;
```

NULL Pointer Dereference\Path 45:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=113
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3802
Object	null	data

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3802.           _mdns_announce_all_pcbs(&action-
>data.srv_txt_set.service, 1, false);
```

NULL Pointer Dereference\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=114
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3778
Object	null	data

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

▼

File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3778.           value = action->data.srv_txt_set.value;
```

NULL Pointer Dereference\Path 47:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=115
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3777
Object	null	data

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3777.           key = action->data.srv_txt_set.key;
```

NULL Pointer Dereference\Path 48:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=116
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3776
Object	null	data

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3776.           service = action->data.srv_txt_set.service->service;
```

NULL Pointer Dereference\Path 49:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=117
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3833
Object	null	data

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_execute_action(mdns_action_t * action)

```
....
3833.             _mdns_announce_all_pcbs(&action-
>data.srv_txt_set.service, 1, false);
```

NULL Pointer Dereference\Path 50:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=118
Status	New

The variable declared in null at ESP8266_RTOS_SDK/mdns.c in line 4003 is not initialized when it is used by data at ESP8266_RTOS_SDK/mdns.c in line 3719.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4005	3807
Object	null	data

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static void _mdns_service_task(void *pvParameters)

```
....
4005.       mdns_action_t * a = NULL;
```

▼

File Name ESP8266_RTOS_SDK/mdns.c

Method	static void _mdns_execute_action(mdns_action_t * action)

```
....
3807.           key = action->data.srv_txt_del.key;
```

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=1050073&projectid=50063&pathid=250
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	721	721
Object	fgets	fgets

Code Snippet

File Name	ESP8266_RTOS_SDK/conf.c
Method	void xfgets(char *str, int size, FILE *in)

```
....
721.        if (fgets(str, size, in) == NULL)
```

Improper Resource Access Authorization\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=251
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	721	721
Object	str	str

Code Snippet

File Name	ESP8266_RTOS_SDK/conf.c
Method	void xfgets(char *str, int size, FILE *in)

```
....
721.        if (fgets(str, size, in) == NULL)
```

Improper Resource Access Authorization\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=252
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	722	722
Object	fprintf	fprintf

Code Snippet

File Name	ESP8266_RTOS_SDK/conf.c
Method	void xfgets(char *str, int size, FILE *in)

```
....
722.            fprintf(stderr, "\nError in reading or end of
file.\n");
```

Improper Resource Access Authorization\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=253
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	540	540
Object	fprintf	fprintf

Code Snippet

File Name	ESP8266_RTOS_SDK/conf.c
Method	int main(int ac, char **av)

```
....
540.                fprintf( stderr, "KCONFIG_SEED=0x%X\n", seed );
```

Improper Resource Access Authorization\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=254
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	570	570
Object	fprintf	fprintf

Code Snippet
File Name ESP8266_RTOS_SDK/conf.c
Method int main(int ac, char **av)

```
....
570.                            fprintf(stderr, _("***\n"
```

Improper Resource Access Authorization\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=255
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	609	609
Object	fprintf	fprintf

Code Snippet
File Name ESP8266_RTOS_SDK/conf.c
Method int main(int ac, char **av)

```
....
609.                                    fprintf(stderr,
```

Improper Resource Access Authorization\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=256
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	626	626
Object	fprintf	fprintf

Code Snippet
File Name ESP8266_RTOS_SDK/conf.c
Method int main(int ac, char **av)

```
....
626.                        fprintf(stderr,
```

Improper Resource Access Authorization\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=257
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	640	640
Object	fprintf	fprintf

Code Snippet
File Name ESP8266_RTOS_SDK/conf.c
Method int main(int ac, char **av)

```
....
640.                              fprintf(stderr,
```

Improper Resource Access Authorization\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=258
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	694	694
Object	fprintf	fprintf

Code Snippet

File Name ESP8266_RTOS_SDK/conf.c
Method int main(int ac, char **av)

```
....
694.                    fprintf(stderr, _("\n*** Error during writing of
the configuration.\n\n"));
```

Improper Resource Access Authorization\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=259
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	698	698
Object	fprintf	fprintf

Code Snippet
File Name ESP8266_RTOS_SDK/conf.c
Method int main(int ac, char **av)

```
....
698.                    fprintf(stderr, _("\n*** Error during update of
the configuration.\n\n"));
```

Improper Resource Access Authorization\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=260
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	703	703
Object	fprintf	fprintf

Code Snippet
File Name ESP8266_RTOS_SDK/conf.c
Method int main(int ac, char **av)

```
....
703.                    fprintf(stderr, _("n*** Error while saving
defconfig to: %s\n\n"),
```

Improper Resource Access Authorization\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=261
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	709	709
Object	fprintf	fprintf

Code Snippet
File Name ESP8266_RTOS_SDK/conf.c
Method int main(int ac, char **av)

```
....
709.                    fprintf(stderr, _("\n*** Error during writing of
the configuration.\n\n"));
```

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=1050073&projectid=50063&pathid=9
Status	New

The _mdns_mangle_name method calls the sprintf function, at line 53 of ESP8266_RTOS_SDK/mdns.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	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	76	76
Object	sprintf	sprintf

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static char * _mdns_mangle_name(char* in) {

```
....
76.             sprintf(ret, "%s-2", in);
```

Unchecked Return Value\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=10
Status	New

The _mdns_mangle_name method calls the sprintf function, at line 53 of ESP8266_RTOS_SDK/mdns.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	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	86	86
Object	sprintf	sprintf

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static char * _mdns_mangle_name(char* in) {

```
....
86.             sprintf(ret + baseLen, "-%d", suffix + 1);
```

Unchecked Return Value\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=11
Status	New

The _mdns_append_txt_record method calls the sprintf function, at line 539 of ESP8266_RTOS_SDK/mdns.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	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	578	578
Object	sprintf	sprintf

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static uint16_t _mdns_append_txt_record(uint8_t * packet, uint16_t * index, mdns_service_t * service, bool flush, bool bye)

```
....
578.                sprintf(tmp, "%s=%s", txt->key, txt->value);
```

Unchecked Return Value\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=12
Status	New

The _mdns_check_txt_collision method calls the sprintf function, at line 2095 of ESP8266_RTOS_SDK/mdns.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	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2126	2126
Object	sprintf	sprintf

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static int _mdns_check_txt_collision(mdns_service_t * service, const uint8_t * data, size_t len)

```
....
2126.                sprintf(tmp, "%s=%s", txt->key, txt->value);
```

Unchecked Return Value\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=13
Status	New

The _mdns_strdup_check method calls the Pointer function, at line 2554 of ESP8266_RTOS_SDK/mdns.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	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2557	2557
Object	Pointer	Pointer

Code Snippet

File Name ESP8266_RTOS_SDK/mdns.c

Method static esp_err_t _mdns_strdup_check(char ** out, char * in)

```
....
2557.           *out = strdup(in);
```

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=1050073&projectid=50063&pathid=143
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	607	607
Object	n2	n2

Code Snippet
File Name ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method crypto_aead_aes256gcm_encrypt_detached_afternm(unsigned char *c,

```
....
607.       COUNTER_INC2(n2);
```

Unchecked Array Index\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=144
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	770	770
Object	n2	n2

Code Snippet
File Name ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c

Method	crypto_aead_aes256gcm_decrypt_detached_afternm(unsigned char *m, unsigned char *nsec,

```
....
770.        COUNTER_INC2(n2);
```

Unchecked Array Index\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=145
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Line	791	791
Object	n2	n2

Code Snippet	
File Name	ESP8266_RTOS_SDK/aead_aes256gcm_aesni.c
Method	crypto_aead_aes256gcm_decrypt_detached_afternm(unsigned char *m, unsigned char *nsec,

```
....
791.        COUNTER_INC2(n2);
```

Unchecked Array Index\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=146
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/json.c	ESP8266_RTOS_SDK/json.c
Line	182	182
Object	len	len

Code Snippet	
File Name	ESP8266_RTOS_SDK/json.c
Method	static int json_parse_number(const char **json_pos, const char *end,

```
....
182.          str[len] = '\0';
```

Potential Precision Problem

Query Path:
CPP\Cx\CPP Buffer Overflow\Potential Precision Problem Version:0

Categories

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

Description

Potential Precision Problem\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=131
Status	New

The size of the buffer used by _mdns_mangle_name in "%s-2", at line 53 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that _mdns_mangle_name passes to "%s-2", at line 53 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	76	76
Object	"%s-2"	"%s-2"

Code Snippet
File Name ESP8266_RTOS_SDK/mdns.c
Method static char * _mdns_mangle_name(char* in) {

```
....
76.          sprintf(ret, "%s-2", in);
```

Potential Precision Problem\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=132
Status	New

The size of the buffer used by _mdns_append_txt_record in "%s=%s", at line 539 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that _mdns_append_txt_record passes to "%s=%s", at line 539 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	578	578
Object	"%s=%s"	"%s=%s"

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static uint16_t _mdns_append_txt_record(uint8_t * packet, uint16_t * index, mdns_service_t * service, bool flush, bool bye)

```
....
578.              sprintf(tmp, "%s=%s", txt->key, txt->value);
```

Potential Precision Problem\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=133
Status	New

The size of the buffer used by _mdns_check_txt_collision in "%s=%s", at line 2095 of ESP8266_RTOS_SDK/mdns.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that _mdns_check_txt_collision passes to "%s=%s", at line 2095 of ESP8266_RTOS_SDK/mdns.c, to overwrite the target buffer.

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	2126	2126
Object	"%s=%s"	"%s=%s"

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static int _mdns_check_txt_collision(mdns_service_t * service, const uint8_t * data, size_t len)

```
....
2126.              sprintf(tmp, "%s=%s", txt->key, txt->value);
```

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=1050073&projectid=50063&pathid=14
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	1489	1489
Object	sizeof	sizeof

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	static void _mdns_init_pcb_probe_new_service(tcpip_adapter_if_t tcpip_if, mdns_ip_protocol_t ip_protocol, mdns_srv_item_t ** services, size_t len, bool probe_ip)

```
....
1489.           _services = (mdns_srv_item_t
**)malloc(sizeof(mdns_srv_item_t *) * services_final_len);
```

Use of Sizeof On a Pointer Type\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=15
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/mdns.c	ESP8266_RTOS_SDK/mdns.c
Line	4172	4172
Object	sizeof	sizeof

Code Snippet

File Name	ESP8266_RTOS_SDK/mdns.c
Method	esp_err_t mdns_init()

```
....
4172.       _mdns_server->action_queue =
xQueueCreate(MDNS_ACTION_QUEUE_LEN, sizeof(mdns_action_t *));
```

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=1050073&projectid=50063&pathid=262
Status	New

The system data read by main in the file ESP8266_RTOS_SDK/conf.c at line 492 is potentially exposed by main found in ESP8266_RTOS_SDK/conf.c at line 492.

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	604	626
Object	getenv	fprintf

Code Snippet
File Name   ESP8266_RTOS_SDK/conf.c
Method   int main(int ac, char **av)

```
....
604.                  name = getenv("KCONFIG_ALLCONFIG");
....
626.                        fprintf(stderr,
```

Exposure of System Data to Unauthorized Control Sphere\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=263
Status	New

The system data read by main in the file ESP8266_RTOS_SDK/conf.c at line 492 is potentially exposed by main found in ESP8266_RTOS_SDK/conf.c at line 492.

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	604	609
Object	getenv	fprintf

Code Snippet
File Name   ESP8266_RTOS_SDK/conf.c
Method   int main(int ac, char **av)

```
....
604.                  name = getenv("KCONFIG_ALLCONFIG");
....
609.                              fprintf(stderr,
```

Inconsistent Implementations

Query Path:
CPP\Cx\CPP Low Visibility\Inconsistent Implementations Version:0

Description

Inconsistent Implementations\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=7
Status	New

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	505	505
Object	getopt_long	getopt_long

Code Snippet
File Name ESP8266_RTOS_SDK/conf.c
Method int main(int ac, char **av)

```
....
505.         while ((opt = getopt_long(ac, av, "s", long_opts, NULL)) !=
-1) {
```

Use of Insufficiently Random Values

Query Path:
CPP\Cx\CPP Low Visibility\Use of Insufficiently Random Values Version:0

Categories

FISMA 2014: Media Protection
NIST SP 800-53: SC-28 Protection of Information at Rest (P1)
OWASP Top 10 2017: A3-Sensitive Data Exposure

Description

Use of Insufficiently Random Values\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050073&projectid=50063&pathid=8
Status	New

Method main at line 492 of ESP8266_RTOS_SDK/conf.c uses a weak method srand to produce random values. These values might be used for secret values, personal identifiers or cryptographic input, allowing an attacker to guess the value.

	Source	Destination
File	ESP8266_RTOS_SDK/conf.c	ESP8266_RTOS_SDK/conf.c
Line	541	541
Object	srand	srand

Code Snippet
File Name ESP8266_RTOS_SDK/conf.c
Method int main(int ac, char **av)

```
....
541.                     srand(seed);
```

Buffer Overflow LongString

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.

Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

Buffer Overflow OutOfBound

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.

Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

Buffer Overflow Indexes

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.

Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

Buffer Overflow fgets

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.

Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

CPP

Overflowing Buffers

```
const int BUFFER_SIZE = 10;
char buffer[BUFFER_SIZE];

void copyStringToBuffer(char* inputString)
{
    strcpy(buffer, inputString);
}
```

Checked Buffers

```
const int BUFFER_SIZE = 10;
const int MAX_INPUT_SIZE = 256;
```

```
char buffer[BUFFER_SIZE];

void copyStringToBuffer(char* inputString)
{
    if (strnlen(inputString, MAX_INPUT_SIZE) < sizeof(buffer))
    {
        strncpy(buffer, inputString, sizeof(buffer));
    }
}
```

Buffer Overflow boundcpy WrongSizeParam

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.

Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

MemoryFree on StackVariable

Risk

What might happen

Undefined Behavior may result with a crash. Crashes may give an attacker valuable information about the system and the program internals. Furthermore, it may leave unprotected files (e.g memory) that may be exploited.

Cause

How does it happen

Calling free() on a variable that was not dynamically allocated (e.g. malloc) will result with an Undefined Behavior.

General Recommendations

How to avoid it

Use free() only on dynamically allocated variables in order to prevent unexpected behavior from the compiler.

Source Code Examples

CPP

Bad - Calling free() on a static variable

```
void clean_up(){
  char temp[256];
  do_something();
  free(tmp);
  return;
}
```

Good - Calling free() only on variables that were dynamically allocated

```
void clean_up(){
  char *buff;
  buff = (char*) malloc(1024);
  free(buff);
  return;
}
```

Wrong Size t Allocation

Risk

What might happen

Incorrect allocation of memory may result in unexpected behavior by either overwriting sections of memory with unexpected values. Under certain conditions where both an incorrect allocation of memory and the values being written can be controlled by an attacker, such an issue may result in execution of malicious code.

Cause

How does it happen

Some memory allocation functions require a size value to be provided as a parameter. The allocated size should be derived from the provided value, by providing the length value of the intended source, multiplied by the size of that length. Failure to perform the correct arithmetic to obtain the exact size of the value will likely result in the source overflowing its destination.

General Recommendations

How to avoid it

- Always perform the correct arithmetic to determine size.
- Specifically for memory allocation, calculate the allocation size from the allocation source:
    - Derive the size value from the length of intended source to determine the amount of units to be processed.
    - Always programmatically consider the size of the each unit and their conversion to memory units - for example, by using sizeof() on the unit's type.
    - Memory allocation should be a multiplication of the amount of units being written, times the size of each unit.

Source Code Examples

CPP

Allocating and Assigning Memory without Sizeof Arithmetic

```
int *ptr;
ptr = (int*)malloc(5);
for (int i = 0; i < 5; i++)
{
      ptr[i] = i * 2 + 1;
}
```

Allocating and Assigning Memory with Sizeof Arithmetic

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
```

```
for (int i = 0; i < 5; i++)
{
      ptr[i] = i * 2 + 1;
}
```

Incorrect Arithmetic of Multi-Byte String Allocation

```
wchar_t * dest;
dest = (wchar_t *)malloc(wcslen(source) + 1); // Would not crash for a short "source"
wcscpy((wchar_t *)dest, source);
wprintf(L"Dest: %s\r\n", dest);
```

Correct Arithmetic of Multi-Byte String Allocation

```
wchar_t * dest;
dest = (wchar_t *)malloc((wcslen(source) + 1) * sizeof(wchar_t));
wcscpy((wchar_t *)dest, source);
wprintf(L"Dest: %s\r\n", dest);
```

Char Overflow

Risk

What might happen

Assigning large data types into smaller data types, without proper checks and explicit casting, will lead to undefined behavior and unintentional effects, such as data corruption (e.g. value wraparound, wherein maximum values become minimum values); system crashes; infinite loops; logic errors, such as bypassing of security mechanisms; or even buffer overflows leading to arbitrary code execution.

Cause

How does it happen

This flaw can occur when implicitly casting numerical data types of a larger size, into a variable with a data type of a smaller size. This forces the program to discard some bits of information from the number. Depending on how the numerical data types are stored in memory, this is often the bits with the highest value, causing substantial corruption of the stored number. Alternatively, the sign bit of a signed integer could be lost, completely reversing the intention of the number.

General Recommendations

How to avoid it

- Avoid casting larger data types to smaller types.
- Prefer promoting the target variable to a large enough data type.
- If downcasting is necessary, always check that values are valid and in range of the target type, before casting

Source Code Examples

CPP

Unsafe Downsize Casting

```
int unsafe_addition(short op1, int op2) {

    // op2 gets forced from int into a short
    short total = op1 + op2;

    return total;
}
```

Safer Use of Proper Data Types

```
int safe_addition(short op1, int op2) {

    // total variable is of type int, the largest type that is needed
    int total = 0;

    // check if total will overflow available integer size
    if (INT_MAX - abs(op2) > op1)
    {
        total = op1 + op2;
    }
    else
    {
        // instead of overflow, saturate (but this is not always a good thing)
        total = INT_MAX
    }

    return total;
}
```

Integer Overflow

Risk

What might happen

Assigning large data types into smaller data types, without proper checks and explicit casting, will lead to undefined behavior and unintentional effects, such as data corruption (e.g. value wraparound, wherein maximum values become minimum values); system crashes; infinite loops; logic errors, such as bypassing of security mechanisms; or even buffer overflows leading to arbitrary code execution.

Cause

How does it happen

This flaw can occur when implicitly casting numerical data types of a larger size, into a variable with a data type of a smaller size. This forces the program to discard some bits of information from the number. Depending on how the numerical data types are stored in memory, this is often the bits with the highest value, causing substantial corruption of the stored number. Alternatively, the sign bit of a signed integer could be lost, completely reversing the intention of the number.

General Recommendations

How to avoid it

- Avoid casting larger data types to smaller types.
- Prefer promoting the target variable to a large enough data type.
- If downcasting is necessary, always check that values are valid and in range of the target type, before casting

Source Code Examples

Dangerous Functions

Risk

What might happen

Use of dangerous functions may expose varying risks associated with each particular function, with potential impact of improper usage of these functions varying significantly. The presence of such functions indicates a flaw in code maintenance policies and adherence to secure coding practices, in a way that has allowed introducing known dangerous code into the application.

Cause

How does it happen

A dangerous function has been identified within the code. Functions are often deemed dangerous to use for numerous reasons, as there are different sets of vulnerabilities associated with usage of such functions. For example, some string copy and concatenation functions are vulnerable to Buffer Overflow, Memory Disclosure, Denial of Service and more. Use of these functions is not recommended.

General Recommendations

How to avoid it

- Deploy a secure and recommended alternative to any functions that were identified as dangerous.
    - If no secure alternative is found, conduct further researching and testing to identify whether current usage successfully sanitizes and verifies values, and thus successfully avoids the use-cases for whom the function is indeed dangerous
- Conduct a periodical review of methods that are in use, to ensure that all external libraries and built-in functions are up-to-date and whose use has not been excluded from best secure coding practices.

Source Code Examples

CPP

Buffer Overflow in gets()

```
int main()

{

      char buf[10];

      printf("Please enter your name: ");
      gets(buf); // veryveryverylongname
      if (buf == ACCEPTED_NAME)
      {
             // Do something
      }
      return 0;
}
```

Safe reading from user

```
int main()
{
        char buf[10];

        printf("Please enter your name: ");
        fgets(buf, sizeof(buf), stdin); //setting the amount of bytes to read
        if (buf == ACCEPTED_NAME)
        {
                //Do something
        }
        return 0;
}
```

Unsafe function for string copy

```
int main(int argc, char* argv[])
{
        char buf[10];
        strcpy(buf, argv[1]); // overflow occurs when len(argv[1]) > 10 bytes

        return 0;
}
```

Safe string copy

```
int main(int argc, char* argv[])
{
        char buf[10];
        strncpy(buf, argv[1], sizeof(buf));
        buf[9]= '\0'; //strncpy doesn't NULL terminates

        return 0;
}
```

Unsafe format string

```
int main(int argc, char* argv[])
{
        printf(argv[1]); // If argv[1] contains a format token, such as %s,%x or %d, will cause
an access violation
        return 0;
}
```

Safe format string

```
int main(int argc, char* argv[])
{
      printf("%s", argv[1]); // Second parameter is not a formattable string

      return 0;
}
```

Double Free

Weakness ID: 415 *(Weakness Variant)* **Status:** Draft

Description

Description Summary

The product calls free() twice on the same memory address, potentially leading to modification of unexpected memory locations.

Extended Description

When a program calls free() twice with the same argument, the program's memory management data structures become corrupted. This corruption can cause the program to crash or, in some circumstances, cause two later calls to malloc() to return the same pointer. If malloc() returns the same value twice and the program later gives the attacker control over the data that is written into this doubly-allocated memory, the program becomes vulnerable to a buffer overflow attack.

Alternate Terms

Double-free

Time of Introduction

- Architecture and Design
- Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

Scope	Effect
Access Control	Doubly freeing memory may result in a write-what-where condition, allowing an attacker to execute arbitrary code.

Likelihood of Exploit

Low to Medium

Demonstrative Examples

Example 1

The following code shows a simple example of a double free vulnerability.

(Bad Code)

Example Language: **C**

```
char* ptr = (char*)malloc (SIZE);
...
if (abrt) {
free(ptr);
}
...
free(ptr);
```

Double free vulnerabilities have two common (and sometimes overlapping) causes:

- Error conditions and other exceptional circumstances
- Confusion over which part of the program is responsible for freeing the memory

Although some double free vulnerabilities are not much more complicated than the previous example, most are spread out across hundreds of lines of code or even different files. Programmers seem particularly susceptible to freeing global variables more than once.

Example 2

While contrived, this code should be exploitable on Linux distributions which do not ship with heap-chunk check summing turned on.

(Bad Code)

Example Language: **C**

```
#include <stdio.h>
#include <unistd.h>
#define BUFSIZE1 512
#define BUFSIZE2 ((BUFSIZE1/2) - 8)

int main(int argc, char **argv) {
char *buf1R1;
char *buf2R1;
char *buf1R2;
buf1R1 = (char *) malloc(BUFSIZE2);
buf2R1 = (char *) malloc(BUFSIZE2);
free(buf1R1);
free(buf2R1);
buf1R2 = (char *) malloc(BUFSIZE1);
strncpy(buf1R2, argv[1], BUFSIZE1-1);
free(buf2R1);
free(buf1R2);
}
```

Observed Examples

Reference	Description
CVE-2004-0642	Double free resultant from certain error conditions.
CVE-2004-0772	Double free resultant from certain error conditions.
CVE-2005-1689	Double free resultant from certain error conditions.
CVE-2003-0545	Double free from invalid ASN.1 encoding.
CVE-2003-1048	Double free from malformed GIF.
CVE-2005-0891	Double free from malformed GIF.
CVE-2002-0059	Double free from malformed compressed data.

Potential Mitigations

Phase: Architecture and Design

Choose a language that provides automatic memory management.

Phase: Implementation

Ensure that each allocation is freed only once. After freeing a chunk, set the pointer to NULL to ensure the pointer cannot be freed again. In complicated error conditions, be sure that clean-up routines respect the state of allocation properly. If the language is object oriented, ensure that object destructors delete each chunk of memory only once.

Phase: Implementation

Use a static analysis tool to find double free instances.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	398	Indicator of Poor Code Quality	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Category	399	Resource Management Errors	**Development Concepts (primary)699**
ChildOf	Category	633	Weaknesses that Affect Memory	**Resource-specific Weaknesses (primary)631**
ChildOf	Weakness Base	666	Operation on Resource in Wrong Phase of	**Research Concepts (primary)1000**

			Lifetime	
ChildOf	Weakness Class	675	Duplicate Operations on Resource	Research Concepts1000
ChildOf	Category	742	CERT C Secure Coding Section 08 - Memory Management (MEM)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
PeerOf	Weakness Base	123	Write-what-where Condition	Research Concepts1000
PeerOf	Weakness Base	416	Use After Free	Development Concepts699 Research Concepts1000
MemberOf	View	630	Weaknesses Examined by SAMATE	**Weaknesses Examined by SAMATE (primary)630**
PeerOf	Weakness Base	364	Signal Handler Race Condition	Research Concepts1000

Relationship Notes

This is usually resultant from another weakness, such as an unhandled error or race condition between threads. It could also be primary to weaknesses such as buffer overflows.

Affected Resources

- Memory

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
PLOVER			DFREE - Double-Free Vulnerability
7 Pernicious Kingdoms			Double Free
CLASP			Doubly freeing memory
CERT C Secure Coding	MEM00-C		Allocate and free memory in the same module, at the same level of abstraction
CERT C Secure Coding	MEM01-C		Store a new value in pointers immediately after free()
CERT C Secure Coding	MEM31-C		Free dynamically allocated memory exactly once

White Box Definitions

A weakness where code path has:

1. start statement that relinquishes a dynamically allocated memory resource
2. end statement that relinquishes the dynamically allocated memory resource

Maintenance Notes

It could be argued that Double Free would be most appropriately located as a child of "Use after Free", but "Use" and "Release" are considered to be distinct operations within vulnerability theory, therefore this is more accurately "Release of a Resource after Expiration or Release", which doesn't exist yet.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	PLOVER		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Potential Mitigations, Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Description, Maintenance Notes, Relationships, Other Notes, Relationship Notes, Taxonomy Mappings		
2008-11-24	CWE Content Team	MITRE	Internal

	updated Relationships, Taxonomy Mappings		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Other Notes		

BACK TO TOP

Failure to Release Memory Before Removing Last Reference ('Memory Leak')

Weakness ID: 401 *(Weakness Base)* **Status:** Draft

Description

Description Summary

The software does not sufficiently track and release allocated memory after it has been used, which slowly consumes remaining memory.

Extended Description

This is often triggered by improper handling of malformed data or unexpectedly interrupted sessions.

Terminology Notes

"memory leak" has sometimes been used to describe other kinds of issues, e.g. for information leaks in which the contents of memory are inadvertently leaked (CVE-2003-0400 is one such example of this terminology conflict).

Time of Introduction

- Architecture and Design
- Implementation

Applicable Platforms

Languages

C

C++

Modes of Introduction

Memory leaks have two common and sometimes overlapping causes:

- Error conditions and other exceptional circumstances
- Confusion over which part of the program is responsible for freeing the memory

Common Consequences

Scope	Effect
Availability	Most memory leaks result in general software reliability problems, but if an attacker can intentionally trigger a memory leak, the attacker might be able to launch a denial of service attack (by crashing or hanging the program) or take advantage of other unexpected program behavior resulting from a low memory condition.

Likelihood of Exploit

Medium

Demonstrative Examples

Example 1

The following C function leaks a block of allocated memory if the call to read() fails to return the expected number of bytes:

(Bad Code)

Example Language: **C**

```
char* getBlock(int fd) {
char* buf = (char*) malloc(BLOCK_SIZE);
if (!buf) {
return NULL;
}
if (read(fd, buf, BLOCK_SIZE) != BLOCK_SIZE) {

return NULL;
}
return buf;
}
```

Example 2

Here the problem is that every time a connection is made, more memory is allocated. So if one just opened up more and more connections, eventually the machine would run out of memory.

(Bad Code)

Example Language: **C**

```
bar connection(){
foo = malloc(1024);
return foo;
}
endConnection(bar foo) {

free(foo);
}
int main() {

while(1) //thread 1
//On a connection
foo=connection(); //thread 2
//When the connection ends
endConnection(foo)
}
```

Observed Examples

Reference	Description
CVE-2005-3119	Memory leak because function does not free() an element of a data structure.
CVE-2004-0427	Memory leak when counter variable is not decremented.
CVE-2002-0574	Memory leak when counter variable is not decremented.
CVE-2005-3181	Kernel uses wrong function to release a data structure, preventing data from being properly tracked by other code.
CVE-2004-0222	Memory leak via unknown manipulations as part of protocol test suite.
CVE-2001-0136	Memory leak via a series of the same command.

Potential Mitigations

Pre-design: Use a language or compiler that performs automatic bounds checking.

Phase: Architecture and Design

Use an abstraction library to abstract away risky APIs. Not a complete solution.

Pre-design through Build: The Boehm-Demers-Weiser Garbage Collector or valgrind can be used to detect leaks in code. This is not a complete solution as it is not 100% effective.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	398	Indicator of Poor Code Quality	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Category	399	Resource Management Errors	**Development Concepts (primary)699**
ChildOf	Category	633	Weaknesses that Affect Memory	**Resource-specific Weaknesses (primary)631**
ChildOf	Category	730	OWASP Top Ten 2004 Category A9 - Denial of Service	**Weaknesses in OWASP Top Ten (2004) (primary)711**
ChildOf	Weakness Base	772	Missing Release of Resource after Effective	**Research Concepts (primary)1000**

			Lifetime	
MemberOf	View	630	Weaknesses Examined by SAMATE	**Weaknesses Examined by SAMATE (primary)630**
CanFollow	Weakness Class	390	Detection of Error Condition Without Action	Research Concepts1000

Relationship Notes

This is often a resultant weakness due to improper handling of malformed data or early termination of sessions.

Affected Resources

- Memory

Functional Areas

- Memory management

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
PLOVER			Memory leak
7 Pernicious Kingdoms			Memory Leak
CLASP			Failure to deallocate data
OWASP Top Ten 2004	A9	CWE More Specific	Denial of Service

White Box Definitions

A weakness where the code path has:

1. start statement that allocates dynamically allocated memory resource

2. end statement that loses identity of the dynamically allocated memory resource creating situation where dynamically allocated memory resource is never relinquished

Where "loses" is defined through the following scenarios:

1. identity of the dynamic allocated memory resource never obtained

2. the statement assigns another value to the data element that stored the identity of the dynamically allocated memory resource and there are no aliases of that data element

3. identity of the dynamic allocated memory resource obtained but never passed on to function for memory resource release

4. the data element that stored the identity of the dynamically allocated resource has reached the end of its scope at the statement and there are no aliases of that data element

References

J. Whittaker and H. Thompson. "How to Break Software Security". Addison Wesley. 2003.

Content History

Submissions

Submission Date	Submitter	Organization	Source
	PLOVER		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-08-15		Veracode	External
	Suggested OWASP Top Ten 2004 mapping		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Relationships, Other Notes, References, Relationship Notes, Taxonomy Mappings, Terminology Notes		
2008-10-14	CWE Content Team	MITRE	Internal
	updated Description		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Other Notes		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Name		
2009-07-17	KDM Analytics		External
	Improved the White Box Definition		

2009-07-27	CWE Content Team	MITRE	Internal
	updated White Box Definitions		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Modes of Introduction, Other Notes		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		

Previous Entry Names

Change Date	Previous Entry Name
2008-04-11	Memory Leak
2009-05-27	Failure to Release Memory Before Removing Last Reference (aka 'Memory Leak')

BACK TO TOP

Use of Zero Initialized Pointer

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
- Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
- Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.

Source Code Examples

CPP

Explicit NULL Dereference

```
char * input = NULL;
printf("%s", input);
```

Implicit NULL Dereference

```
char * input;
printf("%s", input);
```

Java

Explicit Null Dereference

```
Object o = null;
out.println(o.getClass());
```

Stored Buffer Overflow fgets

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.

Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

Use of Function with Inconsistent Implementations

Weakness ID: 474 *(Weakness Base)* **Status:** Draft

Description

Description Summary

The code uses a function that has inconsistent implementations across operating systems and versions, which might cause security-relevant portability problems.

Time of Introduction

- Architecture and Design
- Implementation

Applicable Platforms

Languages

C: *(Often)*

PHP: *(Often)*

All

Potential Mitigations

Do not accept inconsistent behavior from the API specifications when the deviant behavior increase the risk level.

Other Notes

The behavior of functions in this category varies by operating system, and at times, even by operating system version. Implementation differences can include:

- Slight differences in the way parameters are interpreted leading to inconsistent results.

- Some implementations of the function carry significant security risks.

- The function might not be defined on all platforms.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	398	Indicator of Poor Code Quality	**Development Concepts (primary)699** **Seven Pernicious Kingdoms (primary)700** **Research Concepts (primary)1000**
ParentOf	Weakness Variant	589	Call to Non-ubiquitous API	**Research Concepts (primary)1000**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
7 Pernicious Kingdoms			Inconsistent Implementations

Content History

Submissions

Submission Date	Submitter	Organization	Source
	7 Pernicious Kingdoms		Externally Mined

Modifications

Modification Date	Modifier	Organization	Source
2008-07-01	Eric Dalci	Cigital	External
	updated Potential Mitigations, Time of Introduction		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Relationships, Other Notes, Taxonomy Mappings		

Previous Entry Names

Change Date	Previous Entry Name
2008-04-11	Inconsistent Implementations

BACK TO TOP

Use of Insufficiently Random Values

Risk

What might happen

Random values are often used as a mechanism to prevent malicious users from guessing a value, such as a password, encryption key, or session identifier. Depending on what this random value is used for, an attacker would be able to predict the next numbers generated, or previously generated values. This could enable the attacker to hijack another user's session, impersonate another user, or crack an encryption key (depending on what the pseudo-random value was used for).

Cause

How does it happen

The application uses a weak method of generating pseudo-random values, such that other numbers could be determined from a relatively small sample size. Since the pseudo-random number generator used is designed for statistically uniform distribution of values, it is approximately deterministic. Thus, after collecting a few generated values (e.g. by creating a few individual sessions, and collecting the sessionids), it would be possible for an attacker to calculate another sessionid.

Specifically, if this pseudo-random value is used in any security context, such as passwords, keys, or secret identifiers, an attacker would be able to predict the next numbers generated, or previously generated values.

General Recommendations

How to avoid it

Generic Guidance:

- Whenever unpredicatable numbers are required in a security context, use a cryptographically strong random number generator, instead of a statistical pseudo-random generator.
- Use the cryptorandom generator that is built-in to your language or platform, and ensure it is securely seeded. Do not seed the generator with a weak, non-random seed. (In most cases, the default is securely random).
- Ensure you use a long enough random value, to make brute-force attacks unfeasible.

Specific Recommendations:

- Do not use the statistical pseudo-random number generator, use the cryptorandom generator instead. In Java, this is the SecureRandom class.

Source Code Examples

Java

Use of a weak pseudo-random number generator

```
Random random = new Random();

long sessNum = random.nextLong();

String sessionId = sessNum.toString();
```

Cryptographically secure random number generator

```
SecureRandom random = new SecureRandom();

byte sessBytes[] = new byte[32];

random.nextBytes(sessBytes);

String sessionId = new String(sessBytes);
```

Objc

Use of a weak pseudo-random number generator

```
long sessNum = rand();
NSString* sessionId = [NSString stringWithFormat:@"%ld", sessNum];
```

Cryptographically secure random number generator

```
UInt32 sessBytes;
SecRandomCopyBytes(kSecRandomDefault, sizeof(sessBytes), (uint8_t*)&sessBytes);

NSString* sessionId = [NSString stringWithFormat:@"%llu", sessBytes];
```

Swift

Use of a weak pseudo-random number generator

```
let sessNum = rand();
let sessionId = String(format:"%ld", sessNum)
```

Cryptographically secure random number generator

```
var sessBytes: UInt32 = 0
withUnsafeMutablePointer(&sessBytes, { (sessBytesPointer) -> Void in
    let castedPointer = unsafeBitCast(sessBytesPointer, UnsafeMutablePointer<UInt8>.self)
    SecRandomCopyBytes(kSecRandomDefault, sizeof(UInt32), castedPointer)
})

let sessionId = String(format:"%llu", sessBytes)
```

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 updated Time of Introduction	Cigital	External
2008-08-01	added/updated white box definitions	KDM Analytics	External
2008-09-08	CWE Content Team updated Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities	MITRE	Internal
2008-11-24	CWE Content Team updated Relationships, Taxonomy Mappings	MITRE	Internal
2009-03-10	CWE Content Team updated Demonstrative Examples	MITRE	Internal
2009-12-28	CWE Content Team updated Demonstrative Examples	MITRE	Internal
2010-02-16	CWE Content Team updated Relationships	MITRE	Internal

BACK TO TOP

NULL Pointer Dereference

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
- Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
- Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.

Source Code Examples

Potential Precision Problem

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.

Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

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

```java
    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