# mongoose Scan Report

| | |
|---|---|
| Project Name | mongoose |
| Scan Start | Friday, June 21, 2024 10:59:58 PM |
| Preset | Checkmarx Default |
| Scan Time | 00h:04m:35s |
| Lines Of Code Scanned | 18588 |
| Files Scanned | 8 |
| Report Creation Time | Friday, June 21, 2024 11:05:23 PM |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053 |
| Team | CxServer |
| Checkmarx Version | 8.7.0 |
| Scan Type | Full |
| Source Origin | LocalPath |
| Density | 1/100 (Vulnerabilities/LOC) |
| Visibility | Public |

# Filter Settings

**Severity**

Included: High, Medium, Low, Information

Excluded: None

**Result State**

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

Excluded: None

**Assigned to**

Included: All

**Categories**

Included:

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

Excluded:

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

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

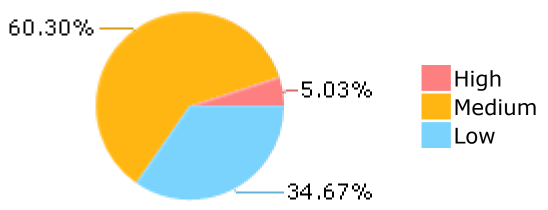
**Results Limit**

Results limit per query was set to 50

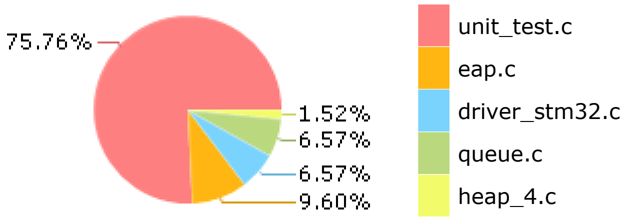
**Selected Queries**

Selected queries are listed in Result Summary

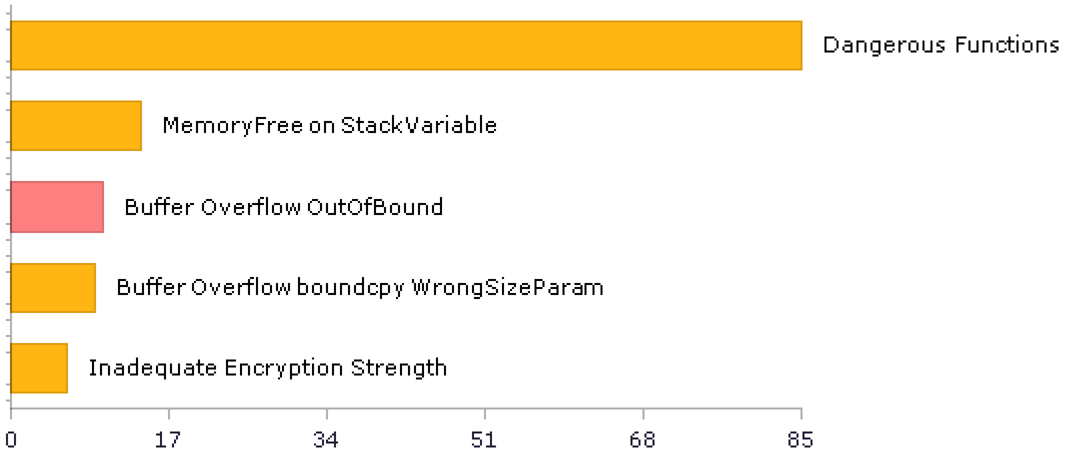
## Result Summary



## Most Vulnerable Files



## Top 5 Vulnerabilities



## Scan Summary - OWASP Top 10 2017

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

| Category | Threat Agent | Exploitability | Weakness Prevalence | Weakness Detectability | Technical Impact | Business Impact | Issues Found | Best Fix Locations |
|---|---|---|---|---|---|---|---|---|
| A1-Injection | App. Specific | EASY | COMMON | EASY | SEVERE | App. Specific | 30 | 17 |
| A2-Broken Authentication | App. Specific | EASY | COMMON | AVERAGE | SEVERE | App. Specific | 1 | 1 |
| A3-Sensitive Data Exposure | App. Specific | AVERAGE | WIDESPREAD | AVERAGE | SEVERE | App. Specific | 6 | 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 | 85 | 85 |
| 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 | 85 | 85 |
| A10-Unvalidated Redirects and Forwards | USERS BROWSERS | AVERAGE | WIDESPREAD | DIFFICULT | MODERATE | AFFECTED DATA AND FUNCTIONS | 0 | 0 |

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

# Scan Summary - PCI DSS v3.2

| Category | Issues Found | Best Fix Locations |
|---|---|---|
| PCI DSS (3.2) - 6.5.1 - Injection flaws - particularly SQL injection | 0 | 0 |
| PCI DSS (3.2) - 6.5.2 - Buffer overflows | 21 | 13 |
| PCI DSS (3.2) - 6.5.3 - Insecure cryptographic storage | 0 | 0 |
| PCI DSS (3.2) - 6.5.4 - Insecure communications | 0 | 0 |
| PCI DSS (3.2) - 6.5.5 - Improper error handling***** | 0 | 0 |
| PCI DSS (3.2) - 6.5.7 - Cross-site scripting (XSS) | 0 | 0 |
| PCI DSS (3.2) - 6.5.8 - Improper access control | 0 | 0 |
| PCI DSS (3.2) - 6.5.9 - Cross-site request forgery | 0 | 0 |
| PCI DSS (3.2) - 6.5.10 - Broken authentication and session management | 0 | 0 |

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

# Scan Summary - FISMA 2014

| Category | Description | Issues Found | Best Fix Locations |
|---|---|---|---|
| Access Control | Organizations must limit information system access to authorized users, processes acting on behalf of authorized users, or devices (including other information systems) and to the types of transactions and functions that authorized users are permitted to exercise. | 0 | 0 |
| Audit And Accountability* | Organizations must: (i) create, protect, and retain information system audit records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful, unauthorized, or inappropriate information system activity; and (ii) ensure that the actions of individual information system users can be uniquely traced to those users so they can be held accountable for their actions. | 0 | 0 |
| Configuration Management | Organizations must: (i) establish and maintain baseline configurations and inventories of organizational information systems (including hardware, software, firmware, and documentation) throughout the respective system development life cycles; and (ii) establish and enforce security configuration settings for information technology products employed in organizational information systems. | 6 | 3 |
| 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. | 1 | 1 |
| Media Protection | Organizations must: (i) protect information system media, both paper and digital; (ii) limit access to information on information system media to authorized users; and (iii) sanitize or destroy information system media before disposal or release for reuse. | 0 | 0 |
| System And Communications Protection | Organizations must: (i) monitor, control, and protect organizational communications (i.e., information transmitted or received by organizational information systems) at the external boundaries and key internal boundaries of the information systems; and (ii) employ architectural designs, software development techniques, and systems engineering principles that promote effective information security within organizational information systems. | 0 | 0 |
| System And Information Integrity | Organizations must: (i) identify, report, and correct information and information system flaws in a timely manner; (ii) provide protection from malicious code at appropriate locations within organizational information systems; and (iii) monitor information system security alerts and advisories and take appropriate actions in response. | 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) | 1 | 1 |
| 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) | 6 | 3 |
| SC-17 Public Key Infrastructure Certificates (P1) | 0 | 0 |
| SC-18 Mobile Code (P2) | 0 | 0 |
| SC-23 Session Authenticity (P1)***** | 0 | 0 |
| SC-28 Protection of Information at Rest (P1) | 0 | 0 |
| SC-4 Information in Shared Resources (P1) | 0 | 0 |
| SC-5 Denial of Service Protection (P1)***** | 14 | 9 |
| SC-8 Transmission Confidentiality and Integrity (P1) | 0 | 0 |
| SI-10 Information Input Validation (P1)***** | 11 | 3 |
| SI-11 Error Handling (P2)***** | 57 | 57 |
| SI-15 Information Output Filtering (P0) | 0 | 0 |
| SI-16 Memory Protection (P1) | 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 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:<br>-Failing to identify the user at all when that should be required<br>-Failure to maintain the user's identity when it is required<br>-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.).<br>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

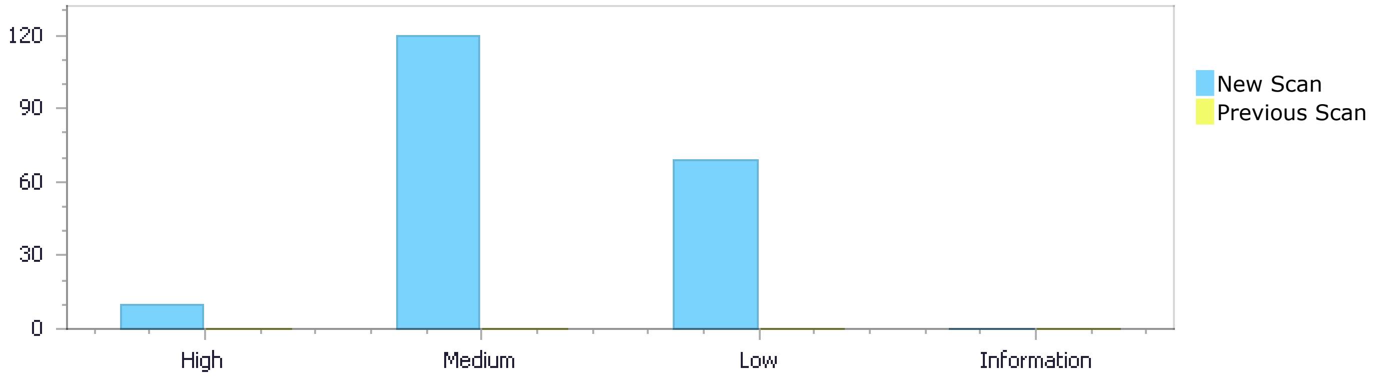
| Category | Issues Found | Best Fix Locations |
|---|---|---|
| Must audit | 0 | 0 |
| Check | 0 | 0 |
| Optional | 0 | 0 |

## Results Distribution By Status

First scan of the project

| | High | Medium | Low | Information | Total |
|---|---|---|---|---|---|
| New Issues | 10 | 120 | 69 | 0 | 199 |
| Recurrent Issues | 0 | 0 | 0 | 0 | 0 |
| Total | 10 | 120 | 69 | 0 | 199 |

| | | | | | |
|---|---|---|---|---|---|
| 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 | 10 | 120 | 69 | 0 | 199 |
| Urgent | 0 | 0 | 0 | 0 | 0 |
| Proposed Not Exploitable | 0 | 0 | 0 | 0 | 0 |
| Total | 10 | 120 | 69 | 0 | 199 |

## Result Summary

| Vulnerability Type | Occurrences | Severity |
|---|---|---|
| Buffer Overflow OutOfBound | 10 | High |
| Dangerous Functions | 85 | Medium |
| MemoryFree on StackVariable | 14 | Medium |
| Buffer Overflow boundcpy WrongSizeParam | 9 | Medium |
| Inadequate Encryption Strength | 6 | Medium |

| | | |
|---|---|---|
| Use of Zero Initialized Pointer | 3 | Medium |
| Buffer Overflow AddressOfLocalVarReturned | 2 | Medium |
| Wrong Size t Allocation | 1 | Medium |
| Unchecked Return Value | 57 | Low |
| NULL Pointer Dereference | 9 | Low |
| Improper Resource Access Authorization | 1 | Low |
| TOCTOU | 1 | Low |
| Unchecked Array Index | 1 | Low |

## 10 Most Vulnerable Files

### High and Medium Vulnerabilities

| File Name | Issues Found |
|---|---|
| mongoose/unit_test.c | 92 |
| mongoose/eap.c | 17 |
| mongoose/driver_stm32.c | 13 |
| mongoose/queue.c | 5 |
| mongoose/heap_4.c | 2 |
| mongoose/net.c | 1 |

# Scan Results Details

## Buffer Overflow OutOfBound

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

### Categories

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

*Description*

**Buffer Overflow OutOfBound\Path 1:**

| | |
|---|---|
| Severity | High |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=1 |
| Status | New |

The size of the buffer used by mg_tcpip_driver_stm32_init in s_rxdesc, at line 119 of mongoose/driver_stm32.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that descriptors passes to s_rxdesc, at line 39 of mongoose/driver_stm32.c, to overwrite the target buffer.

| | Source | Destination |
|---|---|---|
| File | mongoose/driver_stm32.c | mongoose/driver_stm32.c |
| Line | 39 | 127 |
| Object | s_rxdesc | s_rxdesc |

Code Snippet
File Name mongoose/driver_stm32.c
Method static uint32_t s_rxdesc[ETH_DESC_CNT][ETH_DS]; // RX descriptors

```
....
39.  static uint32_t s_rxdesc[ETH_DESC_CNT][ETH_DS];      // RX
descriptors
```

File Name mongoose/driver_stm32.c

Method static bool mg_tcpip_driver_stm32_init(struct mg_tcpip_if *ifp) {

```
....
127.      s_rxdesc[i][1] = sizeof(s_rxbuf[i]) | BIT(14);      // 2nd
address chained
```

**Buffer Overflow OutOfBound\Path 2:**

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

| | |
|---|---|
| | BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=2 |
| Status | New |

The size of the buffer used by mg_tcpip_driver_stm32_init in s_rxdesc, at line 119 of mongoose/driver_stm32.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that descriptors passes to s_rxdesc, at line 39 of mongoose/driver_stm32.c, to overwrite the target buffer.

| | Source | Destination |
|---|---|---|
| File | mongoose/driver_stm32.c | mongoose/driver_stm32.c |
| Line | 39 | 126 |
| Object | s_rxdesc | s_rxdesc |

Code Snippet

File Name mongoose/driver_stm32.c

Method static uint32_t s_rxdesc[ETH_DESC_CNT][ETH_DS]; // RX descriptors

```
....
39.  static uint32_t s_rxdesc[ETH_DESC_CNT][ETH_DS];      // RX
descriptors
```

File Name mongoose/driver_stm32.c

Method static bool mg_tcpip_driver_stm32_init(struct mg_tcpip_if *ifp) {

```
....
126.      s_rxdesc[i][0] = BIT(31);                       // Own
```

**Buffer Overflow OutOfBound\Path 3:**

| | |
|---|---|
| Severity | High |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=3 |
| Status | New |

The size of the buffer used by mg_tcpip_driver_stm32_init in s_rxdesc, at line 119 of mongoose/driver_stm32.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that descriptors passes to s_rxdesc, at line 39 of mongoose/driver_stm32.c, to overwrite the target buffer.

| | Source | Destination |
|---|---|---|
| File | mongoose/driver_stm32.c | mongoose/driver_stm32.c |
| Line | 39 | 128 |
| Object | s_rxdesc | s_rxdesc |

Code Snippet

File Name mongoose/driver_stm32.c

Method static uint32_t s_rxdesc[ETH_DESC_CNT][ETH_DS]; // RX descriptors

```
....
39.  static uint32_t s_rxdesc[ETH_DESC_CNT][ETH_DS];      // RX
descriptors
```

▼

| | |
|---|---|
| File Name | mongoose/driver_stm32.c |
| Method | static bool mg_tcpip_driver_stm32_init(struct mg_tcpip_if *ifp) { |

```
....
128.      s_rxdesc[i][2] = (uint32_t) (uintptr_t) s_rxbuf[i];  // Point
to data buffer
```

### Buffer Overflow OutOfBound\Path 4:

| | |
|---|---|
| Severity | High |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=4 |
| Status | New |

The size of the buffer used by mg_tcpip_driver_stm32_init in s_rxdesc, at line 119 of mongoose/driver_stm32.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that descriptors passes to s_rxdesc, at line 39 of mongoose/driver_stm32.c, to overwrite the target buffer.

| | Source | Destination |
|---|---|---|
| File | mongoose/driver_stm32.c | mongoose/driver_stm32.c |
| Line | 39 | 129 |
| Object | s_rxdesc | s_rxdesc |

Code Snippet

| | |
|---|---|
| File Name | mongoose/driver_stm32.c |
| Method | static uint32_t s_rxdesc[ETH_DESC_CNT][ETH_DS];      // RX descriptors |

```
....
39.  static uint32_t s_rxdesc[ETH_DESC_CNT][ETH_DS];      // RX
descriptors
```

▼

| | |
|---|---|
| File Name | mongoose/driver_stm32.c |
| Method | static bool mg_tcpip_driver_stm32_init(struct mg_tcpip_if *ifp) { |

```
....
129.      s_rxdesc[i][3] =
```

### Buffer Overflow OutOfBound\Path 5:

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

| | |
|---|---|
| | [BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=5](BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=5) |
| Status | New |

The size of the buffer used by mg_tcpip_driver_stm32_init in s_rxdesc, at line 119 of mongoose/driver_stm32.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that buffers passes to s_rxbuf, at line 41 of mongoose/driver_stm32.c, to overwrite the target buffer.

| | Source | Destination |
|---|---|---|
| File | mongoose/driver_stm32.c | mongoose/driver_stm32.c |
| Line | 41 | 127 |
| Object | s_rxbuf | s_rxdesc |

Code Snippet

File Name mongoose/driver_stm32.c

Method static uint8_t s_rxbuf[ETH_DESC_CNT][ETH_PKT_SIZE];  // RX ethernet buffers

```
....
41.  static uint8_t s_rxbuf[ETH_DESC_CNT][ETH_PKT_SIZE];  // RX ethernet
buffers
```

File Name mongoose/driver_stm32.c

Method static bool mg_tcpip_driver_stm32_init(struct mg_tcpip_if *ifp) {

```
....
127.      s_rxdesc[i][1] = sizeof(s_rxbuf[i]) | BIT(14);       // 2nd
address chained
```

**Buffer Overflow OutOfBound\Path 6:**

| | |
|---|---|
| Severity | High |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=6](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=6) |
| Status | New |

The size of the buffer used by mg_tcpip_driver_stm32_init in s_rxdesc, at line 119 of mongoose/driver_stm32.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that buffers passes to s_rxbuf, at line 41 of mongoose/driver_stm32.c, to overwrite the target buffer.

| | Source | Destination |
|---|---|---|
| File | mongoose/driver_stm32.c | mongoose/driver_stm32.c |
| Line | 41 | 126 |
| Object | s_rxbuf | s_rxdesc |

Code Snippet

File Name mongoose/driver_stm32.c

Method static uint8_t s_rxbuf[ETH_DESC_CNT][ETH_PKT_SIZE];  // RX ethernet buffers

```
....
41.  static uint8_t s_rxbuf[ETH_DESC_CNT][ETH_PKT_SIZE];  // RX ethernet
buffers
```

| | |
|---|---|
| File Name | mongoose/driver_stm32.c |
| Method | static bool mg_tcpip_driver_stm32_init(struct mg_tcpip_if *ifp) { |

```
....
126.      s_rxdesc[i][0] = BIT(31);                            // Own
```

**Buffer Overflow OutOfBound\Path 7:**

| | |
|---|---|
| Severity | High |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=7 |
| Status | New |

The size of the buffer used by mg_tcpip_driver_stm32_init in i, at line 119 of mongoose/driver_stm32.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that buffers passes to s_rxbuf, at line 41 of mongoose/driver_stm32.c, to overwrite the target buffer.

| | Source | Destination |
|---|---|---|
| File | mongoose/driver_stm32.c | mongoose/driver_stm32.c |
| Line | 41 | 127 |
| Object | s_rxbuf | i |

Code Snippet

| | |
|---|---|
| File Name | mongoose/driver_stm32.c |
| Method | static uint8_t s_rxbuf[ETH_DESC_CNT][ETH_PKT_SIZE];  // RX ethernet buffers |

```
....
41.  static uint8_t s_rxbuf[ETH_DESC_CNT][ETH_PKT_SIZE];  // RX ethernet
buffers
```

| | |
|---|---|
| File Name | mongoose/driver_stm32.c |
| Method | static bool mg_tcpip_driver_stm32_init(struct mg_tcpip_if *ifp) { |

```
....
127.      s_rxdesc[i][1] = sizeof(s_rxbuf[i]) | BIT(14);       // 2nd
address chained
```

**Buffer Overflow OutOfBound\Path 8:**

| | |
|---|---|
| Severity | High |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=500 |

| | |
|---|---|
| | 53&pathid=8 |
| Status | New |

The size of the buffer used by mg_tcpip_driver_stm32_init in s_rxdesc, at line 119 of mongoose/driver_stm32.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that buffers passes to s_rxbuf, at line 41 of mongoose/driver_stm32.c, to overwrite the target buffer.

| | Source | Destination |
|---|---|---|
| File | mongoose/driver_stm32.c | mongoose/driver_stm32.c |
| Line | 41 | 128 |
| Object | s_rxbuf | s_rxdesc |

Code Snippet
File Name mongoose/driver_stm32.c
Method static uint8_t s_rxbuf[ETH_DESC_CNT][ETH_PKT_SIZE];  // RX ethernet buffers

```
....
41.  static uint8_t s_rxbuf[ETH_DESC_CNT][ETH_PKT_SIZE];  // RX ethernet
buffers
```

File Name mongoose/driver_stm32.c

Method static bool mg_tcpip_driver_stm32_init(struct mg_tcpip_if *ifp) {

```
....
128.      s_rxdesc[i][2] = (uint32_t) (uintptr_t) s_rxbuf[i];  // Point
to data buffer
```

**Buffer Overflow OutOfBound\Path 9:**

| | |
|---|---|
| Severity | High |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=9 |
| Status | New |

The size of the buffer used by mg_tcpip_driver_stm32_init in i, at line 119 of mongoose/driver_stm32.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that buffers passes to s_rxbuf, at line 41 of mongoose/driver_stm32.c, to overwrite the target buffer.

| | Source | Destination |
|---|---|---|
| File | mongoose/driver_stm32.c | mongoose/driver_stm32.c |
| Line | 41 | 128 |
| Object | s_rxbuf | i |

Code Snippet
File Name mongoose/driver_stm32.c
Method static uint8_t s_rxbuf[ETH_DESC_CNT][ETH_PKT_SIZE];  // RX ethernet buffers

```
....
41.  static uint8_t s_rxbuf[ETH_DESC_CNT][ETH_PKT_SIZE];  // RX ethernet
buffers
```

▼

| | |
|---|---|
| File Name | mongoose/driver_stm32.c |
| Method | static bool mg_tcpip_driver_stm32_init(struct mg_tcpip_if *ifp) { |

```
....
128.       s_rxdesc[i][2] = (uint32_t) (uintptr_t) s_rxbuf[i];  // Point
to data buffer
```

**Buffer Overflow OutOfBound\Path 10:**

| | |
|---|---|
| Severity | High |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=10](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=10) |
| Status | New |

The size of the buffer used by mg_tcpip_driver_stm32_init in s_rxdesc, at line 119 of mongoose/driver_stm32.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that buffers passes to s_rxbuf, at line 41 of mongoose/driver_stm32.c, to overwrite the target buffer.

| | Source | Destination |
|---|---|---|
| File | mongoose/driver_stm32.c | mongoose/driver_stm32.c |
| Line | 41 | 129 |
| Object | s_rxbuf | s_rxdesc |

Code Snippet

| | |
|---|---|
| File Name | mongoose/driver_stm32.c |
| Method | static uint8_t s_rxbuf[ETH_DESC_CNT][ETH_PKT_SIZE];  // RX ethernet buffers |

```
....
41.  static uint8_t s_rxbuf[ETH_DESC_CNT][ETH_PKT_SIZE];  // RX ethernet
buffers
```

▼

| | |
|---|---|
| File Name | mongoose/driver_stm32.c |
| Method | static bool mg_tcpip_driver_stm32_init(struct mg_tcpip_if *ifp) { |

```
....
129.       s_rxdesc[i][3] =
```

## 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=1050063&projectid=50053&pathid=104](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=104) |
| Status | New |

The dangerous function, memcpy, was found in use at line 169 in mongoose/driver_stm32.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/driver_stm32.c | mongoose/driver_stm32.c |
| Line | 180 | 180 |
| Object | memcpy | memcpy |

Code Snippet

| | |
|---|---|
| File Name | mongoose/driver_stm32.c |
| Method | static size_t mg_tcpip_driver_stm32_tx(const void *buf, size_t len, |

```
....
180.      memcpy(s_txbuf[s_txno], buf, len);     // Copy data
```

**Dangerous Functions\Path 2:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=105](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=105) |
| Status | New |

The dangerous function, memcpy, was found in use at line 2070 in mongoose/queue.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/queue.c | mongoose/queue.c |
| Line | 2098 | 2098 |
| Object | memcpy | memcpy |

Code Snippet

| | |
|---|---|
| File Name | mongoose/queue.c |
| Method | static BaseType_t prvCopyDataToQueue( Queue_t * const pxQueue, const void *pvItemToQueue, const BaseType_t xPosition ) |

```
....
2098.               ( void ) memcpy( ( void * ) pxQueue->pcWriteTo,
pvItemToQueue, ( size_t ) pxQueue->uxItemSize ); /*lint !e961 !e418
!e9087 MISRA exception as the casts are only redundant for some ports,
plus previous logic ensures a null pointer can only be passed to
memcpy() if the copy size is 0.  Cast to void required by function
signature and safe as no alignment requirement and copy length specified
in bytes. */
```

### Dangerous Functions\Path 3:

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=106 |
| Status | New |

The dangerous function, memcpy, was found in use at line 2070 in mongoose/queue.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/queue.c | mongoose/queue.c |
| Line | 2111 | 2111 |
| Object | memcpy | memcpy |

Code Snippet

| | |
|---|---|
| File Name | mongoose/queue.c |
| Method | static BaseType_t prvCopyDataToQueue( Queue_t * const pxQueue, const void *pvItemToQueue, const BaseType_t xPosition ) |

```
....
2111.               ( void ) memcpy( ( void * ) pxQueue-
>u.xQueue.pcReadFrom, pvItemToQueue, ( size_t ) pxQueue->uxItemSize );
/*lint !e961 !e9087 !e418 MISRA exception as the casts are only
redundant for some ports.  Cast to void required by function signature
and safe as no alignment requirement and copy length specified in bytes.
Assert checks null pointer only used when length is 0. */
```

### Dangerous Functions\Path 4:

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=107 |
| Status | New |

The dangerous function, memcpy, was found in use at line 2149 in mongoose/queue.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/queue.c | mongoose/queue.c |

| | | |
|---|---|---|
| Line | 2162 | 2162 |
| Object | memcpy | memcpy |

Code Snippet
File Name mongoose/queue.c
Method static void prvCopyDataFromQueue( Queue_t * const pxQueue, void * const pvBuffer )

```
....
2162.              ( void ) memcpy( ( void * ) pvBuffer, ( void * )
pxQueue->u.xQueue.pcReadFrom, ( size_t ) pxQueue->uxItemSize ); /*lint
!e961 !e418 !e9087 MISRA exception as the casts are only redundant for
some ports.  Also previous logic ensures a null pointer can only be
passed to memcpy() when the count is 0.  Cast to void required by
function signature and safe as no alignment requirement and copy length
specified in bytes. */
```

**Dangerous Functions\Path 5:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=108 |
| Status | New |

The dangerous function, memcpy, was found in use at line 2446 in mongoose/queue.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/queue.c | mongoose/queue.c |
| Line | 2496 | 2496 |
| Object | memcpy | memcpy |

Code Snippet
File Name mongoose/queue.c
Method BaseType_t xQueueCRReceive( QueueHandle_t xQueue, void *pvBuffer, TickType_t xTicksToWait )

```
....
2496.                           ( void ) memcpy( ( void * ) pvBuffer, (
void * ) pxQueue->u.xQueue.pcReadFrom, ( unsigned ) pxQueue->uxItemSize
);
```

**Dangerous Functions\Path 6:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=109 |
| Status | New |

The dangerous function, memcpy, was found in use at line 2584 in mongoose/queue.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/queue.c | mongoose/queue.c |
| Line | 2604 | 2604 |
| Object | memcpy | memcpy |

Code Snippet

| | |
|---|---|
| File Name | mongoose/queue.c |
| Method | BaseType_t xQueueCRReceiveFromISR( QueueHandle_t xQueue, void *pvBuffer, BaseType_t *pxCoRoutineWoken ) |

```
....
2604.                           ( void ) memcpy( ( void * ) pvBuffer, ( void * )
pxQueue->u.xQueue.pcReadFrom, ( unsigned ) pxQueue->uxItemSize );
```

**Dangerous Functions\Path 7:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=110 |
| Status | New |

The dangerous function, memcpy, was found in use at line 1664 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1897 | 1897 |
| Object | memcpy | memcpy |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1897.      memcpy(a.ip, &addr, sizeof(uint32_t));
```

**Dangerous Functions\Path 8:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=111 |
| Status | New |

The dangerous function, memcpy, was found in use at line 1993 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 2010 | 2010 |
| Object | memcpy | memcpy |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_util(void) { |

```
....
2010.     memcpy(&ipv4, a.ip, sizeof(ipv4));
```

**Dangerous Functions\Path 9:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=112 |
| Status | New |

The dangerous function, memcpy, was found in use at line 2940 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 2949 | 2949 |
| Object | memcpy | memcpy |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void producer(void *param) { |

```
....
2949.       memcpy(buf, &tmp[ofs], len);
```

**Dangerous Functions\Path 10:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=113 |
| Status | New |

The dangerous function, strcat, was found in use at line 1250 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1255 | 1255 |
| Object | strcat | strcat |

Code Snippet
File Name mongoose/unit_test.c
Method static void f4(struct mg_connection *c, int ev, void *ev_data, void *fn_data) {

```
....
1255.       strcat((char *) fn_data, "m");
```

**Dangerous Functions\Path 11:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=114 |
| Status | New |

The dangerous function, strcat, was found in use at line 1250 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1258 | 1258 |
| Object | strcat | strcat |

Code Snippet
File Name mongoose/unit_test.c
Method static void f4(struct mg_connection *c, int ev, void *ev_data, void *fn_data) {

```
....
1258.       strcat((char *) fn_data, "f");
```

**Dangerous Functions\Path 12:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=115 |
| Status | New |

The dangerous function, strcat, was found in use at line 1250 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |

| | | |
|---|---|---|
| Line | 1260 | 1260 |
| Object | strcat | strcat |

Code Snippet
File Name mongoose/unit_test.c
Method static void f4(struct mg_connection *c, int ev, void *ev_data, void *fn_data) {

```
....
1260.       strcat((char *) fn_data, "c");
```

**Dangerous Functions\Path 13:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=116 |
| Status | New |

The dangerous function, strcat, was found in use at line 1264 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1270 | 1270 |
| Object | strcat | strcat |

Code Snippet
File Name mongoose/unit_test.c
Method static void f4c(struct mg_connection *c, int ev, void *ev_data, void *fn_data) {

```
....
1270.       strcat((char *) fn_data, "m");
```

**Dangerous Functions\Path 14:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=117 |
| Status | New |

The dangerous function, strcat, was found in use at line 1264 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1274 | 1274 |
| Object | strcat | strcat |

Code Snippet
File Name mongoose/unit_test.c
Method static void f4c(struct mg_connection *c, int ev, void *ev_data, void *fn_data) {

```
....
1274.       strcat((char *) fn_data, "f");
```

**Dangerous Functions\Path 15:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=118 |
| Status | New |

The dangerous function, strcat, was found in use at line 1264 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1276 | 1276 |
| Object | strcat | strcat |

Code Snippet
File Name mongoose/unit_test.c
Method static void f4c(struct mg_connection *c, int ev, void *ev_data, void *fn_data) {

```
....
1276.       strcat((char *) fn_data, "c");
```

**Dangerous Functions\Path 16:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=119 |
| Status | New |

The dangerous function, strcpy, was found in use at line 1311 in mongoose/eap.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/eap.c | mongoose/eap.c |
| Line | 1365 | 1365 |
| Object | strcpy | strcpy |

Code Snippet
File Name mongoose/eap.c

| | |
|---|---|
| Method | static void eap_request(ppp_pcb *pcb, u_char *inp, int id, int len) { |

```
....
1365.                          strcpy(rhostname, SRP_PSEUDO_ID);
```

**Dangerous Functions\Path 17:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=120 |
| Status | New |

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

| | Source | Destination |
|---|---|---|
| File | mongoose/eap.c | mongoose/eap.c |
| Line | 232 | 232 |
| Object | strlen | strlen |

Code Snippet

| | |
|---|---|
| File Name | mongoose/eap.c |
| Method | void eap_authwithpeer(ppp_pcb *pcb, const char *localname) { |

```
....
232.          pcb->eap.es_client.ea_namelen = strlen(localname);
```

**Dangerous Functions\Path 18:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=121 |
| Status | New |

The dangerous function, strlen, was found in use at line 315 in mongoose/eap.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/eap.c | mongoose/eap.c |
| Line | 328 | 328 |
| Object | strlen | strlen |

Code Snippet

| | |
|---|---|
| File Name | mongoose/eap.c |
| Method | pncrypt_setkey(int timeoffs) |

```
....
328.        SHA1Update(&ctxt, pn_secret, strlen(pn_secret));
```

### Dangerous Functions\Path 19:

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=122 |
| Status | New |

The dangerous function, strlen, was found in use at line 315 in mongoose/eap.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/eap.c | mongoose/eap.c |
| Line | 330 | 330 |
| Object | strlen | strlen |

Code Snippet

| | |
|---|---|
| File Name | mongoose/eap.c |
| Method | pncrypt_setkey(int timeoffs) |

```
....
330.        SHA1Update(&ctxt, tbuf, strlen(tbuf));
```

### Dangerous Functions\Path 20:

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=123 |
| Status | New |

The dangerous function, strlen, was found in use at line 638 in mongoose/eap.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/eap.c | mongoose/eap.c |
| Line | 664 | 664 |
| Object | strlen | strlen |

Code Snippet

| | |
|---|---|
| File Name | mongoose/eap.c |
| Method | static void eap_send_request(ppp_pcb *pcb) { |

```
....
664.                    int len = (int)strlen(pcb->remote_name);
```

**Dangerous Functions\Path 21:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=124 |
| Status | New |

The dangerous function, strlen, was found in use at line 638 in mongoose/eap.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/eap.c | mongoose/eap.c |
| Line | 707 | 707 |
| Object | strlen | strlen |

Code Snippet

| | |
|---|---|
| File Name | mongoose/eap.c |
| Method | static void eap_send_request(ppp_pcb *pcb) { |

```
....
707.               len = strlen(str);
```

**Dangerous Functions\Path 22:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=125 |
| Status | New |

The dangerous function, strlen, was found in use at line 876 in mongoose/eap.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/eap.c | mongoose/eap.c |
| Line | 880 | 880 |
| Object | strlen | strlen |

Code Snippet

| | |
|---|---|
| File Name | mongoose/eap.c |
| Method | void eap_authpeer(ppp_pcb *pcb, const char *localname) { |

```
....
880.        pcb->eap.es_server.ea_namelen = strlen(localname);
```

**Dangerous Functions\Path 23:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=126 |
| Status | New |

The dangerous function, strlen, was found in use at line 1197 in mongoose/eap.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/eap.c | mongoose/eap.c |
| Line | 1210 | 1210 |
| Object | strlen | strlen |

Code Snippet

| | |
|---|---|
| File Name | mongoose/eap.c |
| Method | name_of_pn_file() |

```
....
1210.       pl = strlen(user) + strlen(file) + 2;
```

**Dangerous Functions\Path 24:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=127 |
| Status | New |

The dangerous function, strlen, was found in use at line 1197 in mongoose/eap.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/eap.c | mongoose/eap.c |
| Line | 1210 | 1210 |
| Object | strlen | strlen |

Code Snippet

| | |
|---|---|
| File Name | mongoose/eap.c |
| Method | name_of_pn_file() |

```
....
1210.        pl = strlen(user) + strlen(file) + 2;
```

**Dangerous Functions\Path 25:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=128 |
| Status | New |

The dangerous function, strlen, was found in use at line 1311 in mongoose/eap.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/eap.c | mongoose/eap.c |
| Line | 1515 | 1515 |
| Object | strlen | strlen |

Code Snippet
File Name mongoose/eap.c
Method static void eap_request(ppp_pcb *pcb, u_char *inp, int id, int len) {

```
....
1515.                        rhostnamelen = (int)strlen(rhostname);
```

**Dangerous Functions\Path 26:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=129 |
| Status | New |

The dangerous function, strlen, was found in use at line 356 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 397 | 397 |
| Object | strlen | strlen |

Code Snippet
File Name mongoose/unit_test.c
Method static void mqtt_cb(struct mg_connection *c, int ev, void *evd, void *fnd) {

```
....
397.                prop.val.len == strlen("test_content_val_2"));
```

**Dangerous Functions\Path 27:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=130 |
| Status | New |

The dangerous function, strlen, was found in use at line 356 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 402 | 402 |
| Object | strlen | strlen |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void mqtt_cb(struct mg_connection *c, int ev, void *evd, void *fnd) { |

```
....
402.                prop.key.len == strlen("test_key_1"));
```

**Dangerous Functions\Path 28:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=131 |
| Status | New |

The dangerous function, strlen, was found in use at line 356 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 404 | 404 |
| Object | strlen | strlen |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void mqtt_cb(struct mg_connection *c, int ev, void *evd, void *fnd) { |

```
....
404.                  prop.val.len == strlen("test_value_1"));
```

**Dangerous Functions\Path 29:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=132](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=132) |
| Status | New |

The dangerous function, strlen, was found in use at line 356 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 409 | 409 |
| Object | strlen | strlen |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void mqtt_cb(struct mg_connection *c, int ev, void *evd, void *fnd) { |

```
....
409.                  prop.key.len == strlen("test_key_2"));
```

**Dangerous Functions\Path 30:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=133](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=133) |
| Status | New |

The dangerous function, strlen, was found in use at line 356 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 411 | 411 |
| Object | strlen | strlen |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void mqtt_cb(struct mg_connection *c, int ev, void *evd, void *fnd) { |

```
....
411.                  prop.val.len == strlen("test_value_2"));
```

**Dangerous Functions\Path 31:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=134](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=134) |
| Status | New |

The dangerous function, strlen, was found in use at line 459 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 461 | 461 |
| Object | strlen | strlen |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void check_mqtt_message(struct mg_mqtt_opts *opts, |

```
....
461.     if (opts->topic.len != strlen(data->topic) ||
```

**Dangerous Functions\Path 32:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=135](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=135) |
| Status | New |

The dangerous function, strlen, was found in use at line 459 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 466 | 466 |
| Object | strlen | strlen |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void check_mqtt_message(struct mg_mqtt_opts *opts, |

```
....
466.     if (*data->msg != 'X' || opts->message.len != (strlen(&data-
>msg[1])) ||
```

**Dangerous Functions\Path 33:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=136 |
| Status | New |

The dangerous function, strlen, was found in use at line 735 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 738 | 738 |
| Object | strlen | strlen |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static struct mg_http_message gethm(const char *buf) { |

```
....
738.     mg_http_parse(buf, strlen(buf), &hm);
```

**Dangerous Functions\Path 34:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=137 |
| Status | New |

The dangerous function, strlen, was found in use at line 742 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 745 | 745 |
| Object | strlen | strlen |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static int cmpbody(const char *buf, const char *str) { |

```
....
745.      size_t len = strlen(buf);
```

### Dangerous Functions\Path 35:

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=138 |
| Status | New |

The dangerous function, strlen, was found in use at line 835 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 883 | 883 |
| Object | strlen | strlen |

| | |
|---|---|
| Code Snippet | |
| File Name | mongoose/unit_test.c |
| Method | static void test_http_server(void) { |

```
....
883.      mg_hexdump(buf, strlen(buf));
```

### Dangerous Functions\Path 36:

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=139 |
| Status | New |

The dangerous function, strlen, was found in use at line 835 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 913 | 913 |
| Object | strlen | strlen |

| | |
|---|---|
| Code Snippet | |
| File Name | mongoose/unit_test.c |
| Method | static void test_http_server(void) { |

```
....
913.        ASSERT(mg_http_parse(buf, strlen(buf), &hm) > 0);
```

**Dangerous Functions\Path 37:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=140 |
| Status | New |

The dangerous function, strlen, was found in use at line 835 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 924 | 924 |
| Object | strlen | strlen |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_http_server(void) {

```
....
924.        mg_http_parse(buf, strlen(buf), &hm);
```

**Dangerous Functions\Path 38:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=141 |
| Status | New |

The dangerous function, strlen, was found in use at line 835 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 952 | 952 |
| Object | strlen | strlen |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_http_server(void) {

```
....
952.        mg_http_parse(buf, strlen(buf), &hm);
```

### Dangerous Functions\Path 39:

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=142 |
| Status | New |

The dangerous function, strlen, was found in use at line 835 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 973 | 973 |
| Object | strlen | strlen |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_http_server(void) {

```
....
973.        mg_http_parse(buf, strlen(buf), &hm);
```

### Dangerous Functions\Path 40:

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=143 |
| Status | New |

The dangerous function, strlen, was found in use at line 835 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1000 | 1000 |
| Object | strlen | strlen |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_http_server(void) {

```
....
1000.       mg_http_parse(buf, strlen(buf), &hm);
```

**Dangerous Functions\Path 41:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=144 |
| Status | New |

The dangerous function, strlen, was found in use at line 835 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1006 | 1006 |
| Object | strlen | strlen |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_http_server(void) {

```
....
1006.       mg_http_parse(buf, strlen(buf), &hm);
```

**Dangerous Functions\Path 42:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=145 |
| Status | New |

The dangerous function, strlen, was found in use at line 835 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1012 | 1012 |
| Object | strlen | strlen |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_http_server(void) {

```
....
1012.        mg_http_parse(buf, strlen(buf), &hm);
```

**Dangerous Functions\Path 43:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=146 |
| Status | New |

The dangerous function, strlen, was found in use at line 835 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1018 | 1018 |
| Object | strlen | strlen |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_http_server(void) { |

```
....
1018.        mg_http_parse(buf, strlen(buf), &hm);
```

**Dangerous Functions\Path 44:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=147 |
| Status | New |

The dangerous function, strlen, was found in use at line 835 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1070 | 1070 |
| Object | strlen | strlen |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_http_server(void) { |

```
....
1070.       mg_http_parse(buf, strlen(buf), &hm);
```

### Dangerous Functions\Path 45:

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=148 |
| Status | New |

The dangerous function, strlen, was found in use at line 1320 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1328 | 1328 |
| Object | strlen | strlen |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_http_parse(void) {

```
....
1328.       ASSERT(mg_http_parse(s, strlen(s) - 1, &req) == 0);
```

### Dangerous Functions\Path 46:

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=149 |
| Status | New |

The dangerous function, strlen, was found in use at line 1320 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1329 | 1329 |
| Object | strlen | strlen |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_http_parse(void) {

```
....
1329.      ASSERT(mg_http_parse(s, strlen(s), &req) == (int) strlen(s));
```

**Dangerous Functions\Path 47:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=150 |
| Status | New |

The dangerous function, strlen, was found in use at line 1320 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1329 | 1329 |
| Object | strlen | strlen |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_http_parse(void) {

```
....
1329.      ASSERT(mg_http_parse(s, strlen(s), &req) == (int) strlen(s));
```

**Dangerous Functions\Path 48:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=151 |
| Status | New |

The dangerous function, strlen, was found in use at line 1320 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1330 | 1330 |
| Object | strlen | strlen |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_http_parse(void) {

```
....
1330.      ASSERT(req.message.len == strlen(s));
```

**Dangerous Functions\Path 49:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=152 |
| Status | New |

The dangerous function, strlen, was found in use at line 1320 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1336 | 1336 |
| Object | strlen | strlen |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_http_parse(void) { |

```
....
1336.      size_t idx, len = strlen(s);
```

**Dangerous Functions\Path 50:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=153 |
| Status | New |

The dangerous function, strlen, was found in use at line 1320 in mongoose/unit_test.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1337 | 1337 |
| Object | strlen | strlen |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_http_parse(void) { |

```
....
1337.        ASSERT(mg_http_parse(s, strlen(s), &req) == (int) len);
```

## MemoryFree on StackVariable

Query Path:
CPP\Cx\CPP Medium Threat\MemoryFree on StackVariable Version:0

Description

**MemoryFree on StackVariable\Path 1:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=79 |
| Status | New |

Calling free() (line 835) on a variable that was not dynamically allocated (line 835) in file mongoose/unit_test.c may result with a crash.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 966 | 966 |
| Object | data | data |

| | |
|---|---|
| Code Snippet | |
| File Name | mongoose/unit_test.c |
| Method | static void test_http_server(void) { |

```
....
966.       free(data);
```

**MemoryFree on StackVariable\Path 2:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=80 |
| Status | New |

Calling free() (line 835) on a variable that was not dynamically allocated (line 835) in file mongoose/unit_test.c may result with a crash.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1041 | 1041 |
| Object | p | p |

| | |
|---|---|
| Code Snippet | |
| File Name | mongoose/unit_test.c |

| | |
|---|---|
| Method | static void test_http_server(void) { |

```
....
1041.       free(p);
```

**MemoryFree on StackVariable\Path 3:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=81 |
| Status | New |

Calling free() (line 835) on a variable that was not dynamically allocated (line 835) in file mongoose/unit_test.c may result with a crash.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1056 | 1056 |
| Object | p | p |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_http_server(void) { |

```
....
1056.       free(p);
```

**MemoryFree on StackVariable\Path 4:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=82 |
| Status | New |

Calling free() (line 1664) on a variable that was not dynamically allocated (line 1664) in file mongoose/unit_test.c may result with a crash.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1781 | 1781 |
| Object | p | p |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1781.      free(p);
```

### MemoryFree on StackVariable\Path 5:

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=83 |
| Status | New |

Calling free() (line 1664) on a variable that was not dynamically allocated (line 1664) in file mongoose/unit_test.c may result with a crash.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1785 | 1785 |
| Object | p | p |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1785.      free(p);
```

### MemoryFree on StackVariable\Path 6:

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=84 |
| Status | New |

Calling free() (line 1924) on a variable that was not dynamically allocated (line 1924) in file mongoose/unit_test.c may result with a crash.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1939 | 1939 |
| Object | buf | buf |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_dns_error(const char *dns_server_url, const char *errstr) { |

```
....
1939.     free(buf);
```

### MemoryFree on StackVariable\Path 7:

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=85 |
| Status | New |

Calling free() (line 1993) on a variable that was not dynamically allocated (line 1993) in file mongoose/unit_test.c may result with a crash.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 2002 | 2002 |
| Object | p | p |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_util(void) { |

```
....
2002.     free(p);
```

### MemoryFree on StackVariable\Path 8:

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=86 |
| Status | New |

Calling free() (line 1993) on a variable that was not dynamically allocated (line 1993) in file mongoose/unit_test.c may result with a crash.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 2081 | 2081 |
| Object | s | s |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_util(void) { |

```
....
2081.       free(s);
```

### MemoryFree on StackVariable\Path 9:

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=87](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=87) |
| Status | New |

Calling free() (line 2447) on a variable that was not dynamically allocated (line 2447) in file mongoose/unit_test.c may result with a crash.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 2460 | 2460 |
| Object | data | data |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_packed(void) { |

```
....
2460.     free(data);
```

### MemoryFree on StackVariable\Path 10:

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=88](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=88) |
| Status | New |

Calling free() (line 2447) on a variable that was not dynamically allocated (line 2447) in file mongoose/unit_test.c may result with a crash.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 2466 | 2466 |
| Object | data | data |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_packed(void) { |

```
....
2466.     free(data);
```

### MemoryFree on StackVariable\Path 11:

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=89 |
| Status | New |

Calling free() (line 2669) on a variable that was not dynamically allocated (line 2669) in file mongoose/unit_test.c may result with a crash.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 2758 | 2758 |
| Object | str | str |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_json(void) {

```
....
2758.      free(str);
```

### MemoryFree on StackVariable\Path 12:

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=90 |
| Status | New |

Calling free() (line 2669) on a variable that was not dynamically allocated (line 2669) in file mongoose/unit_test.c may result with a crash.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 2765 | 2765 |
| Object | str | str |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_json(void) {

```
....
2765.       free(str);
```

**MemoryFree on StackVariable\Path 13:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=91](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=91) |
| Status | New |

Calling free() (line 2669) on a variable that was not dynamically allocated (line 2669) in file mongoose/unit_test.c may result with a crash.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 2784 | 2784 |
| Object | str | str |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_json(void) { |

```
....
2784.       free(str);
```

**MemoryFree on StackVariable\Path 14:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=92](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=92) |
| Status | New |

Calling free() (line 2669) on a variable that was not dynamically allocated (line 2669) in file mongoose/unit_test.c may result with a crash.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 2787 | 2787 |
| Object | str | str |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_json(void) { |

```
....
2787.       free(str);
```

## 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=1050063&projectid=50053&pathid=70 |
| Status | New |

The size of the buffer used by test_str in uint32_t, at line 1664 of mongoose/unit_test.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that test_str passes to uint32_t, at line 1664 of mongoose/unit_test.c, to overwrite the target buffer.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1897 | 1897 |
| Object | uint32_t | uint32_t |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1897.       memcpy(a.ip, &addr, sizeof(uint32_t));
```

**Buffer Overflow boundcpy WrongSizeParam\Path 2:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=71 |
| Status | New |

The size of the buffer used by test_str in Namespace253051155, at line 1664 of mongoose/unit_test.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that test_str passes to Namespace253051155, at line 1664 of mongoose/unit_test.c, to overwrite the target buffer.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |

| | | |
|---|---|---|
| Line | 1907 | 1907 |
| Object | Namespace253051155 | Namespace253051155 |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1907.       memset(a.ip, 0, sizeof(a.ip));
```

**Buffer Overflow boundcpy WrongSizeParam\Path 3:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=72 |
| Status | New |

The size of the buffer used by test_util in Namespace253051155, at line 1993 of mongoose/unit_test.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that test_util passes to Namespace253051155, at line 1993 of mongoose/unit_test.c, to overwrite the target buffer.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 2020 | 2020 |
| Object | Namespace253051155 | Namespace253051155 |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_util(void) {

```
....
2020.     memset(a.ip, 0xaa, sizeof(a.ip));
```

**Buffer Overflow boundcpy WrongSizeParam\Path 4:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=73 |
| Status | New |

The size of the buffer used by test_util in Namespace253051155, at line 1993 of mongoose/unit_test.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that test_util passes to Namespace253051155, at line 1993 of mongoose/unit_test.c, to overwrite the target buffer.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |

| | | |
|---|---|---|
| Line | 2028 | 2028 |
| Object | Namespace253051155 | Namespace253051155 |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_util(void) {

```
....
2028.     memset(a.ip, 0xaa, sizeof(a.ip));
```

**Buffer Overflow boundcpy WrongSizeParam\Path 5:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=74](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=74) |
| Status | New |

The size of the buffer used by test_util in Namespace253051155, at line 1993 of mongoose/unit_test.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that test_util passes to Namespace253051155, at line 1993 of mongoose/unit_test.c, to overwrite the target buffer.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 2036 | 2036 |
| Object | Namespace253051155 | Namespace253051155 |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_util(void) {

```
....
2036.     memset(a.ip, 0xaa, sizeof(a.ip));
```

**Buffer Overflow boundcpy WrongSizeParam\Path 6:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=75](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=75) |
| Status | New |

The size of the buffer used by test_util in Namespace253051155, at line 1993 of mongoose/unit_test.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that test_util passes to Namespace253051155, at line 1993 of mongoose/unit_test.c, to overwrite the target buffer.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |

| | | |
|---|---|---|
| Line | 2044 | 2044 |
| Object | Namespace253051155 | Namespace253051155 |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_util(void) {

```
....
2044.    memset(a.ip, 0xaa, sizeof(a.ip));
```

**Buffer Overflow boundcpy WrongSizeParam\Path 7:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=76 |
| Status | New |

The size of the buffer used by test_util in Namespace253051155, at line 1993 of mongoose/unit_test.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that test_util passes to Namespace253051155, at line 1993 of mongoose/unit_test.c, to overwrite the target buffer.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 2052 | 2052 |
| Object | Namespace253051155 | Namespace253051155 |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_util(void) {

```
....
2052.    memset(a.ip, 0xaa, sizeof(a.ip));
```

**Buffer Overflow boundcpy WrongSizeParam\Path 8:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=77 |
| Status | New |

The size of the buffer used by mg_tcpip_driver_stm32_tx in len, at line 169 of mongoose/driver_stm32.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that mg_tcpip_driver_stm32_tx passes to len, at line 169 of mongoose/driver_stm32.c, to overwrite the target buffer.

| | Source | Destination |
|---|---|---|
| File | mongoose/driver_stm32.c | mongoose/driver_stm32.c |

| | | |
|---|---|---|
| Line | 180 | 180 |
| Object | len | len |

Code Snippet
File Name mongoose/driver_stm32.c
Method static size_t mg_tcpip_driver_stm32_tx(const void *buf, size_t len,

```
....
180.        memcpy(s_txbuf[s_txno], buf, len);      // Copy data
```

**Buffer Overflow boundcpy WrongSizeParam\Path 9:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=78 |
| Status | New |

The size of the buffer used by producer in len, at line 2940 of mongoose/unit_test.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that producer passes to len, at line 2940 of mongoose/unit_test.c, to overwrite the target buffer.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 2949 | 2949 |
| Object | len | len |

Code Snippet
File Name mongoose/unit_test.c
Method static void producer(void *param) {

```
....
2949.       memcpy(buf, &tmp[ofs], len);
```

## Inadequate Encryption Strength

Query Path:
CPP\Cx\CPP Medium Threat\Inadequate Encryption Strength Version:1

### Categories

FISMA 2014: Configuration Management
NIST SP 800-53: SC-13 Cryptographic Protection (P1)
OWASP Top 10 2017: A3-Sensitive Data Exposure

*Description*

**Inadequate Encryption Strength\Path 1:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=192 |
| Status | New |

The application uses a weak cryptographic algorithm, lwip_md5_update at line 1311 of mongoose/eap.c, to protect sensitive personal information secret_len, from mongoose/eap.c at line 1311.

| | Source | Destination |
|---|---|---|
| File | mongoose/eap.c | mongoose/eap.c |
| Line | 1450 | 1450 |
| Object | secret_len | lwip_md5_update |

Code Snippet
File Name mongoose/eap.c
Method static void eap_request(ppp_pcb *pcb, u_char *inp, int id, int len) {

```
....
1450.           lwip_md5_update(&mdContext, (u_char *)secret,
secret_len);
```

### Inadequate Encryption Strength\Path 2:

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=193 |
| Status | New |

The application uses a weak cryptographic algorithm, lwip_md5_update at line 1311 of mongoose/eap.c, to protect sensitive personal information secret, from mongoose/eap.c at line 1311.

| | Source | Destination |
|---|---|---|
| File | mongoose/eap.c | mongoose/eap.c |
| Line | 1450 | 1450 |
| Object | secret | lwip_md5_update |

Code Snippet
File Name mongoose/eap.c
Method static void eap_request(ppp_pcb *pcb, u_char *inp, int id, int len) {

```
....
1450.           lwip_md5_update(&mdContext, (u_char *)secret,
secret_len);
```

### Inadequate Encryption Strength\Path 3:

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=194 |
| Status | New |

The application uses a weak cryptographic algorithm, lwip_md5_update at line 1725 of mongoose/eap.c, to protect sensitive personal information secret_len, from mongoose/eap.c at line 1725.

| | Source | Destination |
|---|---|---|
| File | mongoose/eap.c | mongoose/eap.c |
| Line | 1877 | 1877 |
| Object | secret_len | lwip_md5_update |

Code Snippet
File Name mongoose/eap.c
Method static void eap_response(ppp_pcb *pcb, u_char *inp, int id, int len) {

```
....
1877.           lwip_md5_update(&mdContext, (u_char *)secret,
secret_len);
```

**Inadequate Encryption Strength\Path 4:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=195 |
| Status | New |

The application uses a weak cryptographic algorithm, lwip_md5_update at line 1725 of mongoose/eap.c, to protect sensitive personal information secret, from mongoose/eap.c at line 1725.

| | Source | Destination |
|---|---|---|
| File | mongoose/eap.c | mongoose/eap.c |
| Line | 1877 | 1877 |
| Object | secret | lwip_md5_update |

Code Snippet
File Name mongoose/eap.c
Method static void eap_response(ppp_pcb *pcb, u_char *inp, int id, int len) {

```
....
1877.           lwip_md5_update(&mdContext, (u_char *)secret,
secret_len);
```

**Inadequate Encryption Strength\Path 5:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=196 |
| Status | New |

The application uses a weak cryptographic algorithm, SHA1Update at line 315 of mongoose/eap.c, to protect sensitive personal information pn_secret, from mongoose/eap.c at line 315.

| | Source | Destination |
|---|---|---|
| File | mongoose/eap.c | mongoose/eap.c |

| | | |
|---|---|---|
| Line | 328 | 328 |
| Object | pn_secret | SHA1Update |

Code Snippet
File Name mongoose/eap.c
Method pncrypt_setkey(int timeoffs)

```
....
328.        SHA1Update(&ctxt, pn_secret, strlen(pn_secret));
```

**Inadequate Encryption Strength\Path 6:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=197 |
| Status | New |

The application uses a weak cryptographic algorithm, SHA1Update at line 315 of mongoose/eap.c, to protect sensitive personal information pn_secret, from mongoose/eap.c at line 315.

| | Source | Destination |
|---|---|---|
| File | mongoose/eap.c | mongoose/eap.c |
| Line | 328 | 328 |
| Object | pn_secret | SHA1Update |

Code Snippet
File Name mongoose/eap.c
Method pncrypt_setkey(int timeoffs)

```
....
328.        SHA1Update(&ctxt, pn_secret, strlen(pn_secret));
```

## 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=1050063&projectid=50053&pathid=189 |
| Status | New |

The variable declared in pvReturn at mongoose/heap_4.c in line 113 is not initialized when it is used by pvReturn at mongoose/heap_4.c in line 113.

| | Source | Destination |
|---|---|---|
| File | mongoose/heap_4.c | mongoose/heap_4.c |
| Line | 116 | 258 |
| Object | pvReturn | pvReturn |

Code Snippet
File Name mongoose/heap_4.c
Method void *pvPortMalloc( size_t xWantedSize )

```
....
116.  void *pvReturn = NULL;
....
258.        configASSERT( ( ( ( size_t ) pvReturn ) & ( size_t )
portBYTE_ALIGNMENT_MASK ) == 0 );
```

**Use of Zero Initialized Pointer\Path 2:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=190 |
| Status | New |

The variable declared in pxNextFreeBlock at mongoose/heap_4.c in line 329 is not initialized when it is used by pxNextFreeBlock at mongoose/heap_4.c in line 329.

| | Source | Destination |
|---|---|---|
| File | mongoose/heap_4.c | mongoose/heap_4.c |
| Line | 360 | 366 |
| Object | pxNextFreeBlock | pxNextFreeBlock |

Code Snippet
File Name mongoose/heap_4.c
Method static void prvHeapInit( void )

```
....
360.        pxEnd->pxNextFreeBlock = NULL;
....
366.        pxFirstFreeBlock->pxNextFreeBlock = pxEnd;
```

**Use of Zero Initialized Pointer\Path 3:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=191 |
| Status | New |

The variable declared in head at mongoose/unit_test.c in line 2845 is not initialized when it is used by head at mongoose/unit_test.c in line 2845.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 2846 | 2886 |
| Object | head | head |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_rpc(void) {

```
....
2846.    struct mg_rpc *head = NULL;
....
2886.      req.head = &head;
```

## 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=1050063&projectid=50053&pathid=68 |
| Status | New |

The pointer result at mongoose/net.c in line 96 is being used after it has been freed.

| | Source | Destination |
|---|---|---|
| File | mongoose/net.c | mongoose/net.c |
| Line | 120 | 120 |
| Object | result | result |

Code Snippet
File Name mongoose/net.c
Method static struct user *authenticate(struct mg_http_message *hm) {

```
....
120.    return result;
```

**Buffer Overflow AddressOfLocalVarReturned\Path 2:**

| | |
|---|---|
| Severity | Medium |
| Result State | To Verify |

| | |
|---|---|
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=69 |
| Status | New |

The pointer ahbptab at mongoose/driver_stm32.c in line 65 is being used after it has been freed.

| | Source | Destination |
|---|---|---|
| File | mongoose/driver_stm32.c | mongoose/driver_stm32.c |
| Line | 88 | 88 |
| Object | ahbptab | ahbptab |

Code Snippet
File Name mongoose/driver_stm32.c
Method static uint32_t get_hclk(void) {

```
....
88.    return ((uint32_t) clk) >> ahbptab[hpre - 8];
```

## 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=1050063&projectid=50053&pathid=93 |
| Status | New |

The function pl in mongoose/eap.c at line 1197 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 | mongoose/eap.c | mongoose/eap.c |
| Line | 1211 | 1211 |
| Object | pl | pl |

Code Snippet
File Name mongoose/eap.c
Method name_of_pn_file()

```
....
1211.        path = malloc(pl);
```

## 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=1050063&projectid=50053&pathid=11 |
| Status | New |

The mqtt_cb method calls the snprintf function, at line 356 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 379 | 379 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void mqtt_cb(struct mg_connection *c, int ev, void *evd, void *fnd) {

```
....
379.       snprintf(test_data->topic, test_data->topicsize, "%.*s",
```

**Unchecked Return Value\Path 2:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=12 |
| Status | New |

The mqtt_cb method calls the snprintf function, at line 356 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 381 | 381 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void mqtt_cb(struct mg_connection *c, int ev, void *evd, void *fnd) {

```
....
381.        snprintf(buf + 1, test_data->msgsize - 2, "%.*s", (int) mm-
>data.len,
```

### Unchecked Return Value\Path 3:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=13 |
| Status | New |

The fcb method calls the snprintf function, at line 687 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 691 | 691 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void fcb(struct mg_connection *c, int ev, void *ev_data, void *fn_data) {

```
....
691.        snprintf(fd->buf, FETCH_BUF_SIZE, "%.*s", (int) hm-
>message.len,
```

### Unchecked Return Value\Path 4:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=14 |
| Status | New |

The test_http_server method calls the remove function, at line 835 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1025 | 1025 |
| Object | remove | remove |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_http_server(void) {

```
....
1025.       remove("uploaded.txt");
```

**Unchecked Return Value\Path 5:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=15 |
| Status | New |

The test_http_server method calls the remove function, at line 835 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1042 | 1042 |
| Object | remove | remove |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_http_server(void) {

```
....
1042.       remove("uploaded.txt");
```

**Unchecked Return Value\Path 6:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=16 |
| Status | New |

The test_http_server method calls the remove function, at line 835 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1048 | 1048 |
| Object | remove | remove |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_http_server(void) {

```
....
1048.       remove("uploaded.txt");
```

### Unchecked Return Value\Path 7:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=17 |
| Status | New |

The test_http_server method calls the remove function, at line 835 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1057 | 1057 |
| Object | remove | remove |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_http_server(void) {

```
....
1057.       remove("uploaded.txt");
```

### Unchecked Return Value\Path 8:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=18 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1813 | 1813 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1813.       TESTDOUBLE("%g", 0.0, "0");
```

### Unchecked Return Value\Path 9:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=19 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1814 | 1814 |
| Object | snprintf | snprintf |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1814.       TESTDOUBLE("%g", 0.123, "0.123");
```

### Unchecked Return Value\Path 10:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=20 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1815 | 1815 |
| Object | snprintf | snprintf |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1815.       TESTDOUBLE("%g", 0.00123, "0.00123");
```

### Unchecked Return Value\Path 11:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=21 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1816 | 1816 |
| Object | snprintf | snprintf |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1816.       TESTDOUBLE("%g", 0.123456333, "0.123456");
```

### Unchecked Return Value\Path 12:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=22 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1817 | 1817 |
| Object | snprintf | snprintf |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1817.       TESTDOUBLE("%g", 123.0, "123");
```

### Unchecked Return Value\Path 13:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=23 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1818 | 1818 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1818.       TESTDOUBLE("%g", 11.5454, "11.5454");
```

### Unchecked Return Value\Path 14:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=24 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1819 | 1819 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1819.       TESTDOUBLE("%g", 11.0001, "11.0001");
```

### Unchecked Return Value\Path 15:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=25](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=25) |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1820 | 1820 |
| Object | snprintf | snprintf |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1820.       TESTDOUBLE("%g", 0.999, "0.999");
```

### Unchecked Return Value\Path 16:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=26](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=26) |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1821 | 1821 |
| Object | snprintf | snprintf |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1821.       TESTDOUBLE("%g", 0.999999, "0.999999");
```

### Unchecked Return Value\Path 17:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=27 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1822 | 1822 |
| Object | snprintf | snprintf |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1822.       TESTDOUBLE("%g", 0.9999999, "1");
```

### Unchecked Return Value\Path 18:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=28 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1823 | 1823 |
| Object | snprintf | snprintf |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1823.       TESTDOUBLE("%g", 10.9, "10.9");
```

### Unchecked Return Value\Path 19:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=29 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1824 | 1824 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1824.       TESTDOUBLE("%g", 10.01, "10.01");
```

### Unchecked Return Value\Path 20:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=30 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1825 | 1825 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1825.       TESTDOUBLE("%g", 1.0, "1");
```

### Unchecked Return Value\Path 21:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=31 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1826 | 1826 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1826.       TESTDOUBLE("%g", 10.0, "10");
```

### Unchecked Return Value\Path 22:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=32 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1827 | 1827 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1827.       TESTDOUBLE("%g", 100.0, "100");
```

### Unchecked Return Value\Path 23:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=33 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1828 | 1828 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1828.       TESTDOUBLE("%g", 1000.0, "1000");
```

### Unchecked Return Value\Path 24:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=34 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1829 | 1829 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1829.       TESTDOUBLE("%g", 10000.0, "10000");
```

### Unchecked Return Value\Path 25:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=35 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1830 | 1830 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1830.       TESTDOUBLE("%g", 100000.0, "100000");
```

### Unchecked Return Value\Path 26:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=36 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1831 | 1831 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1831.       TESTDOUBLE("%g", 1000000.0, "1e+06");
```

**Unchecked Return Value\Path 27:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=37 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1832 | 1832 |
| Object | snprintf | snprintf |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1832.       TESTDOUBLE("%g", 10000000.0, "1e+07");
```

**Unchecked Return Value\Path 28:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=38 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1833 | 1833 |
| Object | snprintf | snprintf |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1833.       TESTDOUBLE("%g", 100000001.0, "1e+08");
```

## Unchecked Return Value\Path 29:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=39](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=39) |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1834 | 1834 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1834.       TESTDOUBLE("%g", 10.5454, "10.5454");
```

## Unchecked Return Value\Path 30:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=40](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=40) |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1835 | 1835 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1835.       TESTDOUBLE("%g", 999999.0, "999999");
```

### Unchecked Return Value\Path 31:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=41 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1836 | 1836 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1836.       TESTDOUBLE("%g", 9999999.0, "1e+07");
```

### Unchecked Return Value\Path 32:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=42 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1837 | 1837 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1837.       TESTDOUBLE("%g", 44556677.0, "4.45567e+07");
```

### Unchecked Return Value\Path 33:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=43](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=43) |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1838 | 1838 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1838.       TESTDOUBLE("%g", 1234567.2, "1.23457e+06");
```

### Unchecked Return Value\Path 34:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=44](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=44) |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1839 | 1839 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1839.       TESTDOUBLE("%g", -987.65432, "-987.654");
```

**Unchecked Return Value\Path 35:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=45 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1840 | 1840 |
| Object | snprintf | snprintf |

| | |
|---|---|
| Code Snippet | |
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1840.       TESTDOUBLE("%g", 0.0000000001, "1e-10");
```

**Unchecked Return Value\Path 36:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=46 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1841 | 1841 |
| Object | snprintf | snprintf |

| | |
|---|---|
| Code Snippet | |
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1841.       TESTDOUBLE("%g", 2.34567e-57, "2.34567e-57");
```

**Unchecked Return Value\Path 37:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=47 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1842 | 1842 |
| Object | snprintf | snprintf |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1842.       TESTDOUBLE("%.*g", DBLWIDTH(7, 9999999.0), "9999999");
```

**Unchecked Return Value\Path 38:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=48 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1843 | 1843 |
| Object | snprintf | snprintf |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1843.        TESTDOUBLE("%.*g", DBLWIDTH(10, 0.123456333), "0.123456333");
```

### Unchecked Return Value\Path 39:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=49 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1844 | 1844 |
| Object | snprintf | snprintf |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1844.        TESTDOUBLE("%g", 123.456222, "123.456");
```

### Unchecked Return Value\Path 40:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=50 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1845 | 1845 |
| Object | snprintf | snprintf |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1845.       TESTDOUBLE("%.*g", DBLWIDTH(10, 123.456222), "123.456222");
```

**Unchecked Return Value\Path 41:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=51 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1846 | 1846 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1846.       TESTDOUBLE("%g", 600.1234, "600.123");
```

**Unchecked Return Value\Path 42:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=52 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1847 | 1847 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1847.       TESTDOUBLE("%g", -600.1234, "-600.123");
```

**Unchecked Return Value\Path 43:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=53 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1848 | 1848 |
| Object | snprintf | snprintf |

Code Snippet
File Name    mongoose/unit_test.c
Method    static void test_str(void) {

```
....
1848.       TESTDOUBLE("%g", 599.1234, "599.123");
```

**Unchecked Return Value\Path 44:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=54 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1849 | 1849 |
| Object | snprintf | snprintf |

Code Snippet
File Name    mongoose/unit_test.c
Method    static void test_str(void) {

```
....
1849.      TESTDOUBLE("%g", -599.1234, "-599.123");
```

**Unchecked Return Value\Path 45:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=55](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=55) |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1850 | 1850 |
| Object | snprintf | snprintf |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1850.      TESTDOUBLE("%g", 0.14, "0.14");
```

**Unchecked Return Value\Path 46:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=56](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=56) |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1851 | 1851 |
| Object | snprintf | snprintf |

Code Snippet

| | |
|---|---|
| File Name | mongoose/unit_test.c |
| Method | static void test_str(void) { |

```
....
1851.       TESTDOUBLE("%f", 0.14, "0.140000");
```

### Unchecked Return Value\Path 47:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=57 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1852 | 1852 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1852.       TESTDOUBLE("%.*f", DBLWIDTH(4, 0.14), "0.1400");
```

### Unchecked Return Value\Path 48:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=58 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1853 | 1853 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1853.       TESTDOUBLE("%.*f", DBLWIDTH(3, 0.14), "0.140");
```

### Unchecked Return Value\Path 49:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=59 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1854 | 1854 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1854.       TESTDOUBLE("%.*f", DBLWIDTH(2, 0.14), "0.14");
```

### Unchecked Return Value\Path 50:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=60 |
| Status | New |

The test_str method calls the snprintf function, at line 1664 of mongoose/unit_test.c. However, the code does not check the return value from this function, and thus would not detect runtime errors or other unexpected states.

| | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 1855 | 1855 |
| Object | snprintf | snprintf |

Code Snippet
File Name mongoose/unit_test.c
Method static void test_str(void) {

```
....
1855.      TESTDOUBLE("%.*f", DBLWIDTH(1, 0.14), "0.1");
```

## 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=1050063&projectid=50053&pathid=94 |
| Status | New |

The variable declared in null at mongoose/queue.c in line 2852 is not initialized when it is used by u at mongoose/queue.c in line 2149.

| | Source | Destination |
|---|---|---|
| File | mongoose/queue.c | mongoose/queue.c |
| Line | 2854 | 2162 |
| Object | null | u |

Code Snippet
File Name mongoose/queue.c
Method QueueSetMemberHandle_t xQueueSelectFromSet( QueueSetHandle_t xQueueSet, TickType_t const xTicksToWait )

```
....
2854.       QueueSetMemberHandle_t xReturn = NULL;
```

File Name mongoose/queue.c

Method static void prvCopyDataFromQueue( Queue_t * const pxQueue, void * const pvBuffer )

```
....
2162.               ( void ) memcpy( ( void * ) pvBuffer, ( void * )
pxQueue->u.xQueue.pcReadFrom, ( size_t ) pxQueue->uxItemSize ); /*lint
!e961 !e418 !e9087 MISRA exception as the casts are only redundant for
some ports.  Also previous logic ensures a null pointer can only be
passed to memcpy() when the count is 0.  Cast to void required by
function signature and safe as no alignment requirement and copy length
specified in bytes. */
```

### NULL Pointer Dereference\Path 2:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=95](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=95) |
| Status | New |

The variable declared in null at mongoose/queue.c in line 2865 is not initialized when it is used by u at mongoose/queue.c in line 2149.

| | Source | Destination |
|---|---|---|
| File | mongoose/queue.c | mongoose/queue.c |
| Line | 2867 | 2162 |
| Object | null | u |

Code Snippet

File Name mongoose/queue.c

Method QueueSetMemberHandle_t xQueueSelectFromSetFromISR( QueueSetHandle_t xQueueSet )

```
....
2867.       QueueSetMemberHandle_t xReturn = NULL;
```

File Name mongoose/queue.c

Method static void prvCopyDataFromQueue( Queue_t * const pxQueue, void * const pvBuffer )

```
....
2162.               ( void ) memcpy( ( void * ) pvBuffer, ( void * )
pxQueue->u.xQueue.pcReadFrom, ( size_t ) pxQueue->uxItemSize ); /*lint
!e961 !e418 !e9087 MISRA exception as the casts are only redundant for
some ports.  Also previous logic ensures a null pointer can only be
passed to memcpy() when the count is 0.  Cast to void required by
function signature and safe as no alignment requirement and copy length
specified in bytes. */
```

### NULL Pointer Dereference\Path 3:

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=96](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=96) |
| Status | New |

The variable declared in null at mongoose/queue.c in line 2852 is not initialized when it is used by u at mongoose/queue.c in line 2149.

| | Source | Destination |
|---|---|---|
| File | mongoose/queue.c | mongoose/queue.c |

| | | |
|---|---|---|
| Line | 2854 | 2154 |
| Object | null | u |

Code Snippet
File Name mongoose/queue.c
Method QueueSetMemberHandle_t xQueueSelectFromSet( QueueSetHandle_t xQueueSet, TickType_t const xTicksToWait )

```
....
2854.       QueueSetMemberHandle_t xReturn = NULL;
```

▼

File Name mongoose/queue.c

Method static void prvCopyDataFromQueue( Queue_t * const pxQueue, void * const pvBuffer )

```
....
2154.               if( pxQueue->u.xQueue.pcReadFrom >= pxQueue->u.xQueue.pcTail ) /*lint !e946 MISRA exception justified as use of the relational operator is the cleanest solutions. */
```

**NULL Pointer Dereference\Path 4:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=97 |
| Status | New |

The variable declared in null at mongoose/queue.c in line 2865 is not initialized when it is used by u at mongoose/queue.c in line 2149.

| | Source | Destination |
|---|---|---|
| File | mongoose/queue.c | mongoose/queue.c |
| Line | 2867 | 2154 |
| Object | null | u |

Code Snippet
File Name mongoose/queue.c
Method QueueSetMemberHandle_t xQueueSelectFromSetFromISR( QueueSetHandle_t xQueueSet )

```
....
2867.       QueueSetMemberHandle_t xReturn = NULL;
```

▼

File Name mongoose/queue.c

Method static void prvCopyDataFromQueue( Queue_t * const pxQueue, void * const pvBuffer )

```
....
2154.              if( pxQueue->u.xQueue.pcReadFrom >= pxQueue-
>u.xQueue.pcTail ) /*lint !e946 MISRA exception justified as use of the
relational operator is the cleanest solutions. */
```

**NULL Pointer Dereference\Path 5:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=98 |
| Status | New |

The variable declared in null at mongoose/queue.c in line 2852 is not initialized when it is used by u at mongoose/queue.c in line 2149.

| | Source | Destination |
|---|---|---|
| File | mongoose/queue.c | mongoose/queue.c |
| Line | 2854 | 2154 |
| Object | null | u |

Code Snippet

| | |
|---|---|
| File Name | mongoose/queue.c |
| Method | QueueSetMemberHandle_t xQueueSelectFromSet( QueueSetHandle_t xQueueSet, TickType_t const xTicksToWait ) |

```
....
2854.        QueueSetMemberHandle_t xReturn = NULL;
```

| | |
|---|---|
| File Name | mongoose/queue.c |
| Method | static void prvCopyDataFromQueue( Queue_t * const pxQueue, void * const pvBuffer ) |

```
....
2154.              if( pxQueue->u.xQueue.pcReadFrom >= pxQueue-
>u.xQueue.pcTail ) /*lint !e946 MISRA exception justified as use of the
relational operator is the cleanest solutions. */
```

**NULL Pointer Dereference\Path 6:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=99 |
| Status | New |

The variable declared in null at mongoose/queue.c in line 2865 is not initialized when it is used by u at mongoose/queue.c in line 2149.

| | Source | Destination |
|---|---|---|
| File | mongoose/queue.c | mongoose/queue.c |
| Line | 2867 | 2154 |
| Object | null | u |

Code Snippet

File Name mongoose/queue.c

Method QueueSetMemberHandle_t xQueueSelectFromSetFromISR( QueueSetHandle_t xQueueSet )

```
....
2867.       QueueSetMemberHandle_t xReturn = NULL;
```

▼

File Name mongoose/queue.c

Method static void prvCopyDataFromQueue( Queue_t * const pxQueue, void * const pvBuffer )

```
....
2154.              if( pxQueue->u.xQueue.pcReadFrom >= pxQueue-
>u.xQueue.pcTail ) /*lint !e946 MISRA exception justified as use of the
relational operator is the cleanest solutions. */
```

**NULL Pointer Dereference\Path 7:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | [http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=100](http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=100) |
| Status | New |

The variable declared in null at mongoose/queue.c in line 2852 is not initialized when it is used by pxQueue at mongoose/queue.c in line 2287.

| | Source | Destination |
|---|---|---|
| File | mongoose/queue.c | mongoose/queue.c |
| Line | 2854 | 2293 |
| Object | null | pxQueue |

Code Snippet

File Name mongoose/queue.c

Method QueueSetMemberHandle_t xQueueSelectFromSet( QueueSetHandle_t xQueueSet, TickType_t const xTicksToWait )

```
....
2854.       QueueSetMemberHandle_t xReturn = NULL;
```

▼

| | |
|---|---|
| File Name | mongoose/queue.c |
| Method | static BaseType_t prvIsQueueEmpty( const Queue_t *pxQueue ) |

```
....
2293.            if( pxQueue->uxMessagesWaiting == ( UBaseType_t )  0 )
```

**NULL Pointer Dereference\Path 8:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=101 |
| Status | New |

The variable declared in 0 at mongoose/heap_4.c in line 329 is not initialized when it is used by xStart at mongoose/heap_4.c in line 329.

| | Source | Destination |
|---|---|---|
| File | mongoose/heap_4.c | mongoose/heap_4.c |
| Line | 351 | 351 |
| Object | 0 | xStart |

Code Snippet

| | |
|---|---|
| File Name | mongoose/heap_4.c |
| Method | static void prvHeapInit( void ) |

```
....
351.        xStart.xBlockSize = ( size_t ) 0;
```

**NULL Pointer Dereference\Path 9:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=102 |
| Status | New |

The variable declared in 0 at mongoose/queue.c in line 255 is not initialized when it is used by pxQueue at mongoose/queue.c in line 255.

| | Source | Destination |
|---|---|---|
| File | mongoose/queue.c | mongoose/queue.c |
| Line | 264 | 264 |
| Object | 0 | pxQueue |

Code Snippet

| | |
|---|---|
| File Name | mongoose/queue.c |

|  |  |
|---|---|
| Method | BaseType_t xQueueGenericReset( QueueHandle_t xQueue, BaseType_t xNewQueue ) |

```
....
264.            pxQueue->uxMessagesWaiting = ( UBaseType_t ) 0U;
```

## 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=1050063&projectid=50053&pathid=103 |
| Status | New |

|  | Source | Destination |
|---|---|---|
| File | mongoose/unit_test.c | mongoose/unit_test.c |
| Line | 2586 | 2586 |
| Object | ofs | ofs |

|  |  |
|---|---|
| Code Snippet |  |
| File Name | mongoose/unit_test.c |
| Method | static void w2(struct mg_connection *c, int ev, void *ev_data, void *fn_data) { |

```
....
2586.        if (n < msg.len - 1) c->send.buf[ofs] = op;  // Clear FIN
flag
```

## 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=1050063&projectid=50053&pathid=198 |
| Status | New |

| | Source | Destination |
|---|---|---|
| File | mongoose/eap.c | mongoose/eap.c |
| Line | 1366 | 1366 |
| Object | BinaryExpr | BinaryExpr |

Code Snippet
File Name mongoose/eap.c
Method static void eap_request(ppp_pcb *pcb, u_char *inp, int id, int len) {

```
....
1366.                           len = read(fd, rhostname + SRP_PSEUDO_LEN,
```

## TOCTOU

Query Path:
CPP\Cx\CPP Low Visibility\TOCTOU Version:1
*Description*

**TOCTOU\Path 1:**

| | |
|---|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050063&projectid=50053&pathid=199 |
| Status | New |

The open_pn_file method in mongoose/eap.c file utilizes open that is accessed by other concurrent functionality in a way that is not thread-safe, which may result in a Race Condition over this resource.

| | Source | Destination |
|---|---|---|
| File | mongoose/eap.c | mongoose/eap.c |
| Line | 1231 | 1231 |
| Object | open | open |

Code Snippet
File Name mongoose/eap.c
Method open_pn_file(modebits)

```
....
1231.       fd = open(path, modebits, S_IRUSR | S_IWUSR);
```

# Buffer Overflow OutOfBound

## Risk

### What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

## Cause

### How does it happen

Buffer Overflows can manifest in numerous different variations. In it's most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.
Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

## General Recommendations

### How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
- Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
- Consistently apply tests for the size of buffers.
- Do not return variable addresses outside the scope of their variables.

## Source Code Examples

### CPP

### Overflowing Buffers

```
const int BUFFER_SIZE = 10;
char buffer[BUFFER_SIZE];

void copyStringToBuffer(char* inputString)
{
    strcpy(buffer, inputString);
}
```

### Checked Buffers

```
const int BUFFER_SIZE = 10;
const int MAX_INPUT_SIZE = 256;
char buffer[BUFFER_SIZE];

void copyStringToBuffer(char* inputString)
{
    if (strnlen(inputString, MAX_INPUT_SIZE) < sizeof(buffer))
    {
        strncpy(buffer, inputString, sizeof(buffer));
    }
}
```

# Buffer Overflow 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

# 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);
```

# 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

```c
int main(int argc, char* argv[])
{
      printf("%s", argv[1]); // Second parameter is not a formattable string
      return 0;
}
```

# Use of Zero Initialized Pointer

## Risk

### What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

## Cause

### How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

## General Recommendations

### How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
- Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
- Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.

## Source Code Examples

### CPP

### Explicit NULL Dereference

```
char * input = NULL;
printf("%s", input);
```

### Implicit NULL Dereference

```
char * input;
printf("%s", input);
```

### Java

### Explicit Null Dereference

```
Object o = null;
out.println(o.getClass());
```

# Inadequate Encryption Strength

## Risk

### What might happen

Using weak or outdated cryptography does not provide sufficient protection for sensitive data. An attacker that gains access to the encrypted data would likely be able to break the encryption, using either cryptanalysis or brute force attacks. Thus, the attacker would be able to steal user passwords and other personal data. This could lead to user impersonation or identity theft.

## Cause

### How does it happen

The application uses a weak algorithm, that is considered obselete since it is relatively easy to break. These obselete algorithms are vulnerable to several different kinds of attacks, including brute force.

## General Recommendations

### How to avoid it

Generic Guidance:

- Always use strong, modern algorithms for encryption, hashing, and so on.
- Do not use weak, outdated, or obsolete algorithms.
- Ensure you select the correct cryptographic mechanism according to the specific requirements.
- Passwords should be protected with a dedicated password protection scheme, such as bcrypt, scrypt, PBKDF2, or Argon2.

Specific Recommendations:

- Do not use SHA-1, MD5, or any other weak hash algorithm to protect passwords or personal data. Instead, use a stronger hash such as SHA-256 when a secure hash is required.
- Do not use DES, Triple-DES, RC2, or any other weak encryption algorithm to protect passwords or personal data. Instead, use a stronger encryption algorithm such as AES to protect personal data.
- Do not use weak encryption modes such as ECB, or rely on insecure defaults. Explicitly specify a stronger encryption mode, such as GCM.
- For symmetric encryption, use a key length of at least 256 bits.

## Source Code Examples

### Java

### Weakly Hashed PII

```
string protectSSN(HttpServletRequest req) {
    string socialSecurityNum = req.getParameter("SocialSecurityNo");

    return DigestUtils.md5Hex(socialSecurityNum);
}
```

### Stronger Hash for PII

```
string protectSSN(HttpServletRequest req) {
    string socialSecurityNum = req.getParameter("SocialSecurityNo");

    return DigestUtils.sha256Hex(socialSecurityNum);
}
```

# 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';
```

# 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

## 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<br>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<br>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<br>Availability<br>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**<br>**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**

```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**<br>**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**<br>Research Concepts1000 |
| ParentOf | Weakness Class | 638 | Failure to Use Complete Mediation | Research Concepts1000 |
| ParentOf | Weakness Base | 804 | Guessable CAPTCHA | **Development Concepts (primary)699**<br>**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

# TOCTOU

## Risk

### What might happen

At best, a Race Condition may cause errors in accuracy, overidden values or unexpected behavior that may result in denial-of-service. At worst, it may allow attackers to retrieve data or bypass security processes by replaying a controllable Race Condition until it plays out in their favor.

## Cause

### How does it happen

Race Conditions occur when a public, single instance of a resource is used by multiple concurrent logical processes. If the these logical processes attempt to retrieve and update the resource without a timely management system, such as a lock, a Race Condition will occur.

An example for when a Race Condition occurs is a resource that may return a certain value to a process for further editing, and then updated by a second process, resulting in the original process' data no longer being valid. Once the original process edits and updates the incorrect value back into the resource, the second process' update has been overwritten and lost.

## General Recommendations

### How to avoid it

When sharing resources between concurrent processes across the application ensure that these resources are either thread-safe, or implement a locking mechanism to ensure expected concurrent activity.

## Source Code Examples

### Java

### Different Threads Increment and Decrement The Same Counter Repeatedly, Resulting in a Race Condition

```
     public static int counter = 0;
     public static void start() throws InterruptedException {
            incrementCounter ic;
            decrementCounter dc;
            while(counter == 0) {
                   counter = 0;
                   ic = new incrementCounter();
                   dc = new decrementCounter();
                   ic.start();
                   dc.start();
                   ic.join();
                   dc.join();
            }
            System.out.println(counter); //Will stop and return either -1 or 1 due to race
condition over counter
      }

     public static class incrementCounter extends Thread {
         public void run() {
            counter++;
         }
```

```
        }

        public static class decrementCounter extends Thread {
            public void run() {
               counter--;
            }
        }
```

### Different Threads Increment and Decrement The Same Thread-Safe Counter Repeatedly, Never Resulting in a Race Condition

```
        public static int counter = 0;
        public static Object lock = new Object();

        public static void start() throws InterruptedException {
               incrementCounter ic;
               decrementCounter dc;
               while(counter == 0) { // because of proper locking, this condition is never false
                      counter = 0;
                      ic = new incrementCounter();
                      dc = new decrementCounter();
                      ic.start();
                      dc.start();
                      ic.join();
                      dc.join();
               }
               System.out.println(counter); // Never reached
        }

        public static class incrementCounter extends Thread {
            public void run() {
               synchronized (lock) {
                      counter++;
               }
            }
        }

        public static class decrementCounter extends Thread {
            public void run() {
               synchronized (lock) {
                      counter--;
               }
            }
        }
```

## Scanned Languages

| Language | Hash Number | Change Date |
|---|---|---|
| CPP | 4541647240435660 | 6/19/2024 |
| Common | 0105849645654507 | 6/19/2024 |