Tasmota Scan Report

Project Name	Tasmota
Scan Start	Friday, June 21, 2024 10:31:37 PM
Preset	Checkmarx Default
Scan Time	00h:09m:07s
Lines Of Code Scanned	41018
Files Scanned	35
Report Creation Time	Friday, June 21, 2024 10:43:13 PM
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042
Team	CxServer
Checkmarx Version	8.7.0
Scan Type	Full
Source Origin	LocalPath
Density	6/1000 (Vulnerabilities/LOC)
Visibility	Public

Filter Settings

Severity

Included: High, Medium, Low, Information

Excluded: None

Result State

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

Excluded: None

Assigned to

Included: All

Categories

Included:

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

Excluded:

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

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

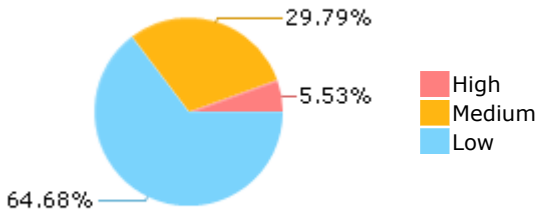
Results Limit

Results limit per query was set to 50

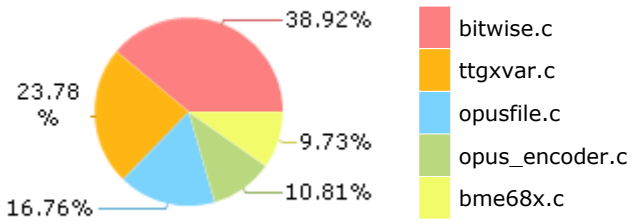
Selected Queries

Selected queries are listed in Result Summary

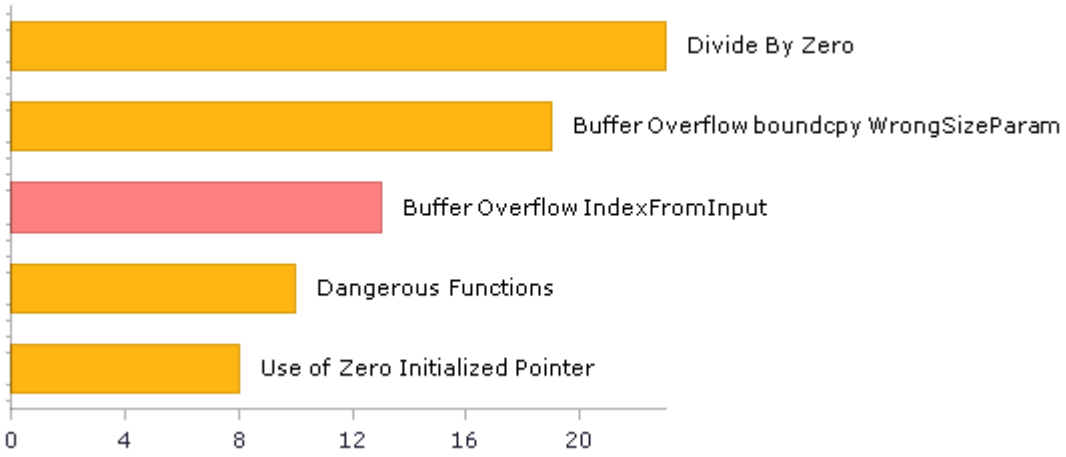
Result Summary



Most Vulnerable Files



Top 5 Vulnerabilities



Scan Summary - OWASP Top 10 2017

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

Category	Threat Agent	Exploitability	Weakness Prevalence	Weakness Detectability	Technical Impact	Business Impact	Issues Found	Best Fix Locations
A1-Injection	App. Specific	EASY	COMMON	EASY	SEVERE	App. Specific	78	34
A2-Broken Authentication	App. Specific	EASY	COMMON	AVERAGE	SEVERE	App. Specific	75	75
A3-Sensitive Data Exposure	App. Specific	AVERAGE	WIDESPREAD	AVERAGE	SEVERE	App. Specific	3	3
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	10	10
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	10	10
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	7	7
PCI DSS (3.2) - 6.5.2 - Buffer overflows	24	24
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.	6	6
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.	0	0
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.	75	75
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.	3	3
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.	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 - NIST SP 800-53

Category	Issues Found	Best Fix Locations
AC-12 Session Termination (P2)	0	0
AC-3 Access Enforcement (P1)	75	75
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)	3	3
SC-4 Information in Shared Resources (P1)	0	0
SC-5 Denial of Service Protection (P1)*	56	21
SC-8 Transmission Confidentiality and Integrity (P1)	0	0
SI-10 Information Input Validation (P1)*	19	19
SI-11 Error Handling (P2)*	7	7
SI-15 Information Output Filtering (P0)	0	0
SI-16 Memory Protection (P1)	7	7

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

Scan Summary - OWASP Mobile Top 10 2016

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

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

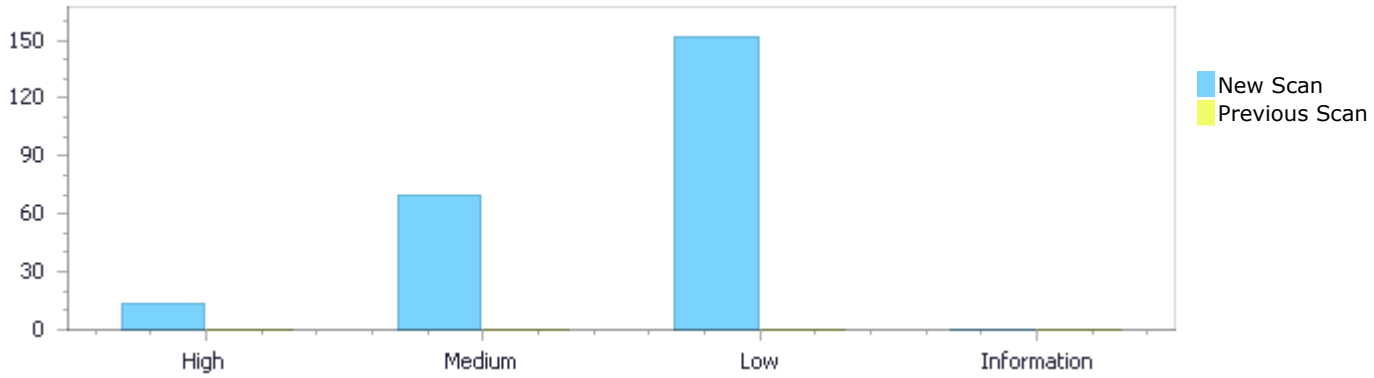
Scan Summary - Custom

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

Results Distribution By Status

First scan of the project

	High	Medium	Low	Information	Total
New Issues	13	70	152	0	235
Recurrent Issues	0	0	0	0	0
Total	13	70	152	0	235
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	13	70	152	0	235
Urgent	0	0	0	0	0
Proposed Not Exploitable	0	0	0	0	0
Total	13	70	152	0	235

Result Summary

Vulnerability Type	Occurrences	Severity
Buffer Overflow IndexFromInput	13	High
Divide By Zero	23	Medium
Buffer Overflow boundcpy WrongSizeParam	19	Medium
Dangerous Functions	10	Medium
Use of Zero Initialized Pointer	8	Medium

Memory Leak	4	Medium
Char Overflow	3	Medium
Buffer Overflow AddressOfLocalVarReturned	1	Medium
MemoryFree on StackVariable	1	Medium
Wrong Size t Allocation	1	Medium
Improper Resource Access Authorization	75	Low
NULL Pointer Dereference	37	Low
Unchecked Array Index	15	Low
Potential Off by One Error in Loops	7	Low
Unchecked Return Value	7	Low
Arithmenic Operation On Boolean	6	Low
Use of Insufficiently Random Values	3	Low
Heuristic 2nd Order Buffer Overflow read	1	Low
Use of Sizeof On a Pointer Type	1	Low

10 Most Vulnerable Files

High and Medium Vulnerabilities

File Name	Issues Found
Tasmota/opusfile.c	23
Tasmota/opus_encoder.c	16
Tasmota/bme68x.c	13
Tasmota/common.cpp	12
Tasmota/lv_canvas.c	8
Tasmota/ttgxvar.c	7
Tasmota/sbrfreq.c	2
Tasmota/ESPTimeHelper.cpp	1
Tasmota/t1parse.c	1

Scan Results Details

Buffer Overflow IndexFromInput

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

Categories

OWASP Top 10 2017: A1-Injection

Description

Buffer Overflow IndexFromInput\Path 1:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=1
Status	New

The size of the buffer used by read_all_field_data in i, at line 1280 of Tasmota/bme68x.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that bme68x_get_regs passes to reg_data, at line 235 of Tasmota/bme68x.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/bme68x.c	Tasmota/bme68x.c
Line	253	1340
Object	reg_data	i

Code Snippet

File Name Tasmota/bme68x.c

Method int8_t bme68x_get_regs(uint8_t reg_addr, uint8_t *reg_data, uint32_t len, struct bme68x_dev *dev)

```
....
253.          dev->intf_rslt = dev->read(reg_addr, reg_data, len, dev-
>intf_ptr);
```

File Name Tasmota/bme68x.c

Method static int8_t read_all_field_data(struct bme68x_data * const data[], struct bme68x_dev *dev)

```
....
1340.          data[i]->gas_wait = set_val[20 + data[i]->gas_index];
```

Buffer Overflow IndexFromInput\Path 2:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=2

Status	New

The size of the buffer used by read_all_field_data in i, at line 1280 of Tasmota/bme68x.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that bme68x_get_regs passes to reg_data, at line 235 of Tasmota/bme68x.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/bme68x.c	Tasmota/bme68x.c
Line	253	1339
Object	reg_data	i

Code Snippet

File Name Tasmota/bme68x.c

Method int8_t bme68x_get_regs(uint8_t reg_addr, uint8_t *reg_data, uint32_t len, struct bme68x_dev *dev)

```
....
253.          dev->intf_rslt = dev->read(reg_addr, reg_data, len, dev-
>intf_ptr);
```

File Name Tasmota/bme68x.c

Method static int8_t read_all_field_data(struct bme68x_data * const data[], struct bme68x_dev *dev)

```
....
1339.          data[i]->res_heat = set_val[10 + data[i]->gas_index];
```

Buffer Overflow IndexFromInput\Path 3:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=3
Status	New

The size of the buffer used by read_all_field_data in i, at line 1280 of Tasmota/bme68x.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that bme68x_get_regs passes to reg_data, at line 235 of Tasmota/bme68x.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/bme68x.c	Tasmota/bme68x.c
Line	253	1338
Object	reg_data	i

Code Snippet

File Name Tasmota/bme68x.c

Method int8_t bme68x_get_regs(uint8_t reg_addr, uint8_t *reg_data, uint32_t len, struct bme68x_dev *dev)

```
....
253.          dev->intf_rslt = dev->read(reg_addr, reg_data, len, dev-
>intf_ptr);
```

File Name Tasmota/bme68x.c

Method static int8_t read_all_field_data(struct bme68x_data * const data[], struct bme68x_dev *dev)

```
....
1338.          data[i]->idac = set_val[data[i]->gas_index];
```

Buffer Overflow IndexFromInput\Path 4:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=4
Status	New

The size of the buffer used by read_all_field_data in i, at line 1280 of Tasmota/bme68x.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that bme68x_get_regs passes to reg_data, at line 235 of Tasmota/bme68x.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/bme68x.c	Tasmota/bme68x.c
Line	253	1330
Object	reg_data	i

Code Snippet

File Name Tasmota/bme68x.c

Method int8_t bme68x_get_regs(uint8_t reg_addr, uint8_t *reg_data, uint32_t len, struct bme68x_dev *dev)

```
....
253.          dev->intf_rslt = dev->read(reg_addr, reg_data, len, dev-
>intf_ptr);
```

File Name Tasmota/bme68x.c

Method static int8_t read_all_field_data(struct bme68x_data * const data[], struct bme68x_dev *dev)

```
....
1330.              data[i]->status |= buff[off + 16] &
BME68X_HEAT_STAB_MSK;
```

Buffer Overflow IndexFromInput\Path 5:

Severity	High

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=5
Status	New

The size of the buffer used by read_all_field_data in i, at line 1280 of Tasmota/bme68x.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that bme68x_get_regs passes to reg_data, at line 235 of Tasmota/bme68x.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/bme68x.c	Tasmota/bme68x.c
Line	253	1329
Object	reg_data	i

Code Snippet

File Name Tasmota/bme68x.c

Method int8_t bme68x_get_regs(uint8_t reg_addr, uint8_t *reg_data, uint32_t len, struct bme68x_dev *dev)

```
....
253.          dev->intf_rslt = dev->read(reg_addr, reg_data, len, dev->intf_ptr);
```

File Name Tasmota/bme68x.c

Method static int8_t read_all_field_data(struct bme68x_data * const data[], struct bme68x_dev *dev)

```
....
1329.              data[i]->status |= buff[off + 16] & BME68X_GASM_VALID_MSK;
```

Buffer Overflow IndexFromInput\Path 6:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=6
Status	New

The size of the buffer used by read_all_field_data in i, at line 1280 of Tasmota/bme68x.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that bme68x_get_regs passes to reg_data, at line 235 of Tasmota/bme68x.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/bme68x.c	Tasmota/bme68x.c
Line	253	1335
Object	reg_data	i

Code Snippet

File Name	Tasmota/bme68x.c
Method	int8_t bme68x_get_regs(uint8_t reg_addr, uint8_t *reg_data, uint32_t len, struct bme68x_dev *dev)

```
....
253.          dev->intf_rslt = dev->read(reg_addr, reg_data, len, dev-
>intf_ptr);
```

File Name	Tasmota/bme68x.c
Method	static int8_t read_all_field_data(struct bme68x_data * const data[], struct bme68x_dev *dev)

```
....
1335.              data[i]->status |= buff[off + 14] &
BME68X_HEAT_STAB_MSK;
```

Buffer Overflow IndexFromInput\Path 7:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=7
Status	New

The size of the buffer used by read_all_field_data in i, at line 1280 of Tasmota/bme68x.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that bme68x_get_regs passes to reg_data, at line 235 of Tasmota/bme68x.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/bme68x.c	Tasmota/bme68x.c
Line	253	1334
Object	reg_data	i

Code Snippet

File Name	Tasmota/bme68x.c
Method	int8_t bme68x_get_regs(uint8_t reg_addr, uint8_t *reg_data, uint32_t len, struct bme68x_dev *dev)

```
....
253.          dev->intf_rslt = dev->read(reg_addr, reg_data, len, dev-
>intf_ptr);
```

File Name	Tasmota/bme68x.c
Method	static int8_t read_all_field_data(struct bme68x_data * const data[], struct bme68x_dev *dev)

```
....
1334.              data[i]->status |= buff[off + 14] &
BME68X_GASM_VALID_MSK;
```

Buffer Overflow IndexFromInput\Path 8:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=8
Status	New

The size of the buffer used by read_all_field_data in i, at line 1280 of Tasmota/bme68x.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that bme68x_get_regs passes to reg_data, at line 235 of Tasmota/bme68x.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/bme68x.c	Tasmota/bme68x.c
Line	253	1313
Object	reg_data	i

Code Snippet

File Name Tasmota/bme68x.c

Method int8_t bme68x_get_regs(uint8_t reg_addr, uint8_t *reg_data, uint32_t len, struct bme68x_dev *dev)

```
....
253.          dev->intf_rslt = dev->read(reg_addr, reg_data, len, dev-
>intf_ptr);
```

File Name Tasmota/bme68x.c

Method static int8_t read_all_field_data(struct bme68x_data * const data[], struct bme68x_dev *dev)

```
....
1313.           data[i]->meas_index = buff[off + 1];
```

Buffer Overflow IndexFromInput\Path 9:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=9
Status	New

The size of the buffer used by read_all_field_data in i, at line 1280 of Tasmota/bme68x.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that bme68x_get_regs passes to reg_data, at line 235 of Tasmota/bme68x.c, to overwrite the target buffer.

Source	Destination

File	Tasmota/bme68x.c	Tasmota/bme68x.c
Line	253	1312
Object	reg_data	i

Code Snippet
File Name Tasmota/bme68x.c
Method int8_t bme68x_get_regs(uint8_t reg_addr, uint8_t *reg_data, uint32_t len, struct bme68x_dev *dev)

```
....
253.          dev->intf_rslt = dev->read(reg_addr, reg_data, len, dev-
>intf_ptr);
```

File Name Tasmota/bme68x.c

Method static int8_t read_all_field_data(struct bme68x_data * const data[], struct bme68x_dev *dev)

```
....
1312.          data[i]->gas_index = buff[off] & BME68X_GAS_INDEX_MSK;
```

Buffer Overflow IndexFromInput\Path 10:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=10
Status	New

The size of the buffer used by read_all_field_data in BinaryExpr, at line 1280 of Tasmota/bme68x.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that bme68x_get_regs passes to reg_data, at line 235 of Tasmota/bme68x.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/bme68x.c	Tasmota/bme68x.c
Line	253	1340
Object	reg_data	BinaryExpr

Code Snippet
File Name Tasmota/bme68x.c
Method int8_t bme68x_get_regs(uint8_t reg_addr, uint8_t *reg_data, uint32_t len, struct bme68x_dev *dev)

```
....
253.          dev->intf_rslt = dev->read(reg_addr, reg_data, len, dev-
>intf_ptr);
```

File Name	Tasmota/bme68x.c
Method	static int8_t read_all_field_data(struct bme68x_data * const data[], struct bme68x_dev *dev)

```
....
1340.          data[i]->gas_wait = set_val[20 + data[i]->gas_index];
```

Buffer Overflow IndexFromInput\Path 11:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=11
Status	New

The size of the buffer used by read_all_field_data in BinaryExpr, at line 1280 of Tasmota/bme68x.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that bme68x_get_regs passes to reg_data, at line 235 of Tasmota/bme68x.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/bme68x.c	Tasmota/bme68x.c
Line	253	1339
Object	reg_data	BinaryExpr

Code Snippet

File Name	Tasmota/bme68x.c
Method	int8_t bme68x_get_regs(uint8_t reg_addr, uint8_t *reg_data, uint32_t len, struct bme68x_dev *dev)

```
....
253.          dev->intf_rslt = dev->read(reg_addr, reg_data, len, dev-
>intf_ptr);
```

File Name	Tasmota/bme68x.c
Method	static int8_t read_all_field_data(struct bme68x_data * const data[], struct bme68x_dev *dev)

```
....
1339.          data[i]->res_heat = set_val[10 + data[i]->gas_index];
```

Buffer Overflow IndexFromInput\Path 12:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=12
Status	New

The size of the buffer used by read_all_field_data in gas_index, at line 1280 of Tasmota/bme68x.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that bme68x_get_regs passes to reg_data, at line 235 of Tasmota/bme68x.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/bme68x.c	Tasmota/bme68x.c
Line	253	1338
Object	reg_data	gas_index

Code Snippet

File Name Tasmota/bme68x.c

Method int8_t bme68x_get_regs(uint8_t reg_addr, uint8_t *reg_data, uint32_t len, struct bme68x_dev *dev)

```
....
253.          dev->intf_rslt = dev->read(reg_addr, reg_data, len, dev->intf_ptr);
```

File Name Tasmota/bme68x.c

Method static int8_t read_all_field_data(struct bme68x_data * const data[], struct bme68x_dev *dev)

```
....
1338.          data[i]->idac = set_val[data[i]->gas_index];
```

Buffer Overflow IndexFromInput\Path 13:

Severity	High
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=13
Status	New

The size of the buffer used by read_all_field_data in i, at line 1280 of Tasmota/bme68x.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that bme68x_get_regs passes to reg_data, at line 235 of Tasmota/bme68x.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/bme68x.c	Tasmota/bme68x.c
Line	253	1311
Object	reg_data	i

Code Snippet

File Name Tasmota/bme68x.c

Method int8_t bme68x_get_regs(uint8_t reg_addr, uint8_t *reg_data, uint32_t len, struct bme68x_dev *dev)

```
....
253.          dev->intf_rslt = dev->read(reg_addr, reg_data, len, dev-
>intf_ptr);
```

File Name	Tasmota/bme68x.c
Method	static int8_t read_all_field_data(struct bme68x_data * const data[], struct bme68x_dev *dev)

```
....
1311.          data[i]->status = buff[off] & BME68X_NEW_DATA_MSK;
```

Divide By Zero

Query Path:
CPP\Cx\CPP Medium Threat\Divide By Zero Version:1

Description

Divide By Zero\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=25
Status	New

The application performs an illegal operation in lv_canvas_blur_hor, in Tasmota/lv_canvas.c. In line 226, the program attempts to divide by r, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input r in lv_canvas_blur_hor of Tasmota/lv_canvas.c, at line 226.

	Source	Destination
File	Tasmota/lv_canvas.c	Tasmota/lv_canvas.c
Line	305	305
Object	r	r

Code Snippet

File Name	Tasmota/lv_canvas.c
Method	void lv_canvas_blur_hor(lv_obj_t * obj, const lv_area_t * area, uint16_t r)

```
....
305.                  c.ch.red = rsum / r;
```

Divide By Zero\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=26
Status	New

The application performs an illegal operation in lv_canvas_blur_hor, in Tasmota/lv_canvas.c. In line 226, the program attempts to divide by r, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input r in lv_canvas_blur_hor of Tasmota/lv_canvas.c, at line 226.

	Source	Destination
File	Tasmota/lv_canvas.c	Tasmota/lv_canvas.c
Line	307	307
Object	r	r

Code Snippet
File Name Tasmota/lv_canvas.c
Method void lv_canvas_blur_hor(lv_obj_t * obj, const lv_area_t * area, uint16_t r)

```
....
307.                    uint8_t gtmp = gsum / r;
```

Divide By Zero\Path 3:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=27
Status New

The application performs an illegal operation in lv_canvas_blur_hor, in Tasmota/lv_canvas.c. In line 226, the program attempts to divide by r, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input r in lv_canvas_blur_hor of Tasmota/lv_canvas.c, at line 226.

	Source	Destination
File	Tasmota/lv_canvas.c	Tasmota/lv_canvas.c
Line	313	313
Object	r	r

Code Snippet
File Name Tasmota/lv_canvas.c
Method void lv_canvas_blur_hor(lv_obj_t * obj, const lv_area_t * area, uint16_t r)

```
....
313.                    c.ch.blue = bsum / r;
```

Divide By Zero\Path 4:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=28
Status New

The application performs an illegal operation in lv_canvas_blur_hor, in Tasmota/lv_canvas.c. In line 226, the program attempts to divide by r, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input r in lv_canvas_blur_hor of Tasmota/lv_canvas.c, at line 226.

	Source	Destination
File	Tasmota/lv_canvas.c	Tasmota/lv_canvas.c
Line	314	314
Object	r	r

Code Snippet
File Name Tasmota/lv_canvas.c
Method void lv_canvas_blur_hor(lv_obj_t * obj, const lv_area_t * area, uint16_t r)

```
....
314.                    if(has_alpha) opa = asum / r;
```

Divide By Zero\Path 5:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=29
Status New

The application performs an illegal operation in lv_canvas_blur_ver, in Tasmota/lv_canvas.c. In line 354, the program attempts to divide by r, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input r in lv_canvas_blur_ver of Tasmota/lv_canvas.c, at line 354.

	Source	Destination
File	Tasmota/lv_canvas.c	Tasmota/lv_canvas.c
Line	433	433
Object	r	r

Code Snippet
File Name Tasmota/lv_canvas.c
Method void lv_canvas_blur_ver(lv_obj_t * obj, const lv_area_t * area, uint16_t r)

```
....
433.                    c.ch.red = rsum / r;
```

Divide By Zero\Path 6:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=30

Status	New

The application performs an illegal operation in lv_canvas_blur_ver, in Tasmota/lv_canvas.c. In line 354, the program attempts to divide by r, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input r in lv_canvas_blur_ver of Tasmota/lv_canvas.c, at line 354.

	Source	Destination
File	Tasmota/lv_canvas.c	Tasmota/lv_canvas.c
Line	435	435
Object	r	r

Code Snippet
File Name Tasmota/lv_canvas.c
Method void lv_canvas_blur_ver(lv_obj_t * obj, const lv_area_t * area, uint16_t r)

```
....
435.                    uint8_t gtmp = gsum / r;
```

Divide By Zero\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=31
Status	New

The application performs an illegal operation in lv_canvas_blur_ver, in Tasmota/lv_canvas.c. In line 354, the program attempts to divide by r, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input r in lv_canvas_blur_ver of Tasmota/lv_canvas.c, at line 354.

	Source	Destination
File	Tasmota/lv_canvas.c	Tasmota/lv_canvas.c
Line	441	441
Object	r	r

Code Snippet
File Name Tasmota/lv_canvas.c
Method void lv_canvas_blur_ver(lv_obj_t * obj, const lv_area_t * area, uint16_t r)

```
....
441.                    c.ch.blue = bsum / r;
```

Divide By Zero\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=500

	42&pathid=32
Status	New

The application performs an illegal operation in lv_canvas_blur_ver, in Tasmota/lv_canvas.c. In line 354, the program attempts to divide by r, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input r in lv_canvas_blur_ver of Tasmota/lv_canvas.c, at line 354.

	Source	Destination
File	Tasmota/lv_canvas.c	Tasmota/lv_canvas.c
Line	442	442
Object	r	r

Code Snippet
File Name Tasmota/lv_canvas.c
Method void lv_canvas_blur_ver(lv_obj_t * obj, const lv_area_t * area, uint16_t r)

```
....
442.                    if(has_alpha) opa = asum / r;
```

Divide By Zero\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=33
Status	New

The application performs an illegal operation in compute_stereo_width, in Tasmota/opus_encoder.c. In line 636, the program attempts to divide by frame_size, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input frame_size in compute_stereo_width of Tasmota/opus_encoder.c, at line 636.

	Source	Destination
File	Tasmota/opus_encoder.c	Tasmota/opus_encoder.c
Line	645	645
Object	frame_size	frame_size

Code Snippet
File Name Tasmota/opus_encoder.c
Method opus_val16 compute_stereo_width(const opus_val16 *pcm, int frame_size, opus_int32 Fs, StereoWidthState *mem)

```
....
645.     frame_rate = Fs/frame_size;
```

Divide By Zero\Path 10:

Severity	Medium
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=34
Status	New

The application performs an illegal operation in encode_multiframe_packet, in Tasmota/opus_encoder.c. In line 939, the program attempts to divide by nb_frames, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input nb_frames in encode_multiframe_packet of Tasmota/opus_encoder.c, at line 939.

	Source	Destination
File	Tasmota/opus_encoder.c	Tasmota/opus_encoder.c
Line	972	972
Object	nb_frames	nb_frames

Code Snippet
File Name Tasmota/opus_encoder.c
Method static opus_int32 encode_multiframe_packet(OpusEncoder *st,

```
....
972.     bytes_per_frame = IMIN(1276, 1+(repacketize_len-
max_header_bytes)/nb_frames);
```

Divide By Zero\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=35
Status	New

The application performs an illegal operation in compute_redundancy_bytes, in Tasmota/opus_encoder.c. In line 1038, the program attempts to divide by BinaryExpr, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input BinaryExpr in compute_redundancy_bytes of Tasmota/opus_encoder.c, at line 1038.

	Source	Destination
File	Tasmota/opus_encoder.c	Tasmota/opus_encoder.c
Line	1056	1056
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name Tasmota/opus_encoder.c
Method static int compute_redundancy_bytes(opus_int32 max_data_bytes, opus_int32 bitrate_bps, int frame_rate, int channels)

```
....
1056.     redundancy_bytes_cap =
(available_bits*240/(240+48000/frame_rate) + base_bits)/8;
```

Divide By Zero\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=36
Status	New

The application performs an illegal operation in compute_redundancy_bytes, in Tasmota/opus_encoder.c. In line 1038, the program attempts to divide by frame_rate, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input frame_rate in compute_redundancy_bytes of Tasmota/opus_encoder.c, at line 1038.

	Source	Destination
File	Tasmota/opus_encoder.c	Tasmota/opus_encoder.c
Line	1056	1056
Object	frame_rate	frame_rate

Code Snippet

File Name Tasmota/opus_encoder.c

Method static int compute_redundancy_bytes(opus_int32 max_data_bytes, opus_int32 bitrate_bps, int frame_rate, int channels)

```
....
1056.     redundancy_bytes_cap =
(available_bits*240/(240+48000/frame_rate) + base_bits)/8;
```

Divide By Zero\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=37
Status	New

The application performs an illegal operation in opus_encode_native, in Tasmota/opus_encoder.c. In line 1066, the program attempts to divide by frame_size, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input frame_size in opus_encode_native of Tasmota/opus_encoder.c, at line 1066.

	Source	Destination
File	Tasmota/opus_encoder.c	Tasmota/opus_encoder.c
Line	1211	1211
Object	frame_size	frame_size

Code Snippet

File Name Tasmota/opus_encoder.c

Method opus_int32 opus_encode_native(OpusEncoder *st, const opus_val16 *pcm, int frame_size,

```
....
1211.       frame_rate = st->Fs/frame_size;
```

Divide By Zero\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=38
Status	New

The application performs an illegal operation in opus_encode_native, in Tasmota/opus_encoder.c. In line 1066, the program attempts to divide by frame_size, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input frame_size in opus_encode_native of Tasmota/opus_encoder.c, at line 1066.

	Source	Destination
File	Tasmota/opus_encoder.c	Tasmota/opus_encoder.c
Line	1216	1216
Object	frame_size	frame_size

Code Snippet
File Name Tasmota/opus_encoder.c
Method opus_int32 opus_encode_native(OpusEncoder *st, const opus_val16 *pcm, int frame_size,

```
....
1216.          int frame_rate12 = 12*st->Fs/frame_size;
```

Divide By Zero\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=39
Status	New

The application performs an illegal operation in opus_encode_native, in Tasmota/opus_encoder.c. In line 1066, the program attempts to divide by frame_rate, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input frame_rate in opus_encode_native of Tasmota/opus_encoder.c, at line 1066.

	Source	Destination
File	Tasmota/opus_encoder.c	Tasmota/opus_encoder.c
Line	1256	1256
Object	frame_rate	frame_rate

Code Snippet

File Name	Tasmota/opus_encoder.c
Method	opus_int32 opus_encode_native(OpusEncoder *st, const opus_val16 *pcm, int frame_size,

```
....
1256.               num_multiframes = 50/frame_rate;
```

Divide By Zero\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=40
Status	New

The application performs an illegal operation in opus_encode_native, in Tasmota/opus_encoder.c. In line 1066, the program attempts to divide by frame_size, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input frame_size in opus_encode_native of Tasmota/opus_encoder.c, at line 1066.

	Source	Destination
File	Tasmota/opus_encoder.c	Tasmota/opus_encoder.c
Line	1293	1293
Object	frame_size	frame_size

Code Snippet

File Name	Tasmota/opus_encoder.c
Method	opus_int32 opus_encode_native(OpusEncoder *st, const opus_val16 *pcm, int frame_size,

```
....
1293.      equiv_rate = compute_equiv_rate(st->bitrate_bps, st-
>channels, st->Fs/frame_size,
```

Divide By Zero\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=41
Status	New

The application performs an illegal operation in opus_encode_native, in Tasmota/opus_encoder.c. In line 1066, the program attempts to divide by frame_size, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input frame_size in opus_encode_native of Tasmota/opus_encoder.c, at line 1066.

	Source	Destination
File	Tasmota/opus_encoder.c	Tasmota/opus_encoder.c
Line	1336	1336

Object	frame_size	frame_size

Code Snippet
File Name Tasmota/opus_encoder.c
Method opus_int32 opus_encode_native(OpusEncoder *st, const opus_val16 *pcm, int frame_size,

```
....
1336.      equiv_rate = compute_equiv_rate(st->bitrate_bps, st-
>stream_channels, st->Fs/frame_size,
```

Divide By Zero\Path 18:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=42
Status New

The application performs an illegal operation in opus_encode_native, in Tasmota/opus_encoder.c. In line 1066, the program attempts to divide by frame_size, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input frame_size in opus_encode_native of Tasmota/opus_encoder.c, at line 1066.

	Source	Destination
File	Tasmota/opus_encoder.c	Tasmota/opus_encoder.c
Line	1445	1445
Object	frame_size	frame_size

Code Snippet
File Name Tasmota/opus_encoder.c
Method opus_int32 opus_encode_native(OpusEncoder *st, const opus_val16 *pcm, int frame_size,

```
....
1445.      equiv_rate = compute_equiv_rate(st->bitrate_bps, st-
>stream_channels, st->Fs/frame_size,
```

Divide By Zero\Path 19:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=43
Status New

The application performs an illegal operation in opus_encode_native, in Tasmota/opus_encoder.c. In line 1066, the program attempts to divide by frame_size, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input frame_size in opus_encode_native of Tasmota/opus_encoder.c, at line 1066.

	Source	Destination
File	Tasmota/opus_encoder.c	Tasmota/opus_encoder.c
Line	1807	1807
Object	frame_size	frame_size

Code Snippet
File Name Tasmota/opus_encoder.c
Method opus_int32 opus_encode_native(OpusEncoder *st, const opus_val16 *pcm, int frame_size,

```
....
1807.                opus_int32 maxBitRate =
compute_silk_rate_for_hybrid(st->silk_mode.maxBits*st->Fs / frame_size,
```

Divide By Zero\Path 20:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=44
Status New

The application performs an illegal operation in opus_encode_native, in Tasmota/opus_encoder.c. In line 1066, the program attempts to divide by frame_size, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input frame_size in opus_encode_native of Tasmota/opus_encoder.c, at line 1066.

	Source	Destination
File	Tasmota/opus_encoder.c	Tasmota/opus_encoder.c
Line	1871	1871
Object	frame_size	frame_size

Code Snippet
File Name Tasmota/opus_encoder.c
Method opus_int32 opus_encode_native(OpusEncoder *st, const opus_val16 *pcm, int frame_size,

```
....
1871.             data[-1] = gen_toc(st->mode, st->Fs/frame_size,
curr_bandwidth, st->stream_channels);
```

Divide By Zero\Path 21:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=45
Status New

The application performs an illegal operation in opus_encode_native, in Tasmota/opus_encoder.c. In line 1066, the program attempts to divide by frame_size, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input frame_size in opus_encode_native of Tasmota/opus_encoder.c, at line 1066.

	Source	Destination
File	Tasmota/opus_encoder.c	Tasmota/opus_encoder.c
Line	2130	2130
Object	frame_size	frame_size

Code Snippet
File Name Tasmota/opus_encoder.c
Method opus_int32 opus_encode_native(OpusEncoder *st, const opus_val16 *pcm, int frame_size,

```
....
2130.      data[0] = gen_toc(st->mode, st->Fs/frame_size,
curr_bandwidth, st->stream_channels);
```

Divide By Zero\Path 22:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=46
Status New

The application performs an illegal operation in opus_encode_native, in Tasmota/opus_encoder.c. In line 1066, the program attempts to divide by frame_size, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input frame_size in opus_encode_native of Tasmota/opus_encoder.c, at line 1066.

	Source	Destination
File	Tasmota/opus_encoder.c	Tasmota/opus_encoder.c
Line	2151	2151
Object	frame_size	frame_size

Code Snippet
File Name Tasmota/opus_encoder.c
Method opus_int32 opus_encode_native(OpusEncoder *st, const opus_val16 *pcm, int frame_size,

```
....
2151.            data[0] = gen_toc(st->mode, st->Fs/frame_size,
curr_bandwidth, st->stream_channels);
```

Divide By Zero\Path 23:

Severity Medium
Result State To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=47
Status	New

The application performs an illegal operation in CalcFreqNoise, in Tasmota/sbrfreq.c. In line 370, the program attempts to divide by BinaryExpr, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input BinaryExpr in CalcFreqNoise of Tasmota/sbrfreq.c, at line 370.

	Source	Destination
File	Tasmota/sbrfreq.c	Tasmota/sbrfreq.c
Line	386	386
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name Tasmota/sbrfreq.c
Method static int CalcFreqNoise(unsigned char *freqNoise, unsigned char *freqLow, int nLow, int kStart, int k2, int noiseBands)

```
....
386.               i = iLast + (nLow - iLast) / (nQ + 1 - k);      /*
truncating division */
```

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=1050052&projectid=50042&pathid=56
Status	New

The size of the buffer used by TfLiteIntArrayCopy in src, at line 68 of Tasmota/common.cpp, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that TfLiteIntArrayCopy passes to src, at line 68 of Tasmota/common.cpp, to overwrite the target buffer.

	Source	Destination
File	Tasmota/common.cpp	Tasmota/common.cpp
Line	72	72
Object	src	src

Code Snippet

File Name	Tasmota/common.cpp
Method	TfLiteIntArray* TfLiteIntArrayCopy(const TfLiteIntArray* src) {

```
....
72.      memcpy(ret->data, src->data, src->size * sizeof(int));
```

Buffer Overflow boundcpy WrongSizeParam\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=57
Status	New

The size of the buffer used by TfLiteIntArrayCopy in int, at line 68 of Tasmota/common.cpp, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that TfLiteIntArrayCopy passes to int, at line 68 of Tasmota/common.cpp, to overwrite the target buffer.

	Source	Destination
File	Tasmota/common.cpp	Tasmota/common.cpp
Line	72	72
Object	int	int

Code Snippet	
File Name	Tasmota/common.cpp
Method	TfLiteIntArray* TfLiteIntArrayCopy(const TfLiteIntArray* src) {

```
....
72.      memcpy(ret->data, src->data, src->size * sizeof(int));
```

Buffer Overflow boundcpy WrongSizeParam\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=58
Status	New

The size of the buffer used by op_make_decode_ready in channel_count, at line 1358 of Tasmota/opusfile.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that op_make_decode_ready passes to channel_count, at line 1358 of Tasmota/opusfile.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	1387	1387
Object	channel_count	channel_count

Code Snippet	
File Name	Tasmota/opusfile.c
Method	static int op_make_decode_ready(OggOpusFile *_of){

```
....
1387.       memcpy(_of->od_mapping,head->mapping,sizeof(*head-
>mapping)*channel_count);
```

Buffer Overflow boundcpy WrongSizeParam\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=59
Status	New

The size of the buffer used by op_make_decode_ready in head, at line 1358 of Tasmota/opusfile.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that op_make_decode_ready passes to head, at line 1358 of Tasmota/opusfile.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	1387	1387
Object	head	head

Code Snippet
File Name Tasmota/opusfile.c
Method static int op_make_decode_ready(OggOpusFile *_of){

```
....
1387.       memcpy(_of->od_mapping,head->mapping,sizeof(*head-
>mapping)*channel_count);
```

Buffer Overflow boundcpy WrongSizeParam\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=60
Status	New

The size of the buffer used by op_open_seekable2 in start_op_count, at line 1431 of Tasmota/opusfile.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that op_open_seekable2 passes to start_op_count, at line 1431 of Tasmota/opusfile.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	1457	1457
Object	start_op_count	start_op_count

Code Snippet
File Name Tasmota/opusfile.c

Method	static int op_open_seekable2(OggOpusFile *_of){

```
....
1457.    memcpy(op_start,_of->op,sizeof(*op_start)*start_op_count);
```

Buffer Overflow boundcpy WrongSizeParam\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=61
Status	New

The size of the buffer used by op_open_seekable2 in op_start, at line 1431 of Tasmota/opusfile.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that op_open_seekable2 passes to op_start, at line 1431 of Tasmota/opusfile.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	1457	1457
Object	op_start	op_start

Code Snippet	
File Name	Tasmota/opusfile.c
Method	static int op_open_seekable2(OggOpusFile *_of){

```
....
1457.    memcpy(op_start,_of->op,sizeof(*op_start)*start_op_count);
```

Buffer Overflow boundcpy WrongSizeParam\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=62
Status	New

The size of the buffer used by op_open_seekable2 in start_op_count, at line 1431 of Tasmota/opusfile.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that op_open_seekable2 passes to start_op_count, at line 1431 of Tasmota/opusfile.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	1469	1469
Object	start_op_count	start_op_count

Code Snippet	
File Name	Tasmota/opusfile.c
Method	static int op_open_seekable2(OggOpusFile *_of){

```
....
1469.    memcpy(_of->op,op_start,sizeof(*_of->op)*start_op_count);
```

Buffer Overflow boundcpy WrongSizeParam\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=63
Status	New

The size of the buffer used by op_open_seekable2 in _of, at line 1431 of Tasmota/opusfile.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that op_open_seekable2 passes to _of, at line 1431 of Tasmota/opusfile.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	1469	1469
Object	_of	_of

Code Snippet
File Name Tasmota/opusfile.c
Method static int op_open_seekable2(OggOpusFile *_of){

```
....
1469.    memcpy(_of->op,op_start,sizeof(*_of->op)*start_op_count);
```

Buffer Overflow boundcpy WrongSizeParam\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=64
Status	New

The size of the buffer used by op_open1 in _initial_bytes, at line 1519 of Tasmota/opusfile.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that op_open1 passes to _initial_bytes, at line 1519 of Tasmota/opusfile.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	1545	1545
Object	_initial_bytes	_initial_bytes

Code Snippet
File Name Tasmota/opusfile.c
Method static int op_open1(OggOpusFile *_of,

```
....
1545.      memcpy(buffer,_initial_data,_initial_bytes*sizeof(*buffer));
```

Buffer Overflow boundcpy WrongSizeParam\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=65
Status	New

The size of the buffer used by op_open1 in buffer, at line 1519 of Tasmota/opusfile.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that op_open1 passes to buffer, at line 1519 of Tasmota/opusfile.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	1545	1545
Object	buffer	buffer

Code Snippet
File Name Tasmota/opusfile.c
Method static int op_open1(OggOpusFile *_of,

```
....
1545.      memcpy(buffer,_initial_data,_initial_bytes*sizeof(*buffer));
```

Buffer Overflow boundcpy WrongSizeParam\Path 11:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=66
Status	New

The size of the buffer used by op_stereo_filter in _nsamples, at line 3060 of Tasmota/opusfile.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that op_stereo_filter passes to _nsamples, at line 3060 of Tasmota/opusfile.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	3064	3064
Object	_nsamples	_nsamples

Code Snippet
File Name Tasmota/opusfile.c
Method static int op_stereo_filter(OggOpusFile *_of,void *_dst,int _dst_sz,

```
....
3064.    if(_nchannels==2)memcpy(_dst,_src,_nsamples*2*sizeof(*_src));
```

Buffer Overflow boundcpy WrongSizeParam\Path 12:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=67
Status	New

The size of the buffer used by op_stereo_filter in _src, at line 3060 of Tasmota/opusfile.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that op_stereo_filter passes to _src, at line 3060 of Tasmota/opusfile.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	3064	3064
Object	_src	_src

Code Snippet
File Name Tasmota/opusfile.c
Method static int op_stereo_filter(OggOpusFile *_of,void *_dst,int _dst_sz,

```
....
3064.    if(_nchannels==2)memcpy(_dst,_src,_nsamples*2*sizeof(*_src));
```

Buffer Overflow boundcpy WrongSizeParam\Path 13:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=68
Status	New

The size of the buffer used by op_read_native in nsamples, at line 2827 of Tasmota/opusfile.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that op_read_native passes to nsamples, at line 2827 of Tasmota/opusfile.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	2850	2850
Object	nsamples	nsamples

Code Snippet
File Name Tasmota/opusfile.c
Method static int op_read_native(OggOpusFile *_of,

```
....
2850.             sizeof(*_pcm)*nchannels*nsamples);
```

Buffer Overflow boundcpy WrongSizeParam\Path 14:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=69
Status	New

The size of the buffer used by op_read_native in nchannels, at line 2827 of Tasmota/opusfile.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that op_read_native passes to nchannels, at line 2827 of Tasmota/opusfile.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	2850	2850
Object	nchannels	nchannels

Code Snippet
File Name Tasmota/opusfile.c
Method static int op_read_native(OggOpusFile *_of,

```
....
2850.             sizeof(*_pcm)*nchannels*nsamples);
```

Buffer Overflow boundcpy WrongSizeParam\Path 15:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=70
Status	New

The size of the buffer used by op_read_native in _pcm, at line 2827 of Tasmota/opusfile.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that op_read_native passes to _pcm, at line 2827 of Tasmota/opusfile.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	2850	2850
Object	_pcm	_pcm

Code Snippet
File Name Tasmota/opusfile.c
Method static int op_read_native(OggOpusFile *_of,

```
....
2850.              sizeof(*_pcm)*nchannels*nsamples);
```

Buffer Overflow boundcpy WrongSizeParam\Path 16:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=71
Status	New

The size of the buffer used by op_read_native in nchannels, at line 2827 of Tasmota/opusfile.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that op_read_native passes to nchannels, at line 2827 of Tasmota/opusfile.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	2920	2920
Object	nchannels	nchannels

Code Snippet
File Name Tasmota/opusfile.c
Method static int op_read_native(OggOpusFile *_of,

```
....
2920.                 sizeof(*_pcm)*trimmed_duration*nchannels);
```

Buffer Overflow boundcpy WrongSizeParam\Path 17:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=72
Status	New

The size of the buffer used by op_read_native in trimmed_duration, at line 2827 of Tasmota/opusfile.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that op_read_native passes to trimmed_duration, at line 2827 of Tasmota/opusfile.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	2920	2920
Object	trimmed_duration	trimmed_duration

Code Snippet
File Name Tasmota/opusfile.c
Method static int op_read_native(OggOpusFile *_of,

```
....
2920.                 sizeof(*_pcm)*trimmed_duration*nchannels);
```

Buffer Overflow boundcpy WrongSizeParam\Path 18:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=73
Status	New

The size of the buffer used by op_read_native in _pcm, at line 2827 of Tasmota/opusfile.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that op_read_native passes to _pcm, at line 2827 of Tasmota/opusfile.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	2920	2920
Object	_pcm	_pcm

Code Snippet
File Name Tasmota/opusfile.c
Method static int op_read_native(OggOpusFile *_of,

```
....
2920.                 sizeof(*_pcm)*trimmed_duration*nchannels);
```

Buffer Overflow boundcpy WrongSizeParam\Path 19:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=74
Status	New

The size of the buffer used by TfLiteTensorCopy in src, at line 202 of Tasmota/common.cpp, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that TfLiteTensorCopy passes to src, at line 202 of Tasmota/common.cpp, to overwrite the target buffer.

	Source	Destination
File	Tasmota/common.cpp	Tasmota/common.cpp
Line	210	210
Object	src	src

Code Snippet
File Name Tasmota/common.cpp
Method TfLiteStatus TfLiteTensorCopy(const TfLiteTensor* src, TfLiteTensor* dst) {

```
....
210.    memcpy(dst->data.raw, src->data.raw, src->bytes);
```

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=1050052&projectid=50042&pathid=139
Status	New

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

	Source	Destination
File	Tasmota/common.cpp	Tasmota/common.cpp
Line	72	72
Object	memcpy	memcpy

Code Snippet
File Name Tasmota/common.cpp
Method TfLiteIntArray* TfLiteIntArrayCopy(const TfLiteIntArray* src) {

```
....
72.      memcpy(ret->data, src->data, src->size * sizeof(int));
```

Dangerous Functions\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=140
Status	New

The dangerous function, memcpy, was found in use at line 202 in Tasmota/common.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	Tasmota/common.cpp	Tasmota/common.cpp
Line	210	210

Object	memcpy	memcpy

Code Snippet
File Name Tasmota/common.cpp
Method TfLiteStatus TfLiteTensorCopy(const TfLiteTensor* src, TfLiteTensor* dst) {

```
....
210.    memcpy(dst->data.raw, src->data.raw, src->bytes);
```

Dangerous Functions\Path 3:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=141
Status New

The dangerous function, memcpy, was found in use at line 73 in Tasmota/opusfile.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	96	96
Object	memcpy	memcpy

Code Snippet
File Name Tasmota/opusfile.c
Method int op_test(OpusHead *_head,

```
....
96.      memcpy(data,_initial_data,_initial_bytes);
```

Dangerous Functions\Path 4:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=142
Status New

The dangerous function, memcpy, was found in use at line 1358 in Tasmota/opusfile.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	1387	1387
Object	memcpy	memcpy

Code Snippet	
File Name	Tasmota/opusfile.c
Method	static int op_make_decode_ready(OggOpusFile *_of){

```
....
1387.       memcpy(_of->od_mapping,head->mapping,sizeof(*head-
>mapping)*channel_count);
```

Dangerous Functions\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=143
Status	New

The dangerous function, memcpy, was found in use at line 1431 in Tasmota/opusfile.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	1457	1457
Object	memcpy	memcpy

Code Snippet	
File Name	Tasmota/opusfile.c
Method	static int op_open_seekable2(OggOpusFile *_of){

```
....
1457.    memcpy(op_start,_of->op,sizeof(*op_start)*start_op_count);
```

Dangerous Functions\Path 6:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=144
Status	New

The dangerous function, memcpy, was found in use at line 1431 in Tasmota/opusfile.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	1469	1469
Object	memcpy	memcpy

Code Snippet	
File Name	Tasmota/opusfile.c

Method	static int op_open_seekable2(OggOpusFile *_of){

```
....
1469.    memcpy(_of->op,op_start,sizeof(*_of->op)*start_op_count);
```

Dangerous Functions\Path 7:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=145
Status	New

The dangerous function, memcpy, was found in use at line 1519 in Tasmota/opusfile.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	1545	1545
Object	memcpy	memcpy

Code Snippet	
File Name	Tasmota/opusfile.c
Method	static int op_open1(OggOpusFile *_of,

```
....
1545.      memcpy(buffer,_initial_data,_initial_bytes*sizeof(*buffer));
```

Dangerous Functions\Path 8:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=146
Status	New

The dangerous function, memcpy, was found in use at line 2827 in Tasmota/opusfile.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	2849	2849
Object	memcpy	memcpy

Code Snippet	
File Name	Tasmota/opusfile.c
Method	static int op_read_native(OggOpusFile *_of,

```
....
2849.           memcpy(_pcm,_of->od_buffer+nchannels*od_buffer_pos,
```

Dangerous Functions\Path 9:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=147
Status	New

The dangerous function, memcpy, was found in use at line 3060 in Tasmota/opusfile.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	3064	3064
Object	memcpy	memcpy

Code Snippet
File Name Tasmota/opusfile.c
Method static int op_stereo_filter(OggOpusFile *_of,void *_dst,int _dst_sz,

```
....
3064.    if(_nchannels==2)memcpy(_dst,_src,_nsamples*2*sizeof(*_src));
```

Dangerous Functions\Path 10:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=148
Status	New

The dangerous function, realloc, was found in use at line 218 in Tasmota/common.cpp file. Such functions may expose information and allow an attacker to get full control over the host machine.

	Source	Destination
File	Tasmota/common.cpp	Tasmota/common.cpp
Line	235	235
Object	realloc	realloc

Code Snippet
File Name Tasmota/common.cpp
Method void TfLiteTensorResizeMaybeCopy(size_t num_bytes, TfLiteTensor* tensor,

```
....
235.        tensor->data.data = (char*)realloc(tensor->data.data,
num_bytes);
```

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=1050052&projectid=50042&pathid=153
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by cvt_deltas at Tasmota/ttgxvar.c in line 3194.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	3421
Object	mmvar	cvt_deltas

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2045.      FT_MM_Var*           mmvar = NULL;
```

File Name Tasmota/ttgxvar.c
Method tt_face_vary_cvt(TT_Face face,

```
....
3421.            cvt_deltas[j] = old_cvt_delta + FT_MulFix( deltas[j],
apply );
```

Use of Zero Initialized Pointer\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=154

Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by cvt_deltas at Tasmota/ttgxvar.c in line 3194.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	3463
Object	mmvar	cvt_deltas

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2045.       FT_MM_Var*           mmvar = NULL;
```

File Name Tasmota/ttgxvar.c
Method tt_face_vary_cvt(TT_Face face,

```
....
3463.             cvt_deltas[pindex] = old_cvt_delta + FT_MulFix(
deltas[j], apply );
```

Use of Zero Initialized Pointer\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=155
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by cvt at Tasmota/ttgxvar.c in line 3194.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	3430
Object	mmvar	cvt

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2045.       FT_MM_Var*           mmvar = NULL;
```

File Name Tasmota/ttgxvar.c

Method tt_face_vary_cvt(TT_Face face,

```
....
3430.                           ( FT_fdot6ToFixed( face->cvt[j] ) +
```

Use of Zero Initialized Pointer\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=156
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by cvt at Tasmota/ttgxvar.c in line 3194.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	3472
Object	mmvar	cvt

Code Snippet

File Name Tasmota/ttgxvar.c

Method TT_Get_MM_Var(TT_Face face,

```
....
2045.      FT_MM_Var*           mmvar = NULL;
```

File Name Tasmota/ttgxvar.c

Method tt_face_vary_cvt(TT_Face face,

```
....
3472.                           ( FT_fdot6ToFixed( face->cvt[pindex] ) +
```

Use of Zero Initialized Pointer\Path 5:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=157
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by mmvar at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2213
Object	mmvar	mmvar

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2045.      FT_MM_Var*           mmvar = NULL;
....
2213.          (FT_UShort*)( (char*)mmvar + mmvar_size );
```

Use of Zero Initialized Pointer\Path 6:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=158
Status New

The variable declared in base_dict at Tasmota/t1parse.c in line 260 is not initialized when it is used by private_dict at Tasmota/t1parse.c in line 260.

	Source	Destination
File	Tasmota/t1parse.c	Tasmota/t1parse.c
Line	462	460
Object	base_dict	private_dict

Code Snippet
File Name Tasmota/t1parse.c
Method T1_Get_Private_Dict(T1_Parser parser,

```
....
462.          parser->base_dict    = NULL;
....
460.          parser->private_dict = parser->base_dict;
```

Use of Zero Initialized Pointer\Path 7:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=159
Status New

The variable declared in avar_segment at Tasmota/ttgxvar.c in line 333 is not initialized when it is used by avar_segment at Tasmota/ttgxvar.c in line 333.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	398	395
Object	avar_segment	avar_segment

Code Snippet
File Name Tasmota/ttgxvar.c
Method ft_var_load_avar(TT_Face face)

```
....
398.          blend->avar_segment = NULL;
....
395.            FT_FREE( blend->avar_segment[j].correspondence );
```

Use of Zero Initialized Pointer\Path 8:

Severity Medium
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=160
Status New

The variable declared in avar_segment at Tasmota/ttgxvar.c in line 333 is not initialized when it is used by avar_segment at Tasmota/ttgxvar.c in line 333.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	398	395
Object	avar_segment	avar_segment

Code Snippet
File Name Tasmota/ttgxvar.c
Method ft_var_load_avar(TT_Face face)

```
....
398.          blend->avar_segment = NULL;
....
395.            FT_FREE( blend->avar_segment[j].correspondence );
```

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=1050052&projectid=50042&pathid=149
Status	New

	Source	Destination
File	Tasmota/common.cpp	Tasmota/common.cpp
Line	62	62
Object	ret	ret

Code Snippet
File Name Tasmota/common.cpp
Method TfLiteIntArray* TfLiteIntArrayCreate(int size) {

```
....
62.    TfLiteIntArray* ret = (TfLiteIntArray*)malloc(alloc_size);
```

Memory Leak\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=150
Status	New

	Source	Destination
File	Tasmota/common.cpp	Tasmota/common.cpp
Line	95	95
Object	ret	ret

Code Snippet
File Name Tasmota/common.cpp
Method TfLiteFloatArray* TfLiteFloatArrayCreate(int size) {

```
....
95.    TfLiteFloatArray* ret =
```

Memory Leak\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=151
Status	New

	Source	Destination
File	Tasmota/common.cpp	Tasmota/common.cpp

Line	226	226
Object	data	data

Code Snippet
File Name Tasmota/common.cpp
Method void TfLiteTensorResizeMaybeCopy(size_t num_bytes, TfLiteTensor* tensor,

```
....
226.      tensor->data.data = (char*)malloc(num_bytes);
```

Memory Leak\Path 4:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=152
Status	New

	Source	Destination
File	Tasmota/common.cpp	Tasmota/common.cpp
Line	240	240
Object	data	data

Code Snippet
File Name Tasmota/common.cpp
Method void TfLiteTensorResizeMaybeCopy(size_t num_bytes, TfLiteTensor* tensor,

```
....
240.        tensor->data.data = (char*)malloc(num_bytes);
```

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=1050052&projectid=50042&pathid=115
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 273 of Tasmota/opus_encoder.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	Tasmota/opus_encoder.c	Tasmota/opus_encoder.c
Line	293	293
Object	AssignExpr	AssignExpr

Code Snippet
File Name Tasmota/opus_encoder.c
Method static unsigned char gen_toc(int mode, int framerate, int bandwidth, int channels)

```
....
293.          toc |= tmp << 5;
```

Char Overflow\Path 2:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=116
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 273 of Tasmota/opus_encoder.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination
File	Tasmota/opus_encoder.c	Tasmota/opus_encoder.c
Line	294	294
Object	AssignExpr	AssignExpr

Code Snippet
File Name Tasmota/opus_encoder.c
Method static unsigned char gen_toc(int mode, int framerate, int bandwidth, int channels)

```
....
294.          toc |= period<<3;
```

Char Overflow\Path 3:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=117
Status	New

A variable of a larger data type, AssignExpr, is being assigned to a smaller data type, in 532 of Tasmota/sbrfreq.c. This will cause a loss of data, often the significant bits of a numerical value or the sign bit.

	Source	Destination

File	Tasmota/sbrfreq.c	Tasmota/sbrfreq.c
Line	547	547
Object	AssignExpr	AssignExpr

Code Snippet
File Name Tasmota/sbrfreq.c
Method static int CalcFreqLimiter(unsigned char *freqLimiter, unsigned char *patchNumSubbands, unsigned char *freqLow,

```
....
547.        patchBorders[0] = kStart;
```

Buffer Overflow AddressOfLocalVarReturned

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

Categories

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

Description
Buffer Overflow AddressOfLocalVarReturned\Path 1:

Severity	Medium
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=55
Status	New

The pointer buf at Tasmota/ESPTimeHelper.cpp in line 202 is being used after it has been freed.

	Source	Destination
File	Tasmota/ESPTimeHelper.cpp	Tasmota/ESPTimeHelper.cpp
Line	207	207
Object	buf	buf

Code Snippet
File Name Tasmota/ESPTimeHelper.cpp
Method char *ESPTimeHelper::intStr(int value)

```
....
207.      return buf;
```

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=1050052&projectid=50042&pathid=75
Status	New

Calling free() (line 116) on a variable that was not dynamically allocated (line 116) in file Tasmota/common.cpp may result with a crash.

	Source	Destination
File	Tasmota/common.cpp	Tasmota/common.cpp
Line	128	128
Object	q_params	q_params

Code Snippet
File Name Tasmota/common.cpp
Method void TfLiteQuantizationFree(TfLiteQuantization* quantization) {

```
....
128.      free(q_params);
```

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=1050052&projectid=50042&pathid=76
Status	New

The function alloc_size in Tasmota/common.cpp at line 59 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	Tasmota/common.cpp	Tasmota/common.cpp
Line	62	62
Object	alloc_size	alloc_size

Code Snippet
File Name Tasmota/common.cpp
Method TfLiteIntArray* TfLiteIntArrayCreate(int size) {

```
....
62.    TfLiteIntArray* ret = (TfLiteIntArray*)malloc(alloc_size);
```

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=1050052&projectid=50042&pathid=161
Status	New

	Source	Destination
File	Tasmota/bme68x.c	Tasmota/bme68x.c
Line	253	253
Object	reg_data	reg_data

Code Snippet
File Name Tasmota/bme68x.c
Method int8_t bme68x_get_regs(uint8_t reg_addr, uint8_t *reg_data, uint32_t len, struct bme68x_dev *dev)

```
....
253.          dev->intf_rslt = dev->read(reg_addr, reg_data, len, dev->intf_ptr);
```

Improper Resource Access Authorization\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=162
Status	New

	Source	Destination
File	Tasmota/bme68x.c	Tasmota/bme68x.c
Line	1380	1380
Object	Address	Address

Code Snippet
File Name Tasmota/bme68x.c
Method static int8_t set_mem_page(uint8_t reg_addr, struct bme68x_dev *dev)

```
....
1380.              dev->intf_rslt = dev->read(BME68X_REG_MEM_PAGE |
BME68X_SPI_RD_MSK, &reg, 1, dev->intf_ptr);
```

Improper Resource Access Authorization\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=163
Status	New

	Source	Destination
File	Tasmota/bme68x.c	Tasmota/bme68x.c
Line	1412	1412
Object	Address	Address

Code Snippet
File Name Tasmota/bme68x.c
Method static int8_t get_mem_page(struct bme68x_dev *dev)

```
....
1412.          dev->intf_rslt = dev->read(BME68X_REG_MEM_PAGE |
BME68X_SPI_RD_MSK, &reg, 1, dev->intf_ptr);
```

Improper Resource Access Authorization\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=164
Status	New

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	152	152
Object	buffer	buffer

Code Snippet
File Name Tasmota/opusfile.c
Method static int op_get_data(OggOpusFile *_of,int _nbytes){

```
....
152.    nbytes=(int)(*_of->callbacks.read)(_of->stream,buffer,_nbytes);
```

Improper Resource Access Authorization\Path 5:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=165
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	870	870
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
870.    fprintf(stderr,"\nSmall preclipped packing (LSb): ");
```

Improper Resource Access Authorization\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=166
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	872	872
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
872.    fprintf(stderr,"ok.");
```

Improper Resource Access Authorization\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=167
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c

Line	874	874
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
874.    fprintf(stderr,"\nNull bit call (LSb): ");
```

Improper Resource Access Authorization\Path 8:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=168
Status New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	876	876
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
876.    fprintf(stderr,"ok.");
```

Improper Resource Access Authorization\Path 9:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=169
Status New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	878	878
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
878.    fprintf(stderr,"\nLarge preclipped packing (LSb): ");
```

Improper Resource Access Authorization\Path 10:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=170
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	880	880
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
880.    fprintf(stderr,"ok.");
```

Improper Resource Access Authorization\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=171
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	882	882
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
882.    fprintf(stderr,"\n32 bit preclipped packing (LSb): ");
```

Improper Resource Access Authorization\Path 12:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=172
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	892	892
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
892.        fprintf(stderr,"%ld != %lu
(%lx!=%lx):",oggpack_look(&r,32),large[i],
```

Improper Resource Access Authorization\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=173
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	899	899
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
899.    fprintf(stderr,"ok.");
```

Improper Resource Access Authorization\Path 14:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=174
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c

Line	901	901
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
901.    fprintf(stderr,"\nSmall unclipped packing (LSb): ");
```

Improper Resource Access Authorization\Path 15:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=175
Status New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	903	903
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
903.    fprintf(stderr,"ok.");
```

Improper Resource Access Authorization\Path 16:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=176
Status New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	905	905
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
905.    fprintf(stderr,"\nLarge unclipped packing (LSb): ");
```

Improper Resource Access Authorization\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=177
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	907	907
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
907.    fprintf(stderr,"ok.");
```

Improper Resource Access Authorization\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=178
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	909	909
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
909.    fprintf(stderr,"\nSingle bit unclipped packing (LSb): ");
```

Improper Resource Access Authorization\Path 19:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=179
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	911	911
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
911.    fprintf(stderr,"ok.");
```

Improper Resource Access Authorization\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=180
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	913	913
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
913.    fprintf(stderr,"\nTesting read past end (LSb): ");
```

Improper Resource Access Authorization\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=181
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	917	917

Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
917.        fprintf(stderr,"failed; got -1 prematurely.\n");
```

Improper Resource Access Authorization\Path 22:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=182
Status New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	923	923
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
923.        fprintf(stderr,"failed; read past end without -1.\n");
```

Improper Resource Access Authorization\Path 23:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=183
Status New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	928	928
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
928.        fprintf(stderr,"failed 2; got -1 prematurely.\n");
```

Improper Resource Access Authorization\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=184
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	934	934
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
934.      fprintf(stderr,"failed 3; got -1 prematurely.\n");
```

Improper Resource Access Authorization\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=185
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	939	939
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
939.      fprintf(stderr,"failed; read past end without -1.\n");
```

Improper Resource Access Authorization\Path 26:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=186
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	944	944
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
944.      fprintf(stderr,"failed; read past end without -1.\n");
```

Improper Resource Access Authorization\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=187
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	948	948
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
948.    fprintf(stderr,"ok.");
```

Improper Resource Access Authorization\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=188
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	952	952

Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
952.    fprintf(stderr,"\nTesting aligned writecopies (LSb): ");
```

Improper Resource Access Authorization\Path 29:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=189
Status New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	959	959
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
959.    fprintf(stderr,"ok.      ");
```

Improper Resource Access Authorization\Path 30:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=190
Status New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	961	961
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
961.    fprintf(stderr,"\nTesting unaligned writecopies (LSb): ");
```

Improper Resource Access Authorization\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=191
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	971	971
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
971.    fprintf(stderr,"ok.      \n");
```

Improper Resource Access Authorization\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=192
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	980	980
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
980.    fprintf(stderr,"\nSmall preclipped packing (MSb): ");
```

Improper Resource Access Authorization\Path 33:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=193
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	982	982
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
982.    fprintf(stderr,"ok.");
```

Improper Resource Access Authorization\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=194
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	984	984
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
984.    fprintf(stderr,"\nNull bit call (MSb): ");
```

Improper Resource Access Authorization\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=195
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	986	986

Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
986.    fprintf(stderr,"ok.");
```

Improper Resource Access Authorization\Path 36:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=196
Status New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	988	988
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
988.    fprintf(stderr,"\nLarge preclipped packing (MSb): ");
```

Improper Resource Access Authorization\Path 37:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=197
Status New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	990	990
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
990.    fprintf(stderr,"ok.");
```

Improper Resource Access Authorization\Path 38:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=198
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	992	992
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
992.    fprintf(stderr,"\n32 bit preclipped packing (MSb): ");
```

Improper Resource Access Authorization\Path 39:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=199
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	1002	1002
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
1002.        fprintf(stderr,"%ld != %lu
(%lx!=%lx):",oggpackB_look(&r,32),large[i],
```

Improper Resource Access Authorization\Path 40:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=200
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	1009	1009
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
1009.    fprintf(stderr,"ok.");
```

Improper Resource Access Authorization\Path 41:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=201
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	1011	1011
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
1011.    fprintf(stderr,"\nSmall unclipped packing (MSb): ");
```

Improper Resource Access Authorization\Path 42:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=202
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c

Line	1013	1013
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
1013.    fprintf(stderr,"ok.");
```

Improper Resource Access Authorization\Path 43:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=203
Status New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	1015	1015
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
1015.    fprintf(stderr,"\nLarge unclipped packing (MSb): ");
```

Improper Resource Access Authorization\Path 44:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=204
Status New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	1017	1017
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
1017.    fprintf(stderr,"ok.");
```

Improper Resource Access Authorization\Path 45:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=205
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	1019	1019
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
1019.    fprintf(stderr,"\nSingle bit unclipped packing (MSb): ");
```

Improper Resource Access Authorization\Path 46:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=206
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	1021	1021
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
1021.    fprintf(stderr,"ok.");
```

Improper Resource Access Authorization\Path 47:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=207
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	1023	1023
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
1023.    fprintf(stderr,"\nTesting read past end (MSb): ");
```

Improper Resource Access Authorization\Path 48:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=208
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	1027	1027
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
1027.        fprintf(stderr,"failed; got -1 prematurely.\n");
```

Improper Resource Access Authorization\Path 49:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=209
Status	New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	1033	1033

Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
1033.        fprintf(stderr,"failed; read past end without -1.\n");
```

Improper Resource Access Authorization\Path 50:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=210
Status New

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	1038	1038
Object	fprintf	fprintf

Code Snippet
File Name Tasmota/bitwise.c
Method int main(void){

```
....
1038.        fprintf(stderr,"failed 2; got -1 prematurely.\n");
```

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=1050052&projectid=50042&pathid=77
Status New

The variable declared in null at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by doblend at Tasmota/ttgxvar.c in line 2777.

	Source	Destination

File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2811
Object	null	doblend

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2045.       FT_MM_Var*            mmvar = NULL;
```

File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Blend(TT_Face face,

```
....
2811.       if ( face->doblend )
```

NULL Pointer Dereference\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=78
Status	New

The variable declared in null at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by face at Tasmota/ttgxvar.c in line 1185.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	1204
Object	null	face

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2045.       FT_MM_Var*            mmvar = NULL;
```

File Name Tasmota/ttgxvar.c
Method ft_var_load_mvar(TT_Face face)

```
....
1204.       error = face->goto_table( face, TTAG_MVAR, stream, &table_len
);
```

NULL Pointer Dereference\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=79
Status	New

The variable declared in null at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by is_cff2 at Tasmota/ttgxvar.c in line 2505.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2565
Object	null	is_cff2

Code Snippet

File Name Tasmota/ttgxvar.c

Method TT_Get_MM_Var(TT_Face face,

```
....
2045.       FT_MM_Var*           mmvar = NULL;
```

File Name Tasmota/ttgxvar.c

Method tt_set_mm_blend(TT_Face face,

```
....
2565.       if ( !face->is_cff2 && !blend->glyphoffsets )
```

NULL Pointer Dereference\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=80
Status	New

The variable declared in null at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by face at Tasmota/ttgxvar.c in line 333.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	351

Object	null	face

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2045.       FT_MM_Var*            mmvar = NULL;
```

File Name Tasmota/ttgxvar.c

Method ft_var_load_avar(TT_Face face)

```
....
351.      error = face->goto_table( face, TTAG_avar, stream, &table_len
);
```

NULL Pointer Dereference\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=81
Status	New

The variable declared in null at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by blend at Tasmota/ttgxvar.c in line 2505.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2530
Object	null	blend

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2045.       FT_MM_Var*            mmvar = NULL;
```

File Name Tasmota/ttgxvar.c

Method tt_set_mm_blend(TT_Face face,

```
....
2530.       if ( !face->blend )
```

NULL Pointer Dereference\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=82
Status	New

The variable declared in null at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by blend at Tasmota/ttgxvar.c in line 2858.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2952
Object	null	blend

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2045.      FT_MM_Var*           mmvar = NULL;
```

File Name Tasmota/ttgxvar.c
Method TT_Set_Var_Design(TT_Face face,

```
....
2952.      if ( !face->blend->avar_loaded )
```

NULL Pointer Dereference\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=83
Status	New

The variable declared in null at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by doblend at Tasmota/ttgxvar.c in line 3000.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	3034
Object	null	doblend

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2045.       FT_MM_Var*            mmvar = NULL;
```

File Name Tasmota/ttgxvar.c

Method TT_Get_Var_Design(TT_Face face,

```
....
3034.       if ( face->doblend )
```

NULL Pointer Dereference\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=84
Status	New

The variable declared in null at Tasmota/ttgxvar.c in line 3752 is not initialized when it is used by x at Tasmota/ttgxvar.c in line 3526.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	3763	3544
Object	null	x

Code Snippet

File Name Tasmota/ttgxvar.c

Method TT_Vary_Apply_Glyph_Deltas(TT_Face face,

```
....
3763.       FT_Vector*  points_out = NULL;   /* coordinates in 16.16
format */
```

File Name Tasmota/ttgxvar.c

Method tt_delta_shift(int p1,

```
....
3544.         out_points[p].x += delta.x;
```

NULL Pointer Dereference\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=85
Status	New

The variable declared in null at Tasmota/ttgxvar.c in line 3752 is not initialized when it is used by x at Tasmota/ttgxvar.c in line 3526.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	3763	3550
Object	null	x

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Vary_Apply_Glyph_Deltas(TT_Face face,

```
....
3763.      FT_Vector*  points_out = NULL;  /* coordinates in 16.16
format */
```

File Name Tasmota/ttgxvar.c
Method tt_delta_shift(int p1,

```
....
3550.        out_points[p].x += delta.x;
```

NULL Pointer Dereference\Path 10:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=86
Status New

The variable declared in null at Tasmota/ttgxvar.c in line 3752 is not initialized when it is used by y at Tasmota/ttgxvar.c in line 3526.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	3763	3551
Object	null	y

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Vary_Apply_Glyph_Deltas(TT_Face face,

```
....
3763.      FT_Vector*  points_out = NULL;  /* coordinates in 16.16
format */
```

File Name Tasmota/ttgxvar.c

Method tt_delta_shift(int p1,

```
....
3551.        out_points[p].y += delta.y;
```

NULL Pointer Dereference\Path 11:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=87
Status	New

The variable declared in null at Tasmota/ttgxvar.c in line 3752 is not initialized when it is used by y at Tasmota/ttgxvar.c in line 3526.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	3763	3545
Object	null	y

Code Snippet

File Name Tasmota/ttgxvar.c

Method TT_Vary_Apply_Glyph_Deltas(TT_Face face,

```
....
3763.      FT_Vector*  points_out = NULL;  /* coordinates in 16.16
format */
```

File Name Tasmota/ttgxvar.c

Method tt_delta_shift(int p1,

```
....
3545.        out_points[p].y += delta.y;
```

NULL Pointer Dereference\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=88
Status	New

The variable declared in 0 at Tasmota/bitwise.c in line 179 is not initialized when it is used by Pointer at Tasmota/bitwise.c in line 179.

	Source	Destination
File	Tasmota/bitwise.c	Tasmota/bitwise.c
Line	215	215
Object	0	Pointer

Code Snippet
File Name Tasmota/bitwise.c
Method static void oggpack_writecopy_helper(oggpack_buffer *b,

```
....
215.       *b->ptr=0;
```

NULL Pointer Dereference\Path 13:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=89
Status New

The variable declared in 0 at Tasmota/opusfile.c in line 631 is not initialized when it is used by op at Tasmota/opusfile.c in line 830.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	663	963
Object	0	op

Code Snippet
File Name Tasmota/opusfile.c
Method static int op_granpos_add(ogg_int64_t *_dst_gp,ogg_int64_t _src_gp,

```
....
663.    return 0;
```

File Name Tasmota/opusfile.c
Method static int op_find_initial_pcm_offset(OggOpusFile *_of,

```
....
963.       prev_packet_gp=_of->op[pi].granulepos;
```

NULL Pointer Dereference\Path 14:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=90

Status	New

The variable declared in 0 at Tasmota/opusfile.c in line 631 is not initialized when it is used by op at Tasmota/opusfile.c in line 830.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	663	961
Object	0	op

Code Snippet
File Name Tasmota/opusfile.c
Method static int op_granpos_add(ogg_int64_t *_dst_gp,ogg_int64_t _src_gp,

```
....
663.    return 0;
```

File Name Tasmota/opusfile.c
Method static int op_find_initial_pcm_offset(OggOpusFile *_of,

```
....
961.      OP_ALWAYS_TRUE(!op_granpos_add(&_of->op[pi].granulepos,
```

NULL Pointer Dereference\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=91
Status	New

The variable declared in 0 at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by num_designs at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2205	2204
Object	0	num_designs

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2205.          ~0U;                    /* meaningless in this context;
each glyph */
....
2204.        mmvar->num_designs =
```

NULL Pointer Dereference\Path 16:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=92
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by namedstyle at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2216
Object	mmvar	namedstyle

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2045.      FT_MM_Var*           mmvar = NULL;
....
2216.        mmvar->namedstyle =
```

NULL Pointer Dereference\Path 17:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=93
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by coords at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2223
Object	mmvar	coords

Code Snippet
File Name Tasmota/ttgxvar.c

Method TT_Get_MM_Var(TT_Face face,

```
....
2045.       FT_MM_Var*            mmvar = NULL;
....
2223.           mmvar->namedstyle[i].coords   = next_coords;
```

NULL Pointer Dereference\Path 18:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=94
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by num_axis at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2202
Object	mmvar	num_axis

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2045.       FT_MM_Var*            mmvar = NULL;
....
2202.         mmvar->num_axis =
```

NULL Pointer Dereference\Path 19:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=95
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by num_namedstyles at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2208
Object	mmvar	num_namedstyles

Code Snippet
File Name Tasmota/ttgxvar.c

Method TT_Get_MM_Var(TT_Face face,

```
....
2045.      FT_MM_Var*           mmvar = NULL;
....
2208.        mmvar->num_namedstyles =
```

NULL Pointer Dereference\Path 20:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=96
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by axis at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2214
Object	mmvar	axis

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2045.      FT_MM_Var*           mmvar = NULL;
....
2214.        mmvar->axis =
```

NULL Pointer Dereference\Path 21:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=97
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by namedstyle at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2464
Object	mmvar	namedstyle

Code Snippet
File Name Tasmota/ttgxvar.c

Method TT_Get_MM_Var(TT_Face face,

```
....
2045.       FT_MM_Var*            mmvar = NULL;
....
2464.         mmvar->namedstyle =
```

NULL Pointer Dereference\Path 22:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=98
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by coords at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2471
Object	mmvar	coords

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2045.       FT_MM_Var*            mmvar = NULL;
....
2471.           mmvar->namedstyle[n].coords  = next_coords;
```

NULL Pointer Dereference\Path 23:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=99
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by name at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2231
Object	mmvar	name

Code Snippet
File Name Tasmota/ttgxvar.c

Method	TT_Get_MM_Var(TT_Face face,

```
....
2045.      FT_MM_Var*           mmvar = NULL;
....
2231.          mmvar->axis[i].name  = next_name;
```

NULL Pointer Dereference\Path 24:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=100
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by num_designs at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2204
Object	mmvar	num_designs

Code Snippet	
File Name	Tasmota/ttgxvar.c
Method	TT_Get_MM_Var(TT_Face face,

```
....
2045.      FT_MM_Var*           mmvar = NULL;
....
2204.        mmvar->num_designs =
```

NULL Pointer Dereference\Path 25:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=101
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by axis at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2462
Object	mmvar	axis

Code Snippet	
File Name	Tasmota/ttgxvar.c

Method TT_Get_MM_Var(TT_Face face,

```
....
2045.       FT_MM_Var*            mmvar = NULL;
....
2462.         mmvar->axis =
```

NULL Pointer Dereference\Path 26:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=102
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by axis at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2438
Object	mmvar	axis

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2045.       FT_MM_Var*            mmvar = NULL;
....
2438.               a = mmvar->axis;
```

NULL Pointer Dereference\Path 27:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=103
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by axis at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2465
Object	mmvar	axis

Code Snippet
File Name Tasmota/ttgxvar.c

Method TT_Get_MM_Var(TT_Face face,

```
....
2045.      FT_MM_Var*           mmvar = NULL;
....
2465.          (FT_Var_Named_Style*)( (char*)mmvar->axis+ axis_size );
```

NULL Pointer Dereference\Path 28:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=104
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by axis at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2475
Object	mmvar	axis

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2045.      FT_MM_Var*           mmvar = NULL;
....
2475.        a          = mmvar->axis;
```

NULL Pointer Dereference\Path 29:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=105
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by axis at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2217
Object	mmvar	axis

Code Snippet
File Name Tasmota/ttgxvar.c

Method TT_Get_MM_Var(TT_Face face,

```
....
2045.       FT_MM_Var*            mmvar = NULL;
....
2217.           (FT_Var_Named_Style*)( (char*)mmvar->axis + axis_size );
```

NULL Pointer Dereference\Path 30:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=106
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by axis at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2240
Object	mmvar	axis

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2045.       FT_MM_Var*            mmvar = NULL;
....
2240.         a = mmvar->axis;
```

NULL Pointer Dereference\Path 31:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=107
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by namedstyle at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2433
Object	mmvar	namedstyle

Code Snippet
File Name Tasmota/ttgxvar.c

Method TT_Get_MM_Var(TT_Face face,

```
....
2045.       FT_MM_Var*            mmvar = NULL;
....
2433.               ns = &mmvar->namedstyle[fvar_head.instanceCount];
```

NULL Pointer Dereference\Path 32:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=108
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by namedstyle at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2227
Object	mmvar	namedstyle

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2045.       FT_MM_Var*            mmvar = NULL;
....
2227.         next_name = (FT_String*)( (char*)mmvar->namedstyle +
```

NULL Pointer Dereference\Path 33:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=109
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by namedstyle at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2467
Object	mmvar	namedstyle

Code Snippet
File Name Tasmota/ttgxvar.c

Method TT_Get_MM_Var(TT_Face face,

```
....
2045.      FT_MM_Var*           mmvar = NULL;
....
2467.        next_coords = (FT_Fixed*)( (char*)mmvar->namedstyle +
```

NULL Pointer Dereference\Path 34:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=110
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by namedstyle at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2219
Object	mmvar	namedstyle

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2045.      FT_MM_Var*           mmvar = NULL;
....
2219.        next_coords = (FT_Fixed*)( (char*)mmvar->namedstyle +
```

NULL Pointer Dereference\Path 35:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=111
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by namedstyle at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2322
Object	mmvar	namedstyle

Code Snippet
File Name Tasmota/ttgxvar.c

Method TT_Get_MM_Var(TT_Face face,

```
....
2045.       FT_MM_Var*            mmvar = NULL;
....
2322.         ns   = mmvar->namedstyle;
```

NULL Pointer Dereference\Path 36:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=112
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by namedstyle at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2476
Object	mmvar	namedstyle

Code Snippet
File Name Tasmota/ttgxvar.c
Method TT_Get_MM_Var(TT_Face face,

```
....
2045.       FT_MM_Var*            mmvar = NULL;
....
2476.         next_name = (FT_String*)( (char*)mmvar->namedstyle +
```

NULL Pointer Dereference\Path 37:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=113
Status	New

The variable declared in mmvar at Tasmota/ttgxvar.c in line 2036 is not initialized when it is used by num_namedstyles at Tasmota/ttgxvar.c in line 2036.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	2045	2469
Object	mmvar	num_namedstyles

Code Snippet
File Name Tasmota/ttgxvar.c

Method TT_Get_MM_Var(TT_Face face,

```
....
2045.      FT_MM_Var*           mmvar = NULL;
....
2469.        for ( n = 0; n < mmvar->num_namedstyles; n++ )
```

Unchecked Array Index

Query Path:
CPP\Cx\CPP Low Visibility\Unchecked Array Index Version:1

Categories

NIST SP 800-53: SI-10 Information Input Validation (P1)

Description

Unchecked Array Index\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=124
Status	New

	Source	Destination
File	Tasmota/bme68x.c	Tasmota/bme68x.c
Line	1707	1707
Object	index1	index1

Code Snippet
File Name Tasmota/bme68x.c
Method static void swap_fields(uint8_t index1, uint8_t index2, struct bme68x_data *field[])

```
....
1707.      field[index1] = field[index2];
```

Unchecked Array Index\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=125
Status	New

	Source	Destination
File	Tasmota/bme68x.c	Tasmota/bme68x.c
Line	1708	1708
Object	index2	index2

Code Snippet	
File Name	Tasmota/bme68x.c
Method	static void swap_fields(uint8_t index1, uint8_t index2, struct bme68x_data *field[])

```
....
1708.       field[index2] = temp;
```

Unchecked Array Index\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=126
Status	New

	Source	Destination
File	Tasmota/scalfact.c	Tasmota/scalfact.c
Line	384	384
Object	ch	ch

Code Snippet	
File Name	Tasmota/scalfact.c
Method	int UnpackScaleFactors(MP3DecInfo *mp3DecInfo, unsigned char *buf, int *bitOffset, int bitsAvail, int gr, int ch)

```
....
384.         mp3DecInfo->part23Length[gr][ch] = si-
>sis[gr][ch].part23Length;
```

Unchecked Array Index\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=127
Status	New

	Source	Destination
File	Tasmota/t1parse.c	Tasmota/t1parse.c
Line	490	490
Object	len	len

Code Snippet	
File Name	Tasmota/t1parse.c
Method	T1_Get_Private_Dict(T1_Parser parser,

```
....
490.          parser->private_dict[len] = '\0';
```

Unchecked Array Index\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=128
Status	New

	Source	Destination
File	Tasmota/ttinterp.c	Tasmota/ttinterp.c
Line	1580	1580
Object	idx	idx

Code Snippet
File Name Tasmota/ttinterp.c
Method Write_CVT(TT_ExecContext exc,

```
....
1580.       exc->cvt[idx] = value;
```

Unchecked Array Index\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=129
Status	New

	Source	Destination
File	Tasmota/ttinterp.c	Tasmota/ttinterp.c
Line	1589	1589
Object	idx	idx

Code Snippet
File Name Tasmota/ttinterp.c
Method Write_CVT_Stretched(TT_ExecContext exc,

```
....
1589.       exc->cvt[idx] = FT_DivFix( value, Current_Ratio( exc ) );
```

Unchecked Array Index\Path 7:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=130
Status	New

	Source	Destination
File	Tasmota/ttinterp.c	Tasmota/ttinterp.c
Line	1598	1598
Object	idx	idx

Code Snippet
File Name Tasmota/ttinterp.c
Method Move_CVT(TT_ExecContext exc,

```
....
1598.       exc->cvt[idx] = ADD_LONG( exc->cvt[idx], value );
```

Unchecked Array Index\Path 8:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=131
Status	New

	Source	Destination
File	Tasmota/ttinterp.c	Tasmota/ttinterp.c
Line	1607	1607
Object	idx	idx

Code Snippet
File Name Tasmota/ttinterp.c
Method Move_CVT_Stretched(TT_ExecContext exc,

```
....
1607.       exc->cvt[idx] = ADD_LONG( exc->cvt[idx],
```

Unchecked Array Index\Path 9:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=132
Status	New

	Source	Destination
File	Tasmota/ttinterp.c	Tasmota/ttinterp.c
Line	1788	1788

Object	point	point

Code Snippet
File Name Tasmota/ttinterp.c
Method Direct_Move(TT_ExecContext exc,

```
....
1788.         zone->tags[point] |= FT_CURVE_TAG_TOUCH_X;
```

Unchecked Array Index\Path 10:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=133
Status New

	Source	Destination
File	Tasmota/ttinterp.c	Tasmota/ttinterp.c
Line	1806	1806
Object	point	point

Code Snippet
File Name Tasmota/ttinterp.c
Method Direct_Move(TT_ExecContext exc,

```
....
1806.         zone->tags[point] |= FT_CURVE_TAG_TOUCH_Y;
```

Unchecked Array Index\Path 11:

Severity Low
Result State To Verify
Online Results http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=134
Status New

	Source	Destination
File	Tasmota/ttinterp.c	Tasmota/ttinterp.c
Line	1890	1890
Object	point	point

Code Snippet
File Name Tasmota/ttinterp.c
Method Direct_Move_X(TT_ExecContext exc,

```
....
1890.       zone->tags[point]   |= FT_CURVE_TAG_TOUCH_X;
```

Unchecked Array Index\Path 12:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=135
Status	New

	Source	Destination
File	Tasmota/ttinterp.c	Tasmota/ttinterp.c
Line	1909	1909
Object	point	point

Code Snippet
File Name Tasmota/ttinterp.c
Method Direct_Move_Y(TT_ExecContext exc,

```
....
1909.       zone->tags[point] |= FT_CURVE_TAG_TOUCH_Y;
```

Unchecked Array Index\Path 13:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=136
Status	New

	Source	Destination
File	Tasmota/ttinterp.c	Tasmota/ttinterp.c
Line	5533	5533
Object	point	point

Code Snippet
File Name Tasmota/ttinterp.c
Method Move_Zp2_Point(TT_ExecContext exc,

```
....
5533.           exc->zp2.tags[point] |= FT_CURVE_TAG_TOUCH_X;
```

Unchecked Array Index\Path 14:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=137
Status	New

	Source	Destination
File	Tasmota/ttinterp.c	Tasmota/ttinterp.c
Line	5547	5547
Object	point	point

Code Snippet
File Name Tasmota/ttinterp.c
Method Move_Zp2_Point(TT_ExecContext exc,

```
....
5547.          exc->zp2.tags[point] |= FT_CURVE_TAG_TOUCH_Y;
```

Unchecked Array Index\Path 15:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=138
Status	New

	Source	Destination
File	Tasmota/scalfact.c	Tasmota/scalfact.c
Line	384	384
Object	gr	gr

Code Snippet
File Name Tasmota/scalfact.c
Method int UnpackScaleFactors(MP3DecInfo *mp3DecInfo, unsigned char *buf, int *bitOffset, int bitsAvail, int gr, int ch)

```
....
384.        mp3DecInfo->part23Length[gr][ch] = si-
>sis[gr][ch].part23Length;
```

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=1050052&projectid=50042&pathid=17
Status	New

The TfLiteFloatArrayCreate method calls the ret function, at line 94 of Tasmota/common.cpp. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

	Source	Destination
File	Tasmota/common.cpp	Tasmota/common.cpp
Line	95	95
Object	ret	ret

Code Snippet
File Name Tasmota/common.cpp
Method TfLiteFloatArray* TfLiteFloatArrayCreate(int size) {

```
....
95.    TfLiteFloatArray* ret =
```

Unchecked Return Value\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=18
Status	New

The silk_NLSF2A method calls the cos_LSF_QA function, at line 66 of Tasmota/NLSF2A.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	Tasmota/NLSF2A.c	Tasmota/NLSF2A.c
Line	83	83
Object	cos_LSF_QA	cos_LSF_QA

Code Snippet
File Name Tasmota/NLSF2A.c
Method void silk_NLSF2A(

```
....
83.      opus_int32 *cos_LSF_QA = (opus_int32*)malloc(sizeof(opus_int32)
* SILK_MAX_ORDER_LPC );
```

Unchecked Return Value\Path 3:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=19
Status	New

The silk_NLSF2A method calls the P function, at line 66 of Tasmota/NLSF2A.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	Tasmota/NLSF2A.c	Tasmota/NLSF2A.c
Line	84	84
Object	P	P

Code Snippet
File Name Tasmota/NLSF2A.c
Method void silk_NLSF2A(

```
....
84.      opus_int32 *P = (opus_int32*)malloc(sizeof(opus_int32) *
(SILK_MAX_ORDER_LPC / 2 + 1));
```

Unchecked Return Value\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=20
Status	New

The silk_NLSF2A method calls the Q function, at line 66 of Tasmota/NLSF2A.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	Tasmota/NLSF2A.c	Tasmota/NLSF2A.c
Line	85	85
Object	Q	Q

Code Snippet
File Name Tasmota/NLSF2A.c
Method void silk_NLSF2A(

```
....
85.      opus_int32 *Q= (opus_int32*)malloc(sizeof(opus_int32) *
(SILK_MAX_ORDER_LPC / 2 + 1));
```

Unchecked Return Value\Path 5:

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

	BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=21
Status	New

The silk_NLSF2A method calls the a32_QA1 function, at line 66 of Tasmota/NLSF2A.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	Tasmota/NLSF2A.c	Tasmota/NLSF2A.c
Line	87	87
Object	a32_QA1	a32_QA1

Code Snippet
File Name Tasmota/NLSF2A.c
Method void silk_NLSF2A(

```
....
87.      opus_int32 *a32_QA1 = (opus_int32*)malloc(sizeof(opus_int32) *
SILK_MAX_ORDER_LPC );
```

Unchecked Return Value\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=22
Status	New

The op_open_seekable2_impl method calls the sr function, at line 1402 of Tasmota/opusfile.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	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	1406	1406
Object	sr	sr

Code Snippet
File Name Tasmota/opusfile.c
Method static int op_open_seekable2_impl(OggOpusFile *_of){

```
....
1406.    OpusSeekRecord *sr = (OpusSeekRecord*)malloc(64 *
sizeof(OpusSeekRecord));
```

Unchecked Return Value\Path 7:

Severity	Low
Result State	To Verify

Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=23
Status	New

The op_open_seekable2 method calls the os_start function, at line 1431 of Tasmota/opusfile.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	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	1433	1433
Object	os_start	os_start

Code Snippet
File Name Tasmota/opusfile.c
Method static int op_open_seekable2(OggOpusFile *_of){

```
....
1433.    ogg_stream_state  *os_start =
(ogg_stream_state*)malloc(sizeof(ogg_stream_state));
```

Potential Off by One Error in Loops

Query Path:
CPP\Cx\CPP Heuristic\Potential Off by One Error in Loops Version:1

Categories

PCI DSS v3.2: PCI DSS (3.2) - 6.5.1 - Injection flaws - particularly SQL injection
NIST SP 800-53: SI-16 Memory Protection (P1)
OWASP Top 10 2017: A1-Injection

Description
Potential Off by One Error in Loops\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=48
Status	New

The buffer allocated by <= in Tasmota/sbrfreq.c at line 317 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	Tasmota/sbrfreq.c	Tasmota/sbrfreq.c
Line	323	323
Object	<=	<=

Code Snippet
File Name Tasmota/sbrfreq.c

Method	static int CalcFreqHigh(unsigned char *freqHigh, unsigned char *freqMaster, int nMaster, int crossOverBand)

```
....
323.        for (k = 0; k <= nHigh; k++)
```

Potential Off by One Error in Loops\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=49
Status	New

The buffer allocated by <= in Tasmota/sbrfreq.c at line 532 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	Tasmota/sbrfreq.c	Tasmota/sbrfreq.c
Line	554	554
Object	<=	<=

Code Snippet	
File Name	Tasmota/sbrfreq.c
Method	static int CalcFreqLimiter(unsigned char *freqLimiter, unsigned char *patchNumSubbands, unsigned char *freqLow,

```
....
554.        for (k = 0; k <= nLow; k++)
```

Potential Off by One Error in Loops\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=50
Status	New

The buffer allocated by <= in Tasmota/sbrfreq.c at line 532 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	Tasmota/sbrfreq.c	Tasmota/sbrfreq.c
Line	583	583
Object	<=	<=

Code Snippet	
File Name	Tasmota/sbrfreq.c

Method	static int CalcFreqLimiter(unsigned char *freqLimiter, unsigned char *patchNumSubbands, unsigned char *freqLow,

```
....
583.        for (k = 0; k <= nLimiter; k++)
```

Potential Off by One Error in Loops\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=51
Status	New

The buffer allocated by <= in Tasmota/scalfact.c at line 208 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	Tasmota/scalfact.c	Tasmota/scalfact.c
Line	328	328
Object	<=	<=

Code Snippet	
File Name	Tasmota/scalfact.c
Method	static void UnpackSFMPEG2(BitStreamInfo *bsi, SideInfoSub *sis, ScaleFactorInfoSub *sfis, int gr, int ch, int modeExt, ScaleFactorJS *sfjs)

```
....
328.              for (nrIdx = 0; nrIdx <= 3; nrIdx++) {
```

Potential Off by One Error in Loops\Path 5:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=52
Status	New

The buffer allocated by <= in Tasmota/ttgxvar.c at line 1463 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	1570	1570
Object	<=	<=

Code Snippet	
File Name	Tasmota/ttgxvar.c
Method	ft_var_load_gvar(TT_Face face)

```
....
1570.        for ( i = 0; i <= gvar_head.glyphCount; i++ )
```

Potential Off by One Error in Loops\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=53
Status	New

The buffer allocated by <= in Tasmota/ttgxvar.c at line 1463 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	1600	1600
Object	<=	<=

Code Snippet
File Name Tasmota/ttgxvar.c
Method ft_var_load_gvar(TT_Face face)

```
....
1600.        for ( i = 0; i <= gvar_head.glyphCount; i++ )
```

Potential Off by One Error in Loops\Path 7:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=54
Status	New

The buffer allocated by <= in Tasmota/ttgxvar.c at line 3563 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

	Source	Destination
File	Tasmota/ttgxvar.c	Tasmota/ttgxvar.c
Line	3579	3579
Object	<=	<=

Code Snippet
File Name Tasmota/ttgxvar.c
Method tt_delta_interpolate(int p1,

```
....
3579.       for ( i = 0; i <= 1; i++ )
```

Arithmenic Operation On Boolean

Query Path:
CPP\Cx\CPP Low Visibility\Arithmenic Operation On Boolean Version:1

Categories

FISMA 2014: Audit And Accountability
NIST SP 800-53: SC-5 Denial of Service Protection (P1)

Description

Arithmenic Operation On Boolean\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=118
Status	New

	Source	Destination
File	Tasmota/opus_encoder.c	Tasmota/opus_encoder.c
Line	1985	1985
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name Tasmota/opus_encoder.c

Method opus_int32 opus_encode_native(OpusEncoder *st, const opus_val16 *pcm, int frame_size,

```
....
1985.       if ( st->mode != MODE_CELT_ONLY && ec_tell(&enc)+17+20*(st->mode == MODE_HYBRID) <= 8*(max_data_bytes-1))
```

Arithmenic Operation On Boolean\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=119
Status	New

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	735	735
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name	Tasmota/opusfile.c
Method	static int op_granpos_cmp(ogg_int64_t _gp_a,ogg_int64_t _gp_b){

```
....
735.    return (_gp_a>_gp_b)-(_gp_b>_gp_a);
```

Arithmenic Operation On Boolean\Path 3:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=120
Status	New

	Source	Destination
File	Tasmota/ttinterp.c	Tasmota/ttinterp.c
Line	1471	1471
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name	Tasmota/ttinterp.c
Method	TT_DotFix14(FT_Int32 ax,

```
....
1471.       hi1 = ( m >> 16 ) + ( (FT_Int32)l >> 31 ) + ( lo1 < l );
```

Arithmenic Operation On Boolean\Path 4:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=121
Status	New

	Source	Destination
File	Tasmota/ttinterp.c	Tasmota/ttinterp.c
Line	1478	1478
Object	BinaryExpr	BinaryExpr

Code Snippet

File Name	Tasmota/ttinterp.c
Method	TT_DotFix14(FT_Int32 ax,

```
....
1478.       hi2 = ( m >> 16 ) + ( (FT_Int32)l >> 31 ) + ( lo2 < l );
```

Arithmenic Operation On Boolean\Path 5:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=122
Status	New

	Source	Destination
File	Tasmota/ttinterp.c	Tasmota/ttinterp.c
Line	1482	1482
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name Tasmota/ttinterp.c
Method TT_DotFix14(FT_Int32 ax,

```
....
1482.       hi = hi1 + hi2 + ( lo < lo1 );
```

Arithmenic Operation On Boolean\Path 6:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=123
Status	New

	Source	Destination
File	Tasmota/ttinterp.c	Tasmota/ttinterp.c
Line	1487	1487
Object	BinaryExpr	BinaryExpr

Code Snippet
File Name Tasmota/ttinterp.c
Method TT_DotFix14(FT_Int32 ax,

```
....
1487.       hi += s + ( l < lo );
```

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=1050052&projectid=50042&pathid=14
Status	New

Method opus_encode_native at line 1066 of Tasmota/opus_encoder.c uses a weak method rand 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	Tasmota/opus_encoder.c	Tasmota/opus_encoder.c
Line	1317	1317
Object	rand	rand

Code Snippet
File Name Tasmota/opus_encoder.c
Method opus_int32 opus_encode_native(OpusEncoder *st, const opus_val16 *pcm, int frame_size,

```
....
1317.          if (st->channels == 2 && (rand()&0x1F)==0)
```

Use of Insufficiently Random Values\Path 2:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=15
Status	New

Method opus_encode_native at line 1066 of Tasmota/opus_encoder.c uses a weak method rand 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	Tasmota/opus_encoder.c	Tasmota/opus_encoder.c
Line	1355	1355
Object	rand	rand

Code Snippet
File Name Tasmota/opus_encoder.c
Method opus_int32 opus_encode_native(OpusEncoder *st, const opus_val16 *pcm, int frame_size,

```
....
1355.          if ((rand()&0xF)==0)
```

Use of Insufficiently Random Values\Path 3:

Severity	Low

Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=16
Status	New

Method opus_encode_native at line 1066 of Tasmota/opus_encoder.c uses a weak method rand 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	Tasmota/opus_encoder.c	Tasmota/opus_encoder.c
Line	1357	1357
Object	rand	rand

Code Snippet
File Name Tasmota/opus_encoder.c
Method opus_int32 opus_encode_native(OpusEncoder *st, const opus_val16 *pcm, int frame_size,

```
....
1357.            if ((rand()&0x1)==0)
```

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=1050052&projectid=50042&pathid=24
Status	New

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	1406	1425
Object	sr	sizeof

Code Snippet
File Name Tasmota/opusfile.c
Method static int op_open_seekable2_impl(OggOpusFile *_of){

```
....
1406.    OpusSeekRecord *sr = (OpusSeekRecord*)malloc(64 *
sizeof(OpusSeekRecord));
....
1425.    ret =
op_bisect_forward_serialno(_of,data_offset,sr,sizeof(sr)/sizeof(*sr),
```

Heuristic 2nd Order Buffer Overflow read

Query Path:
CPP\Cx\CPP Heuristic\Heuristic 2nd Order Buffer Overflow read Version:0

Categories

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

Description

Heuristic 2nd Order Buffer Overflow read\Path 1:

Severity	Low
Result State	To Verify
Online Results	http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050052&projectid=50042&pathid=114
Status	New

The size of the buffer used by op_get_data in _nbytes, at line 147 of Tasmota/opusfile.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that op_get_data passes to buffer, at line 147 of Tasmota/opusfile.c, to overwrite the target buffer.

	Source	Destination
File	Tasmota/opusfile.c	Tasmota/opusfile.c
Line	152	152
Object	buffer	_nbytes

Code Snippet
File Name Tasmota/opusfile.c
Method static int op_get_data(OggOpusFile *_of,int _nbytes){

```
....
152.    nbytes=(int)(*_of->callbacks.read)(_of->stream,buffer,_nbytes);
```

Buffer Overflow IndexFromInput

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.

Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

Divide By Zero

Risk

What might happen

When a program divides a number by zero, an exception will be raised. If this exception is not handled by the application, unexpected results may occur, including crashing the application. This can be considered a DoS (Denial of Service) attack, if an external user has control of the value of the denominator or can cause this error to occur.

Cause

How does it happen

The program receives an unexpected value, and uses it for division without filtering, validation, or verifying that the value is not zero. The application does not explicitly handle this error or prevent division by zero from occuring.

General Recommendations

How to avoid it

- Before dividing by an unknown value, validate the number and explicitly ensure it does not evaluate to zero.
- Validate all untrusted input from all sources, in particular verifying that it is not zero before dividing with it.
- Verify output of methods, calculations, dictionary lookups, and so on, and ensure it is not zero before dividing with the result.
- Ensure divide-by-zero errors are caught and handled appropriately.

Source Code Examples

Java

Divide by Zero

```
public float getAverage(HttpServletRequest req) {
      int total = Integer.parseInt(req.getParameter("total"));
      int count = Integer.parseInt(req.getParameter("count"));

      return total / count;
}
```

Checked Division

```
public float getAverage(HttpServletRequest req) {
      int total = Integer.parseInt(req.getParameter("total"));
      int count = Integer.parseInt(req.getParameter("count"));
```

```
	if (count > 0)
		return total / count;
	else
		return 0;
}
```

Buffer Overflow AddressOfLocalVarReturned

Risk

What might happen

A use after free error will cause code to use an area of memory previously assigned with a specific value, which has since been freed and may have been overwritten by another value. This error will likely cause unexpected behavior, memory corruption and crash errors. In some cases where the freed and used section of memory is used to determine execution flow, and the error can be induced by an attacker, this may result in execution of malicious code.

Cause

How does it happen

Pointers to variables allow code to have an address with a set size to a dynamically allocated variable. Eventually, the pointer's destination may become free - either explicitly in code, such as when programmatically freeing this variable, or implicitly, such as when a local variable is returned - once it is returned, the variable's scope is released. Once freed, this memory will be re-used by the application, overwritten with new data. At this point, dereferencing this pointer will potentially resolve newly written and unexpected data.

General Recommendations

How to avoid it

- Do not return local variables or pointers
- Review code to ensure no flow allows use of a pointer after it has been explicitly freed

---

Source Code Examples

CPP

Use of Variable after It was Freed

```
free(input);
printf("%s", input);
```

Use of Pointer to Local Variable That Was Freed On Return

```
int* func1()
{
	int i;
	i = 1;
	return &i;
}

void func2()
```

```
{
	int j;
	j = 5;
}

//..
	int * i = func1();
	printf("%d\r\n", *i); // Output could be 1 or Segmentation Fault
	func2();
	printf("%d\r\n", *i); // Output is 5, which is j's value, as func2() overwrote data in
the stack
//..
```

Buffer Overflow boundcpy WrongSizeParam

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.
Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

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));
    }
}
```

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;
}
```

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;
}
```

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());
```

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	Cigital	External
	updated Time of Introduction		
2008-08-01		KDM Analytics	External
	added/updated white box definitions		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Demonstrative Examples		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Relationships		

BACK TO TOP

Potential Off by One Error in Loops

Risk

What might happen

An off by one error may result in overwriting or over-reading of unintended memory; in most cases, this can result in unexpected behavior and even application crashes. In other cases, where allocation can be controlled by an attacker, a combination of variable assignment and an off by one error can result in execution of malicious code.

Cause

How does it happen

Often when designating variables to memory, a calculation error may occur when determining size or length that is off by one.

For example in loops, when allocating an array of size 2, its cells are counted as 0,1 - therefore, if a For loop iterator on the array is incorrectly set with the start condition i=0 and the continuation condition i<=2, three cells will be accessed instead of 2, and an attempt will be made to write or read cell [2], which was not originally allocated, resulting in potential corruption of memory outside the bounds of the originally assigned array.

Another example occurs when a null-byte terminated string, in the form of a character array, is copied without its terminating null-byte. Without the null-byte, the string representation is unterminated, resulting in certain functions to over-read memory as they expect the missing null terminator.

General Recommendations

How to avoid it

- Always ensure that a given iteration boundary is correct:
  - With array iterations, consider that arrays begin with cell 0 and end with cell n-1, for a size n array.
  - With character arrays and null-byte terminated string representations, consider that the null byte is required and should not be overwritten or ignored; ensure functions in use are not vulnerable to off-by-one, specifically for instances where null-bytes are automatically appended after the buffer, instead of in place of its last character.
- Where possible, use safe functions that manage memory and are not prone to off-by-one errors.

Source Code Examples

CPP

Off-By-One in For Loop

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
for (int i = 0; i <= 5; i++)
{
      ptr[i] = i * 2 + 1; // ptr[5] will be set, but is out of bounds
```

```
}
```

Proper Iteration in For Loop

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
for (int i = 0; i < 5; i++)
{
      ptr[i] = i * 2 + 1; // ptr[0-4] are well defined
}
```

Off-By-One in strncat

```
strncat(buf, input, sizeof(buf) - strlen(buf)); // actual value should be sizeof(buf)-
strlen(buf)-1 - this form will overwrite the terminating nullbyte
```

NULL Pointer Dereference

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
- Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
- Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.

Source Code Examples

Heuristic 2nd Order Buffer Overflow read

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.
Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

Source Code Examples

Indicator of Poor Code Quality

Weakness ID: 398 *(Weakness Class)* **Status:** Draft

Description

Description Summary

The code has features that do not directly introduce a weakness or vulnerability, but indicate that the product has not been carefully developed or maintained.

Extended Description

Programs are more likely to be secure when good development practices are followed. If a program is complex, difficult to maintain, not portable, or shows evidence of neglect, then there is a higher likelihood that weaknesses are buried in the code.

Time of Introduction

- Architecture and Design
- Implementation

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	18	Source Code	**Development Concepts (primary)699**
ChildOf	Weakness Class	710	Coding Standards Violation	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	107	Struts: Unused Validation Form	**Research Concepts (primary)1000**
ParentOf	Weakness Variant	110	Struts: Validator Without Form Field	**Research Concepts (primary)1000**
ParentOf	Category	399	Resource Management Errors	**Development Concepts (primary)699**
ParentOf	Weakness Base	401	Failure to Release Memory Before Removing Last Reference ('Memory Leak')	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	404	Improper Resource Shutdown or Release	Development Concepts699 **Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Variant	415	Double Free	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	416	Use After Free	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Variant	457	Use of Uninitialized Variable	**Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	474	Use of Function with Inconsistent Implementations	**Development Concepts (primary)699 Seven Pernicious Kingdoms (primary)700 Research Concepts (primary)1000**
ParentOf	Weakness Base	475	Undefined Behavior for Input to API	**Development Concepts (primary)699 Seven Pernicious Kingdoms (primary)700**
ParentOf	Weakness Base	476	NULL Pointer	**Development**

			Dereference	**Concepts (primary)699 Seven Pernicious Kingdoms (primary)700 Research Concepts (primary)1000**
ParentOf	Weakness Base	477	Use of Obsolete Functions	**Development Concepts (primary)699 Seven Pernicious Kingdoms (primary)700 Research Concepts (primary)1000**
ParentOf	Weakness Variant	478	Missing Default Case in Switch Statement	**Development Concepts (primary)699**
ParentOf	Weakness Variant	479	Unsafe Function Call from a Signal Handler	**Development Concepts (primary)699**
ParentOf	Weakness Variant	483	Incorrect Block Delimitation	**Development Concepts (primary)699**
ParentOf	Weakness Base	484	Omitted Break Statement in Switch	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Variant	546	Suspicious Comment	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Weakness Variant	547	Use of Hard-coded, Security-relevant Constants	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Weakness Variant	561	Dead Code	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Weakness Base	562	Return of Stack Variable Address	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Variant	563	Unused Variable	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Category	569	Expression Issues	**Development Concepts (primary)699**
ParentOf	Weakness Variant	585	Empty Synchronized Block	**Development Concepts (primary)699 Research Concepts (primary)1000**
ParentOf	Weakness Variant	586	Explicit Call to Finalize()	**Development Concepts (primary)699**
ParentOf	Weakness Variant	617	Reachable Assertion	**Development Concepts (primary)699**
ParentOf	Weakness Base	676	Use of Potentially Dangerous Function	**Development Concepts (primary)699 Research Concepts (primary)1000**
MemberOf	View	700	Seven Pernicious Kingdoms	**Seven Pernicious Kingdoms (primary)700**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name

7 Pernicious Kingdoms			Code Quality

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	7 Pernicious Kingdoms		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Description, Relationships, Taxonomy Mappings		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Relationships		
Previous Entry Names			
Change Date	**Previous Entry Name**		
2008-04-11	Code Quality		

BACK TO TOP

Improper Validation of Array Index

Weakness ID: 129 *(Weakness Base)* **Status:** Draft

Description

Description Summary

The product uses untrusted input when calculating or using an array index, but the product does not validate or incorrectly validates the index to ensure the index references a valid position within the array.

Alternate Terms

out-of-bounds array index

index-out-of-range

array index underflow

Time of Introduction

- Implementation

Applicable Platforms

Languages

C: *(Often)*

C++: *(Often)*

Language-independent

Common Consequences

Scope	Effect
Integrity Availability	Unchecked array indexing will very likely result in the corruption of relevant memory and perhaps instructions, leading to a crash, if the values are outside of the valid memory area.
Integrity	If the memory corrupted is data, rather than instructions, the system will continue to function with improper values.
Confidentiality Integrity	Unchecked array indexing can also trigger out-of-bounds read or write operations, or operations on the wrong objects; i.e., "buffer overflows" are not always the result. This may result in the exposure or modification of sensitive data.
Integrity	If the memory accessible by the attacker can be effectively controlled, it may be possible to execute arbitrary code, as with a standard buffer overflow and possibly without the use of large inputs if a precise index can be controlled.
Integrity Availability Confidentiality	A single fault could allow either an overflow (CWE-788) or underflow (CWE-786) of the array index. What happens next will depend on the type of operation being performed out of bounds, but can expose sensitive information, cause a system crash, or possibly lead to arbitrary code execution.

Likelihood of Exploit

High

Detection Methods

Automated Static Analysis

This weakness can often be detected using automated static analysis tools. Many modern tools use data flow analysis or constraint-based techniques to minimize the number of false positives.

Automated static analysis generally does not account for environmental considerations when reporting out-of-bounds memory operations. This can make it difficult for users to determine which warnings should be investigated first. For example, an analysis tool might report array index errors that originate from command line arguments in a program that is not expected to run with setuid or other special privileges.

Effectiveness: High

This is not a perfect solution, since 100% accuracy and coverage are not feasible.

Automated Dynamic Analysis

This weakness can be detected using dynamic tools and techniques that interact with the software using large test suites with many diverse inputs, such as fuzz testing (fuzzing), robustness testing, and fault injection. The software's operation may slow down, but it should not become unstable, crash, or generate incorrect results.

Black Box

Black box methods might not get the needed code coverage within limited time constraints, and a dynamic test might not produce any noticeable side effects even if it is successful.

Demonstrative Examples

Example 1

The following C/C++ example retrieves the sizes of messages for a pop3 mail server. The message sizes are retrieved from a socket that returns in a buffer the message number and the message size, the message number (num) and size (size) are extracted from the buffer and the message size is placed into an array using the message number for the array index.

(Bad Code)

Example Language: **C**

```
/* capture the sizes of all messages */
int getsizes(int sock, int count, int *sizes) {
...
char buf[BUFFER_SIZE];
int ok;
int num, size;

// read values from socket and added to sizes array
while ((ok = gen_recv(sock, buf, sizeof(buf))) == 0)
{

// continue read from socket until buf only contains '.'
if (DOTLINE(buf))
break;
else if (sscanf(buf, "%d %d", &num, &size) == 2)
sizes[num - 1] = size;
}
...
}
```

In this example the message number retrieved from the buffer could be a value that is outside the allowable range of indices for the array and could possibly be a negative number. Without proper validation of the value to be used for the array index an array overflow could occur and could potentially lead to unauthorized access to memory addresses and system crashes. The value of the array index should be validated to ensure that it is within the allowable range of indices for the array as in the following code.

(Good Code)

Example Language: **C**

```
/* capture the sizes of all messages */
int getsizes(int sock, int count, int *sizes) {
...
char buf[BUFFER_SIZE];
int ok;
int num, size;

// read values from socket and added to sizes array
while ((ok = gen_recv(sock, buf, sizeof(buf))) == 0)
{

// continue read from socket until buf only contains '.'
if (DOTLINE(buf))
```

```
break;
else if (sscanf(buf, "%d %d", &num, &size) == 2) {
if (num > 0 && num <= (unsigned)count)
sizes[num - 1] = size;
else
/* warn about possible attempt to induce buffer overflow */
report(stderr, "Warning: ignoring bogus data for message sizes returned by server.\n");
}
}
...
}
```

Example 2

In the code snippet below, an unchecked integer value is used to reference an object in an array.

(Bad Code)

Example Language: **Java**

```
public String getValue(int index) {
return array[index];
}
```

If index is outside of the range of the array, this may result in an ArrayIndexOutOfBounds Exception being raised.

Example 3

In the following Java example the method displayProductSummary is called from a Web service servlet to retrieve product summary information for display to the user. The servlet obtains the integer value of the product number from the user and passes it to the displayProductSummary method. The displayProductSummary method passes the integer value of the product number to the getProductSummary method which obtains the product summary from the array object containing the project summaries using the integer value of the product number as the array index.

(Bad Code)

Example Language: **Java**

```
// Method called from servlet to obtain product information
public String displayProductSummary(int index) {

String productSummary = new String("");

try {
String productSummary = getProductSummary(index);

} catch (Exception ex) {...}

return productSummary;
}

public String getProductSummary(int index) {
return products[index];
}
```

In this example the integer value used as the array index that is provided by the user may be outside the allowable range of indices for the array which may provide unexpected results or may comes the application to fail. The integer value used for the array index should be validated to ensure that it is within the allowable range of indices for the array as in the following code.

(Good Code)

Example Language: **Java**

```
// Method called from servlet to obtain product information
public String displayProductSummary(int index) {

String productSummary = new String("");
```

```
try {
String productSummary = getProductSummary(index);

} catch (Exception ex) {...}

return productSummary;
}

public String getProductSummary(int index) {
String productSummary = "";

if ((index >= 0) && (index < MAX_PRODUCTS)) {
productSummary = products[index];
}
else {
System.err.println("index is out of bounds");
throw new IndexOutOfBoundsException();
}

return productSummary;
}
```

An alternative in Java would be to use one of the collection objects such as ArrayList that will automatically generate an exception if an attempt is made to access an array index that is out of bounds.

(Good Code)

Example Language: **Java**

```
ArrayList productArray = new ArrayList(MAX_PRODUCTS);
...
try {
productSummary = (String) productArray.get(index);
} catch (IndexOutOfBoundsException ex) {...}
```

Observed Examples

Reference	Description
CVE-2005-0369	large ID in packet used as array index
CVE-2001-1009	negative array index as argument to POP LIST command
CVE-2003-0721	Integer signedness error leads to negative array index
CVE-2004-1189	product does not properly track a count and a maximum number, which can lead to resultant array index overflow.
CVE-2007-5756	chain: device driver for packet-capturing software allows access to an unintended IOCTL with resultant array index error.

Potential Mitigations

Phase: Architecture and Design

Strategies: Input Validation; Libraries or Frameworks

Use an input validation framework such as Struts or the OWASP ESAPI Validation API. If you use Struts, be mindful of weaknesses covered by the CWE-101 category.

Phase: Architecture and Design

For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side, in order to avoid CWE-602. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the server.

Even though client-side checks provide minimal benefits with respect to server-side security, they are still useful. First, they can support intrusion detection. If the server receives input that should have been rejected by the client, then it may be an indication of an attack. Second, client-side error-checking can provide helpful feedback to the user about the expectations for valid input. Third, there may be a reduction in server-side processing time for accidental input errors, although this is typically a small savings.

Phase: Requirements

Strategy: Language Selection

Use a language with features that can automatically mitigate or eliminate out-of-bounds indexing errors.

For example, Ada allows the programmer to constrain the values of a variable and languages such as Java and Ruby will allow the programmer to handle exceptions when an out-of-bounds index is accessed.

Phase: Implementation

Strategy: Input Validation

Assume all input is malicious. Use an "accept known good" input validation strategy (i.e., use a whitelist). Reject any input that does not strictly conform to specifications, or transform it into something that does. Use a blacklist to reject any unexpected inputs and detect potential attacks.

When accessing a user-controlled array index, use a stringent range of values that are within the target array. Make sure that you do not allow negative values to be used. That is, verify the minimum as well as the maximum of the range of acceptable values.

Phase: Implementation

Be especially careful to validate your input when you invoke code that crosses language boundaries, such as from an interpreted language to native code. This could create an unexpected interaction between the language boundaries. Ensure that you are not violating any of the expectations of the language with which you are interfacing. For example, even though Java may not be susceptible to buffer overflows, providing a large argument in a call to native code might trigger an overflow.

Weakness Ordinalities

Ordinality	Description
Resultant	The most common condition situation leading to unchecked array indexing is the use of loop index variables as buffer indexes. If the end condition for the loop is subject to a flaw, the index can grow or shrink unbounded, therefore causing a buffer overflow or underflow. Another common situation leading to this condition is the use of a function's return value, or the resulting value of a calculation directly as an index in to a buffer.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Weakness Class	20	Improper Input Validation	**Development Concepts (primary)699 Research Concepts (primary)1000**
ChildOf	Category	189	Numeric Errors	Development Concepts699
ChildOf	Category	633	Weaknesses that Affect Memory	**Resource-specific Weaknesses (primary)631**
ChildOf	Category	738	CERT C Secure Coding Section 04 - Integers (INT)	**Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734**
ChildOf	Category	740	CERT C Secure Coding Section 06 - Arrays (ARR)	Weaknesses Addressed by the CERT C Secure Coding Standard734
ChildOf	Category	802	2010 Top 25 - Risky Resource Management	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
CanPrecede	Weakness Class	119	Failure to Constrain Operations within the Bounds of a Memory Buffer	Research Concepts1000
CanPrecede	Weakness Variant	789	Uncontrolled Memory Allocation	Research Concepts1000
PeerOf	Weakness Base	124	Buffer Underwrite ('Buffer Underflow')	Research Concepts1000

Theoretical Notes

An improperly validated array index might lead directly to the always-incorrect behavior of "access of array using out-of-bounds index."

Affected Resources

- Memory

f Causal Nature

Explicit

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
CLASP			Unchecked array indexing
PLOVER			INDEX - Array index overflow
CERT C Secure Coding	ARR00-C		Understand how arrays work
CERT C Secure Coding	ARR30-C		Guarantee that array indices are within the valid range
CERT C Secure Coding	ARR38-C		Do not add or subtract an integer to a pointer if the resulting value does not refer to a valid array element
CERT C Secure Coding	INT32-C		Ensure that operations on signed integers do not result in overflow

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
100	Overflow Buffers	

References

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 5, "Array Indexing Errors" Page 144. 2nd Edition. Microsoft. 2002.

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	CLASP		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Sean Eidemiller	Cigital	External
	added/updated demonstrative examples		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Alternate Terms, Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities		
2008-11-24	CWE Content Team	MITRE	Internal
	updated Relationships, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Description, Name, Relationships		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Observed Examples, Other Notes, Potential Mitigations, Theoretical Notes, Weakness Ordinalities		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Demonstrative Examples, Detection Factors, Likelihood of Exploit, Potential Mitigations, References, Related Attack Patterns, Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Related Attack Patterns		
Previous Entry Names			
Change Date	**Previous Entry Name**		
2009-10-29	Unchecked Array Indexing		

BACK TO TOP

Improper Access Control (Authorization)

Weakness ID: 285 *(Weakness Class)* **Status:** Draft

Description

Description Summary

The software does not perform or incorrectly performs access control checks across all potential execution paths.

Extended Description

When access control checks are not applied consistently - or not at all - users are able to access data or perform actions that they should not be allowed to perform. This can lead to a wide range of problems, including information leaks, denial of service, and arbitrary code execution.

Alternate Terms

AuthZ:	"AuthZ" is typically used as an abbreviation of "authorization" within the web application security community. It is also distinct from "AuthC," which is an abbreviation of "authentication." The use of "Auth" as an abbreviation is discouraged, since it could be used for either authentication or authorization.

Time of Introduction

- Architecture and Design
- Implementation
- Operation

Applicable Platforms

Languages

Language-independent

Technology Classes

Web-Server: *(Often)*

Database-Server: *(Often)*

Modes of Introduction

A developer may introduce authorization weaknesses because of a lack of understanding about the underlying technologies. For example, a developer may assume that attackers cannot modify certain inputs such as headers or cookies.

Authorization weaknesses may arise when a single-user application is ported to a multi-user environment.

Common Consequences

Scope	Effect
Confidentiality	An attacker could read sensitive data, either by reading the data directly from a data store that is not properly restricted, or by accessing insufficiently-protected, privileged functionality to read the data.
Integrity	An attacker could modify sensitive data, either by writing the data directly to a data store that is not properly restricted, or by accessing insufficiently-protected, privileged functionality to write the data.
Integrity	An attacker could gain privileges by modifying or reading critical data directly, or by accessing insufficiently-protected, privileged functionality.

Likelihood of Exploit

High

Detection Methods

Automated Static Analysis

Automated static analysis is useful for detecting commonly-used idioms for authorization. A tool may be able to analyze related configuration files, such as .htaccess in Apache web servers, or detect the usage of commonly-used authorization libraries.

Generally, automated static analysis tools have difficulty detecting custom authorization schemes. In addition, the software's design may include some functionality that is accessible to any user and does not require an authorization check; an automated technique that detects the absence of authorization may report false positives.

Effectiveness: Limited

Automated Dynamic Analysis

Automated dynamic analysis may find many or all possible interfaces that do not require authorization, but manual analysis is required to determine if the lack of authorization violates business logic

Manual Analysis

This weakness can be detected using tools and techniques that require manual (human) analysis, such as penetration testing, threat modeling, and interactive tools that allow the tester to record and modify an active session.

Specifically, manual static analysis is useful for evaluating the correctness of custom authorization mechanisms.

Effectiveness: Moderate

These may be more effective than strictly automated techniques. This is especially the case with weaknesses that are related to design and business rules. However, manual efforts might not achieve desired code coverage within limited time constraints.

Demonstrative Examples

Example 1

The following program could be part of a bulletin board system that allows users to send private messages to each other. This program intends to authenticate the user before deciding whether a private message should be displayed. Assume that LookupMessageObject() ensures that the $id argument is numeric, constructs a filename based on that id, and reads the message details from that file. Also assume that the program stores all private messages for all users in the same directory.

(Bad Code)

Example Language: **Perl**

```
sub DisplayPrivateMessage {
my($id) = @_;
my $Message = LookupMessageObject($id);
print "From: " . encodeHTML($Message->{from}) . "<br>\n";
print "Subject: " . encodeHTML($Message->{subject}) . "\n";
print "<hr>\n";
print "Body: " . encodeHTML($Message->{body}) . "\n";
}

my $q = new CGI;
# For purposes of this example, assume that CWE-309 and
# CWE-523 do not apply.
if (! AuthenticateUser($q->param('username'), $q->param('password'))) {
ExitError("invalid username or password");
}

my $id = $q->param('id');
DisplayPrivateMessage($id);
```

While the program properly exits if authentication fails, it does not ensure that the message is addressed to the user. As a result, an authenticated attacker could provide any arbitrary identifier and read private messages that were intended for other users.

One way to avoid this problem would be to ensure that the "to" field in the message object matches the username of the authenticated user.

Observed Examples

Reference	Description
CVE-2009-3168	Web application does not restrict access to admin scripts, allowing authenticated users to reset administrative passwords.

CVE-2009-2960	Web application does not restrict access to admin scripts, allowing authenticated users to modify passwords of other users.
CVE-2009-3597	Web application stores database file under the web root with insufficient access control (CWE-219), allowing direct request.
CVE-2009-2282	Terminal server does not check authorization for guest access.
CVE-2009-3230	Database server does not use appropriate privileges for certain sensitive operations.
CVE-2009-2213	Gateway uses default "Allow" configuration for its authorization settings.
CVE-2009-0034	Chain: product does not properly interpret a configuration option for a system group, allowing users to gain privileges.
CVE-2008-6123	Chain: SNMP product does not properly parse a configuration option for which hosts are allowed to connect, allowing unauthorized IP addresses to connect.
CVE-2008-5027	System monitoring software allows users to bypass authorization by creating custom forms.
CVE-2008-7109	Chain: reliance on client-side security (CWE-602) allows attackers to bypass authorization using a custom client.
CVE-2008-3424	Chain: product does not properly handle wildcards in an authorization policy list, allowing unintended access.
CVE-2009-3781	Content management system does not check access permissions for private files, allowing others to view those files.
CVE-2008-4577	ACL-based protection mechanism treats negative access rights as if they are positive, allowing bypass of intended restrictions.
CVE-2008-6548	Product does not check the ACL of a page accessed using an "include" directive, allowing attackers to read unauthorized files.
CVE-2007-2925	Default ACL list for a DNS server does not set certain ACLs, allowing unauthorized DNS queries.
CVE-2006-6679	Product relies on the X-Forwarded-For HTTP header for authorization, allowing unintended access by spoofing the header.
CVE-2005-3623	OS kernel does not check for a certain privilege before setting ACLs for files.
CVE-2005-2801	Chain: file-system code performs an incorrect comparison (CWE-697), preventing defauls ACLs from being properly applied.
CVE-2001-1155	Chain: product does not properly check the result of a reverse DNS lookup because of operator precedence (CWE-783), allowing bypass of DNS-based access restrictions.

Potential Mitigations

Phase: Architecture and Design

Divide your application into anonymous, normal, privileged, and administrative areas. Reduce the attack surface by carefully mapping roles with data and functionality. Use role-based access control (RBAC) to enforce the roles at the appropriate boundaries.

Note that this approach may not protect against horizontal authorization, i.e., it will not protect a user from attacking others with the same role.

Phase: Architecture and Design

Ensure that you perform access control checks related to your business logic. These checks may be different than the access control checks that you apply to more generic resources such as files, connections, processes, memory, and database records. For example, a database may restrict access for medical records to a specific database user, but each record might only be intended to be accessible to the patient and the patient's doctor.

Phase: Architecture and Design

Strategy: Libraries or Frameworks

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness

easier to avoid.

For example, consider using authorization frameworks such as the JAAS Authorization Framework and the OWASP ESAPI Access Control feature.

Phase: Architecture and Design

For web applications, make sure that the access control mechanism is enforced correctly at the server side on every page. Users should not be able to access any unauthorized functionality or information by simply requesting direct access to that page.

One way to do this is to ensure that all pages containing sensitive information are not cached, and that all such pages restrict access to requests that are accompanied by an active and authenticated session token associated with a user who has the required permissions to access that page.

Phases: System Configuration; Installation

Use the access control capabilities of your operating system and server environment and define your access control lists accordingly. Use a "default deny" policy when defining these ACLs.

Relationships

Nature	Type	ID	Name	View(s) this relationship pertains to
ChildOf	Category	254	Security Features	**Seven Pernicious Kingdoms (primary)700**
ChildOf	Weakness Class	284	Access Control (Authorization) Issues	**Development Concepts (primary)699 Research Concepts (primary)1000**
ChildOf	Category	721	OWASP Top Ten 2007 Category A10 - Failure to Restrict URL Access	**Weaknesses in OWASP Top Ten (2007) (primary)629**
ChildOf	Category	723	OWASP Top Ten 2004 Category A2 - Broken Access Control	**Weaknesses in OWASP Top Ten (2004) (primary)711**
ChildOf	Category	753	2009 Top 25 - Porous Defenses	**Weaknesses in the 2009 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)750**
ChildOf	Category	803	2010 Top 25 - Porous Defenses	**Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800**
ParentOf	Weakness Variant	219	Sensitive Data Under Web Root	**Research Concepts (primary)1000**
ParentOf	Weakness Base	551	Incorrect Behavior Order: Authorization Before Parsing and Canonicalization	**Development Concepts (primary)699** Research Concepts1000
ParentOf	Weakness Class	638	Failure to Use Complete Mediation	Research Concepts1000
ParentOf	Weakness Base	804	Guessable CAPTCHA	**Development Concepts (primary)699 Research Concepts (primary)1000**

Taxonomy Mappings

Mapped Taxonomy Name	Node ID	Fit	Mapped Node Name
7 Pernicious Kingdoms			Missing Access Control
OWASP Top Ten 2007	A10	CWE More Specific	Failure to Restrict URL Access
OWASP Top Ten 2004	A2	CWE More Specific	Broken Access Control

Related Attack Patterns

CAPEC-ID	Attack Pattern Name	*(CAPEC Version: 1.5)*
1	Accessing Functionality Not Properly Constrained by ACLs	
13	Subverting Environment Variable Values	

17	Accessing, Modifying or Executing Executable Files
87	Forceful Browsing
39	Manipulating Opaque Client-based Data Tokens
45	Buffer Overflow via Symbolic Links
51	Poison Web Service Registry
59	Session Credential Falsification through Prediction
60	Reusing Session IDs (aka Session Replay)
77	Manipulating User-Controlled Variables
76	Manipulating Input to File System Calls
104	Cross Zone Scripting

References

NIST. "Role Based Access Control and Role Based Security". <http://csrc.nist.gov/groups/SNS/rbac/>.

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 4, "Authorization" Page 114; Chapter 6, "Determining Appropriate Access Control" Page 171. 2nd Edition. Microsoft. 2002.

Content History

Submissions			
Submission Date	**Submitter**	**Organization**	**Source**
	7 Pernicious Kingdoms		Externally Mined
Modifications			
Modification Date	**Modifier**	**Organization**	**Source**
2008-07-01	Eric Dalci	Cigital	External
	updated Time of Introduction		
2008-08-15		Veracode	External
	Suggested OWASP Top Ten 2004 mapping		
2008-09-08	CWE Content Team	MITRE	Internal
	updated Relationships, Other Notes, Taxonomy Mappings		
2009-01-12	CWE Content Team	MITRE	Internal
	updated Common Consequences, Description, Likelihood of Exploit, Name, Other Notes, Potential Mitigations, References, Relationships		
2009-03-10	CWE Content Team	MITRE	Internal
	updated Potential Mitigations		
2009-05-27	CWE Content Team	MITRE	Internal
	updated Description, Related Attack Patterns		
2009-07-27	CWE Content Team	MITRE	Internal
	updated Relationships		
2009-10-29	CWE Content Team	MITRE	Internal
	updated Type		
2009-12-28	CWE Content Team	MITRE	Internal
	updated Applicable Platforms, Common Consequences, Demonstrative Examples, Detection Factors, Modes of Introduction, Observed Examples, Relationships		
2010-02-16	CWE Content Team	MITRE	Internal
	updated Alternate Terms, Detection Factors, Potential Mitigations, References, Relationships		
2010-04-05	CWE Content Team	MITRE	Internal
	updated Potential Mitigations		
Previous Entry Names			
Change Date	**Previous Entry Name**		
2009-01-12	Missing or Inconsistent Access Control		

BACK TO TOP

Scanned Languages

Language	Hash Number	Change Date
CPP	4541647240435660	6/19/2024
Common	0105849645654507	6/19/2024