

TencentOS-tiny Scan Report

| | |
|-----------------------|---|
| Project Name | TencentOS-tiny |
| Scan Start | Friday, June 21, 2024 10:30:44 PM |
| Preset | Checkmarx Default |
| Scan Time | 00h:04m:03s |
| Lines Of Code Scanned | 27238 |
| Files Scanned | 28 |
| Report Creation Time | Friday, June 21, 2024 10:35:23 PM |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040 |
| Team | CxServer |
| Checkmarx Version | 8.7.0 |
| Scan Type | Full |
| Source Origin | LocalPath |
| Density | 8/1000 (Vulnerabilities/LOC) |
| Visibility | Public |

Filter Settings

Severity

Included: High, Medium, Low, Information

Excluded: None

Result State

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

Excluded: None

Assigned to

Included: All

Categories

Included:

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

Excluded:

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

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

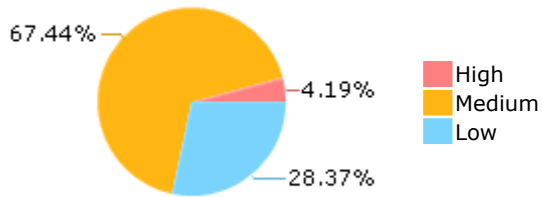
Results Limit

Results limit per query was set to 50

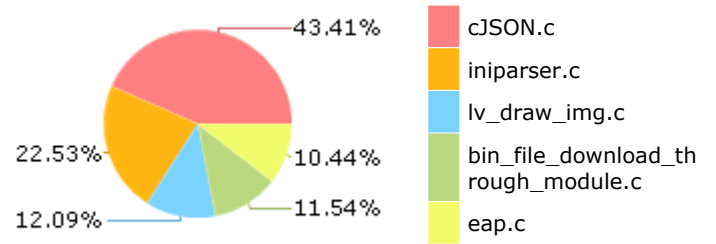
Selected Queries

Selected queries are listed in [Result Summary](#)

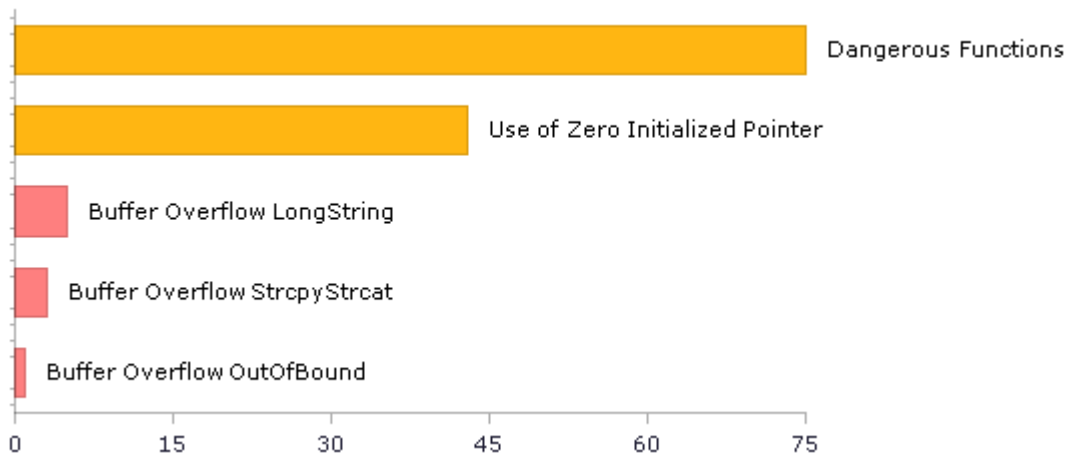
Result Summary



Most Vulnerable Files



Top 5 Vulnerabilities



Scan Summary - OWASP Top 10 2017

Further details and elaboration about vulnerabilities and risks can be found at: [OWASP Top 10 2017](#)

| Category | Threat Agent | Exploitability | Weakness Prevalence | Weakness Detectability | Technical Impact | Business Impact | Issues Found | Best Fix Locations |
|---|---------------|----------------|---------------------|------------------------|------------------|-----------------|--------------|--------------------|
| A1-Injection | App. Specific | EASY | COMMON | EASY | SEVERE | App. Specific | 41 | 32 |
| A2-Broken Authentication | App. Specific | EASY | COMMON | AVERAGE | SEVERE | App. Specific | 10 | 10 |
| 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 | 75 | 75 |
| 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 | 75 | 75 |
| 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 | 3 | 3 |
| PCI DSS (3.2) - 6.5.2 - Buffer overflows | 27 | 24 |
| PCI DSS (3.2) - 6.5.3 - Insecure cryptographic storage | 0 | 0 |
| PCI DSS (3.2) - 6.5.4 - Insecure communications | 0 | 0 |
| PCI DSS (3.2) - 6.5.5 - Improper error handling* | 0 | 0 |
| PCI DSS (3.2) - 6.5.7 - Cross-site scripting (XSS) | 0 | 0 |
| PCI DSS (3.2) - 6.5.8 - Improper access control | 0 | 0 |
| PCI DSS (3.2) - 6.5.9 - Cross-site request forgery | 0 | 0 |
| PCI DSS (3.2) - 6.5.10 - Broken authentication and session management | 0 | 0 |

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

Scan Summary - FISMA 2014

| Category | Description | Issues Found | Best Fix Locations |
|--------------------------------------|--|--------------|--------------------|
| Access Control | Organizations must limit information system access to authorized users, processes acting on behalf of authorized users, or devices (including other information systems) and to the types of transactions and functions that authorized users are permitted to exercise. | 0 | 0 |
| Audit And Accountability* | Organizations must: (i) create, protect, and retain information system audit records to the extent needed to enable the monitoring, analysis, investigation, and reporting of unlawful, unauthorized, or inappropriate information system activity; and (ii) ensure that the actions of individual information system users can be uniquely traced to those users so they can be held accountable for their actions. | 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. | 10 | 10 |
| 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) | 10 | 10 |
| 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)* | 55 | 16 |
| SC-8 Transmission Confidentiality and Integrity (P1) | 0 | 0 |
| SI-10 Information Input Validation (P1)* | 35 | 32 |
| SI-11 Error Handling (P2)* | 9 | 9 |
| SI-15 Information Output Filtering (P0) | 0 | 0 |
| SI-16 Memory Protection (P1) | 3 | 3 |

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

Scan Summary - OWASP Mobile Top 10 2016

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

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

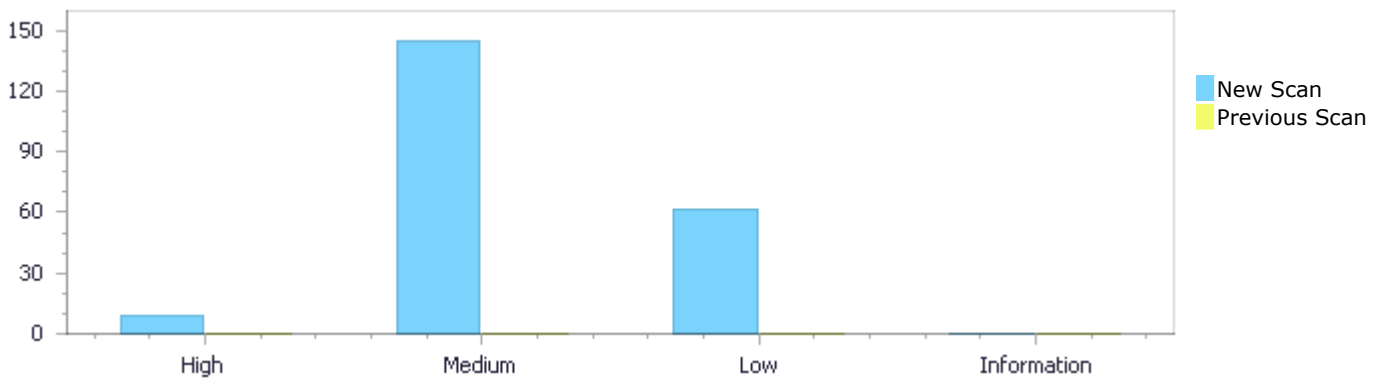
Scan Summary - Custom

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

Results Distribution By Status First scan of the project

| | High | Medium | Low | Information | Total |
|------------------|------|--------|-----|-------------|-------|
| New Issues | 9 | 145 | 61 | 0 | 215 |
| Recurrent Issues | 0 | 0 | 0 | 0 | 0 |
| Total | 9 | 145 | 61 | 0 | 215 |

| | | | | | |
|--------------|---|---|---|---|---|
| Fixed Issues | 0 | 0 | 0 | 0 | 0 |
|--------------|---|---|---|---|---|



Results Distribution By State

| | High | Medium | Low | Information | Total |
|--------------------------|------|--------|-----|-------------|-------|
| Confirmed | 0 | 0 | 0 | 0 | 0 |
| Not Exploitable | 0 | 0 | 0 | 0 | 0 |
| To Verify | 9 | 145 | 61 | 0 | 215 |
| Urgent | 0 | 0 | 0 | 0 | 0 |
| Proposed Not Exploitable | 0 | 0 | 0 | 0 | 0 |
| Total | 9 | 145 | 61 | 0 | 215 |

Result Summary

| Vulnerability Type | Occurrences | Severity |
|---|-------------|----------|
| Buffer Overflow LongString | 5 | High |
| Buffer Overflow StrcpyStrcat | 3 | High |
| Buffer Overflow OutOfBound | 1 | High |
| Dangerous Functions | 75 | Medium |
| Use of Zero Initialized Pointer | 43 | Medium |

| | | |
|---|----|--------|
| Buffer Overflow boundcpy WrongSizeParam | 16 | Medium |
| Inadequate Encryption Strength | 6 | Medium |
| Buffer Overflow AddressOfLocalVarReturned | 2 | Medium |
| Divide By Zero | 1 | Medium |
| Memory Leak | 1 | Medium |
| Wrong Size t Allocation | 1 | Medium |
| Unchecked Array Index | 24 | Low |
| Improper Resource Access Authorization | 10 | Low |
| NULL Pointer Dereference | 9 | Low |
| Unchecked Return Value | 9 | Low |
| Potential Off by One Error in Loops | 3 | Low |
| Potential Precision Problem | 2 | Low |
| Use of Sizeof On a Pointer Type | 2 | Low |
| Sizeof Pointer Argument | 1 | Low |
| TOCTOU | 1 | Low |

10 Most Vulnerable Files

High and Medium Vulnerabilities

| File Name | Issues Found |
|--|--------------|
| TencentOS-tiny/cJSON.c | 68 |
| TencentOS-tiny/iniparser.c | 25 |
| TencentOS-tiny/bin_file_download_through_module.c | 17 |
| TencentOS-tiny/eap.c | 17 |
| TencentOS-tiny/crypt_blowfish.c | 11 |
| TencentOS-tiny/lv_draw_img.c | 8 |
| TencentOS-tiny/lv_draw_basic.c | 6 |
| TencentOS-tiny/arm_depthwise_separable_conv_HWC_q7_nonsquare.c | 1 |
| TencentOS-tiny/fsl_str.c | 1 |

Scan Results Details

Buffer Overflow LongString

Query Path:

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

Categories

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

Description

Buffer Overflow LongString\Path 1:

| | |
|----------------|---|
| Severity | High |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=1 |
| Status | New |

The size of the buffer used by application_entry in file, at line 220 of TencentOS-tiny/bin_file_download_through_module.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that application_entry passes to "http://192.168.1.107:80/test_2048.bin", at line 220 of TencentOS-tiny/bin_file_download_through_module.c, to overwrite the target buffer.

| | Source | Destination |
|--------|---|---|
| File | TencentOS-tiny/bin_file_download_through_module.c | TencentOS-tiny/bin_file_download_through_module.c |
| Line | 231 | 247 |
| Object | "http://192.168.1.107:80/test_2048.bin" | file |

Code Snippet

File Name TencentOS-tiny/bin_file_download_through_module.c
Method void application_entry(void *arg)

```
....  
231.      char *url = "http://192.168.1.107:80/test_2048.bin";  
....  
247.      printf("  -file:%s\r\n", file);
```

Buffer Overflow LongString\Path 2:

| | |
|----------------|---|
| Severity | High |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=2 |
| Status | New |

The size of the buffer used by BF_set_key in tmp, at line 543 of TencentOS-tiny/crypt_blowfish.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *_crypt_blowfish_rn passes to "8b \xd0\xc1\xd2\xcf\xcc\xd8", at line 816 of TencentOS-tiny/crypt_blowfish.c, to overwrite the target buffer.

| | Source | Destination |
|--------|---------------------------------|---------------------------------|
| File | TencentOS-tiny/crypt_blowfish.c | TencentOS-tiny/crypt_blowfish.c |
| Line | 819 | 596 |
| Object | "8b \xd0\xc1\xd2\xcf\xcc\xd8" | tmp |

Code Snippet

File Name TencentOS-tiny/crypt_blowfish.c

Method char *_crypt_blowfish_rn(const char *key, const char *setting,

```
....
819.          const char *test_key = "8b \xd0\xc1\xd2\xcf\xcc\xd8";
```

File Name TencentOS-tiny/crypt_blowfish.c

Method static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

```
....
596.          tmp[1] |= (BF_word_signed)(signed char)*ptr; /*
bug */
```

Buffer Overflow LongString\Path 3:

Severity High

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=3>

Status New

The size of the buffer used by BF_set_key in tmp, at line 543 of TencentOS-tiny/crypt_blowfish.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *_crypt_blowfish_rn passes to "8b \xd0\xc1\xd2\xcf\xcc\xd8", at line 816 of TencentOS-tiny/crypt_blowfish.c, to overwrite the target buffer.

| | Source | Destination |
|--------|---------------------------------|---------------------------------|
| File | TencentOS-tiny/crypt_blowfish.c | TencentOS-tiny/crypt_blowfish.c |
| Line | 819 | 594 |
| Object | "8b \xd0\xc1\xd2\xcf\xcc\xd8" | tmp |

Code Snippet

File Name TencentOS-tiny/crypt_blowfish.c

Method char *_crypt_blowfish_rn(const char *key, const char *setting,

```
....
819.          const char *test_key = "8b \xd0\xc1\xd2\xcf\xcc\xd8";
```

File Name TencentOS-tiny/crypt_blowfish.c
Method static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

```
....  
594.                tmp[0] |= (unsigned char)*ptr; /* correct */
```

Buffer Overflow LongString\Path 4:

Severity High
Result State To Verify
Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=4>
Status New

The size of the buffer used by BF_set_key in tmp, at line 543 of TencentOS-tiny/crypt_blowfish.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *_crypt_blowfish_rn passes to "\xff\xa3", at line 816 of TencentOS-tiny/crypt_blowfish.c, to overwrite the target buffer.

| | Source | Destination |
|--------|---------------------------------|---------------------------------|
| File | TencentOS-tiny/crypt_blowfish.c | TencentOS-tiny/crypt_blowfish.c |
| Line | 861 | 594 |
| Object | "\xff\xa3" | tmp |

Code Snippet

File Name TencentOS-tiny/crypt_blowfish.c
Method char *_crypt_blowfish_rn(const char *key, const char *setting,

```
....  
861.                const char *k = "\xff\xa3" "34" "\xff\xff\xff\xa3"  
"345";
```

File Name TencentOS-tiny/crypt_blowfish.c
Method static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

```
....  
594.                tmp[0] |= (unsigned char)*ptr; /* correct */
```

Buffer Overflow LongString\Path 5:

Severity High
Result State To Verify
Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=5>
Status New

The size of the buffer used by BF_set_key in tmp, at line 543 of TencentOS-tiny/crypt_blowfish.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source

buffer that *_crypt_blowfish_rn passes to "\xff\xa3", at line 816 of TencentOS-tiny/crypt_blowfish.c, to overwrite the target buffer.

| | Source | Destination |
|--------|---------------------------------|---------------------------------|
| File | TencentOS-tiny/crypt_blowfish.c | TencentOS-tiny/crypt_blowfish.c |
| Line | 861 | 596 |
| Object | "\xff\xa3" | tmp |

Code Snippet

File Name TencentOS-tiny/crypt_blowfish.c

Method char *_crypt_blowfish_rn(const char *key, const char *setting,

```
....
861.             const char *k = "\xff\xa3" "34" "\xff\xff\xff\xa3"
"345";
```

File Name TencentOS-tiny/crypt_blowfish.c

Method static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

```
....
596.             tmp[1] |= (BF_word_signed)(signed char)*ptr; /*
bug */
```

Buffer Overflow StrcpyStrcat

Query Path:

CPP\Cx\CPP Buffer Overflow\Buffer Overflow StrcpyStrcat 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 StrcpyStrcat\Path 1:

| | |
|----------------|---|
| Severity | High |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=6 |
| Status | New |

The size of the buffer used by ota_url_parse in file, at line 43 of TencentOS-tiny/bin_file_download_through_module.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ota_url_parse passes to file, at line 43 of TencentOS-tiny/bin_file_download_through_module.c, to overwrite the target buffer.

| | Source | Destination |
|------|---|---|
| File | TencentOS-tiny/bin_file_download_through_module.c | TencentOS-tiny/bin_file_download_through_module.c |

| | | |
|--------|------|------|
| Line | 43 | 83 |
| Object | file | file |

Code Snippet

File Name TencentOS-tiny/bin_file_download_through_module.c

Method static int ota_url_parse(char *url, char *host, uint16_t *port, char *file)

```
....
43. static int ota_url_parse(char *url, char *host, uint16_t *port,
char *file)
....
83.     strcpy(file, url);
```

Buffer Overflow StrcpyStrcat\Path 2:

Severity High

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=7>

Status New

The size of the buffer used by print_string_ptr in output, at line 906 of TencentOS-tiny/cJSON.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that print_string_ptr passes to input, at line 906 of TencentOS-tiny/cJSON.c, to overwrite the target buffer.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 906 | 928 |
| Object | input | output |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_string_ptr(const unsigned char * const input, printbuffer * const output_buffer)

```
....
906. static cJSON_bool print_string_ptr(const unsigned char * const
input, printbuffer * const output_buffer)
....
928.     strcpy((char*) output, "\""");
```

Buffer Overflow StrcpyStrcat\Path 3:

Severity High

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=8>

Status New

The size of the buffer used by iniparser_line in input_line, at line 690 of TencentOS-tiny/iniparser.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source

buffer that iniparser_line passes to input_line, at line 690 of TencentOS-tiny/iniparser.c, to overwrite the target buffer.

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 692 | 713 |
| Object | input_line | input_line |

Code Snippet

File Name TencentOS-tiny/iniparser.c

Method static line_status iniparser_line(

```
....
692.     const char * input_line,
....
713.     strcpy(line, input_line);
```

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=1050050&projectid=50040&pathid=9>

Status New

The size of the buffer used by *BF_crypt in i, at line 649 of TencentOS-tiny/crypt_blowfish.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that BF_set_key passes to tmp, at line 543 of TencentOS-tiny/crypt_blowfish.c, to overwrite the target buffer.

| | Source | Destination |
|--------|---------------------------------|---------------------------------|
| File | TencentOS-tiny/crypt_blowfish.c | TencentOS-tiny/crypt_blowfish.c |
| Line | 548 | 764 |
| Object | tmp | i |

Code Snippet

File Name TencentOS-tiny/crypt_blowfish.c

Method static void BF_set_key(const char *key, BF_key expanded, BF_key initial,

```
....
548.     BF_word safety, sign, diff, tmp[2];
```

File Name TencentOS-tiny/crypt_blowfish.c
Method static char *BF_crypt(const char *key, const char *setting,

```
....  
764. data.binary.output[i] = L;
```

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=1050050&projectid=50040&pathid=80 |
| Status | New |

The dangerous function, memcpy, was found in use at line 74 in TencentOS-tiny/arm_depthwise_separable_conv_HWC_q7_nonsquare.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|--|--|
| File | TencentOS-tiny/arm_depthwise_separable_conv_HWC_q7_nonsquare.c | TencentOS-tiny/arm_depthwise_separable_conv_HWC_q7_nonsquare.c |
| Line | 141 | 141 |
| Object | memcpy | memcpy |

Code Snippet

File Name TencentOS-tiny/arm_depthwise_separable_conv_HWC_q7_nonsquare.c
Method arm_status arm_depthwise_separable_conv_HWC_q7_nonsquare(const q7_t *Im_in,

```
....  
141. memcpy(pBuffer, (q7_t *) Im_in + (i_ker_y  
* dim_im_in_x + i_ker_x) * ch_im_in, ch_im_in);
```

Dangerous Functions\Path 2:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=81 |
| Status | New |

The dangerous function, memcpy, was found in use at line 192 in TencentOS-tiny/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 208 | 208 |
| Object | memcpy | memcpy |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static unsigned char* cJSON_strdup(const unsigned char* string, const internal_hooks * const hooks)

```
....  
208.      memcpy(copy, string, length);
```

Dangerous Functions\Path 3:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=82>

Status New

The dangerous function, memcpy, was found in use at line 443 in TencentOS-tiny/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 519 | 519 |
| Object | memcpy | memcpy |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....  
519.      memcpy(newbuffer, p->buffer, p->offset + 1);
```

Dangerous Functions\Path 4:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=83>

Status New

The dangerous function, memcpy, was found in use at line 906 in TencentOS-tiny/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 969 | 969 |
| Object | memcpy | memcpy |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_string_ptr(const unsigned char * const input, printbuffer * const output_buffer)

```
....  
969.         memcpy(output + 1, input, output_length);
```

Dangerous Functions\Path 5:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=84>

Status New

The dangerous function, memcpy, was found in use at line 1191 in TencentOS-tiny/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 1232 | 1232 |
| Object | memcpy | memcpy |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static unsigned char *print(const cJSON * const item, cJSON_bool format, const internal_hooks * const hooks)

```
....  
1232.         memcpy(printed, buffer->buffer, cJSON_min(buffer->length,  
buffer->offset + 1));
```

Dangerous Functions\Path 6:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=85>

Status New

The dangerous function, memcpy, was found in use at line 1371 in TencentOS-tiny/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 1426 | 1426 |
| Object | memcpy | memcpy |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_value(const cJSON * const item, printbuffer * const output_buffer)

```
....  
1426.             memcpy(output, item->valuelstring, raw_length);
```

Dangerous Functions\Path 7:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=86>

Status New

The dangerous function, memcpy, was found in use at line 1939 in TencentOS-tiny/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 1953 | 1953 |
| Object | memcpy | memcpy |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static cJSON *create_reference(const cJSON *item, const internal_hooks * const hooks)

```
....  
1953.             memcpy(reference, item, sizeof(cJSON));
```

Dangerous Functions\Path 8:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=87>

Status New

The dangerous function, memcpy, was found in use at line 649 in TencentOS-tiny/crypt_blowfish.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|---------------------------------|---------------------------------|
| File | TencentOS-tiny/crypt_blowfish.c | TencentOS-tiny/crypt_blowfish.c |
| Line | 698 | 698 |
| Object | memcpy | memcpy |

Code Snippet

File Name TencentOS-tiny/crypt_blowfish.c

Method static char *BF_crypt(const char *key, const char *setting,

```
.....  
698.         memcpy(data.ctx.S, BF_init_state.S, sizeof(data.ctx.S));
```

Dangerous Functions\Path 9:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=88>

Status New

The dangerous function, memcpy, was found in use at line 649 in TencentOS-tiny/crypt_blowfish.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|---------------------------------|---------------------------------|
| File | TencentOS-tiny/crypt_blowfish.c | TencentOS-tiny/crypt_blowfish.c |
| Line | 768 | 768 |
| Object | memcpy | memcpy |

Code Snippet

File Name TencentOS-tiny/crypt_blowfish.c

Method static char *BF_crypt(const char *key, const char *setting,

```
.....  
768.         memcpy(output, setting, 7 + 22 - 1);
```

Dangerous Functions\Path 10:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=89>

Status New

The dangerous function, memcpy, was found in use at line 816 in TencentOS-tiny/crypt_blowfish.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|------|---------------------------------|---------------------------------|
| File | TencentOS-tiny/crypt_blowfish.c | TencentOS-tiny/crypt_blowfish.c |

| | | |
|--------|--------|--------|
| Line | 845 | 845 |
| Object | memcpy | memcpy |

Code Snippet

File Name TencentOS-tiny/crypt_blowfish.c

Method char *_crypt_blowfish_rn(const char *key, const char *setting,

```
....  
845.         memcpy(buf.s, test_setting, sizeof(buf.s));
```

Dangerous Functions\Path 11:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=90>

Status New

The dangerous function, memcpy, was found in use at line 565 in TencentOS-tiny/lv_draw_basic.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|--------------------------------|--------------------------------|
| File | TencentOS-tiny/lv_draw_basic.c | TencentOS-tiny/lv_draw_basic.c |
| Line | 568 | 568 |
| Object | memcpy | memcpy |

Code Snippet

File Name TencentOS-tiny/lv_draw_basic.c

Method static void sw_mem_blend(lv_color_t * dest, const lv_color_t * src, uint32_t length, lv_opa_t opa)

```
....  
568.         memcpy(dest, src, length * sizeof(lv_color_t));
```

Dangerous Functions\Path 12:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=91>

Status New

The dangerous function, memcpy, was found in use at line 585 in TencentOS-tiny/lv_draw_basic.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|------|--------------------------------|--------------------------------|
| File | TencentOS-tiny/lv_draw_basic.c | TencentOS-tiny/lv_draw_basic.c |
| Line | 616 | 616 |

| | | |
|--------|--------|--------|
| Object | memcpy | memcpy |
|--------|--------|--------|

Code Snippet

File Name TencentOS-tiny/lv_draw_basic.c

Method static void sw_color_fill(lv_color_t * mem, lv_coord_t mem_width, const lv_area_t * fill_area, lv_color_t color,

```
....
616.                memcpy(&mem[fill_area->x1], mem_first, copy_size);
```

Dangerous Functions\Path 13:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=92>

Status New

The dangerous function, memcpy, was found in use at line 78 in TencentOS-tiny/lv_draw_img.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 103 | 103 |
| Object | memcpy | memcpy |

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c

Method lv_color_t lv_img_buf_get_px_color(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y, const lv_style_t * style)

```
....
103.                memcpy(&p_color, &buf_u8[px], sizeof(lv_color_t));
```

Dangerous Functions\Path 14:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=93>

Status New

The dangerous function, memcpy, was found in use at line 213 in TencentOS-tiny/lv_draw_img.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 220 | 220 |

| | | |
|--------|--------|--------|
| Object | memcpy | memcpy |
|--------|--------|--------|

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c

Method void lv_img_buf_set_px_color(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y, lv_color_t c)

```
....
220.         memcpy(&buf_u8[px], &c, px_size);
```

Dangerous Functions\Path 15:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=94>

Status New

The dangerous function, memcpy, was found in use at line 213 in TencentOS-tiny/lv_draw_img.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 224 | 224 |
| Object | memcpy | memcpy |

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c

Method void lv_img_buf_set_px_color(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y, lv_color_t c)

```
....
224.         memcpy(&buf_u8[px], &c, px_size - 1); /*-1 to not
overwrite the alpha value*/
```

Dangerous Functions\Path 16:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=95>

Status New

The dangerous function, memcpy, was found in use at line 310 in TencentOS-tiny/lv_draw_img.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 321 | 321 |

| | | |
|--------|--------|--------|
| Object | memcpy | memcpy |
|--------|--------|--------|

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c

Method void lv_img_buf_set_palette(lv_img_dsc_t * dsc, uint8_t id, lv_color_t c)

```
....
321.      memcpy(&buf[id * sizeof(c32)], &c32, sizeof(c32));
```

Dangerous Functions\Path 17:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=96>

Status New

The dangerous function, sprintf, was found in use at line 220 in TencentOS-tiny/bin_file_download_through_module.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|---|---|
| File | TencentOS-tiny/bin_file_download_through_module.c | TencentOS-tiny/bin_file_download_through_module.c |
| Line | 243 | 243 |
| Object | sprintf | sprintf |

Code Snippet

File Name TencentOS-tiny/bin_file_download_through_module.c

Method void application_entry(void *arg)

```
....
243.      sprintf(port, "%d", port_num);
```

Dangerous Functions\Path 18:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=97>

Status New

The dangerous function, sprintf, was found in use at line 125 in TencentOS-tiny/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 128 | 128 |

| | | |
|--------|---------|---------|
| Object | sprintf | sprintf |
|--------|---------|---------|

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method cJSON_PUBLIC(const char*) cJSON_Version(void)

```
....  
128.      sprintf(version, "%i.%i.%i", cJSON_VERSION_MAJOR,  
cJSON_VERSION_MINOR, cJSON_VERSION_PATCH);
```

Dangerous Functions\Path 19:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=98>

Status New

The dangerous function, sprintf, was found in use at line 549 in TencentOS-tiny/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 567 | 567 |
| Object | sprintf | sprintf |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_number(const cJSON * const item, printbuffer * const output_buffer)

```
....  
567.      length = sprintf((char*)number_buffer, "null");
```

Dangerous Functions\Path 20:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=99>

Status New

The dangerous function, sprintf, was found in use at line 549 in TencentOS-tiny/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 571 | 571 |

| | | |
|--------|---------|---------|
| Object | sprintf | sprintf |
|--------|---------|---------|

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_number(const cJSON * const item, printbuffer * const output_buffer)

```
....
571.          length = sprintf((char*)number_buffer, "%d", item-
>valueint);
```

Dangerous Functions\Path 21:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=100>

Status New

The dangerous function, sprintf, was found in use at line 549 in TencentOS-tiny/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 576 | 576 |
| Object | sprintf | sprintf |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_number(const cJSON * const item, printbuffer * const output_buffer)

```
....
576.          length = sprintf((char*)number_buffer, "%1.15g", d);
```

Dangerous Functions\Path 22:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=101>

Status New

The dangerous function, sprintf, was found in use at line 549 in TencentOS-tiny/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 582 | 582 |

| | | |
|--------|---------|---------|
| Object | sprintf | sprintf |
|--------|---------|---------|

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_number(const cJSON * const item, printbuffer * const output_buffer)

```
....  
582.                length = sprintf((char*)number_buffer, "%1.17g", d);
```

Dangerous Functions\Path 23:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=102>

Status New

The dangerous function, sprintf, was found in use at line 906 in TencentOS-tiny/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 1015 | 1015 |
| Object | sprintf | sprintf |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_string_ptr(const unsigned char * const input, printbuffer * const output_buffer)

```
....  
1015.                sprintf((char*)output_pointer, "u%04x",  
*input_pointer);
```

Dangerous Functions\Path 24:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=103>

Status New

The dangerous function, sscanf, was found in use at line 174 in TencentOS-tiny/bin_file_download_through_module.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|------|--|--|
| File | TencentOS-tiny/bin_file_download_through_module. | TencentOS-tiny/bin_file_download_through_module. |

| | | |
|--------|--------|--------|
| | c | c |
| Line | 190 | 190 |
| Object | sscanf | sscanf |

Code Snippet

File Name TencentOS-tiny/bin_file_download_through_module.c
Method static int ota_http_response_parse(int fd)

```
....  
190.                sscanf(line_buf, "%s %d", header_field,  
&header_value);
```

Dangerous Functions\Path 25:

Severity Medium
Result State To Verify
Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=104>
Status New

The dangerous function, sscanf, was found in use at line 174 in TencentOS-tiny/bin_file_download_through_module.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|---|---|
| File | TencentOS-tiny/bin_file_download_through_module.c | TencentOS-tiny/bin_file_download_through_module.c |
| Line | 197 | 197 |
| Object | sscanf | sscanf |

Code Snippet

File Name TencentOS-tiny/bin_file_download_through_module.c
Method static int ota_http_response_parse(int fd)

```
....  
197.                sscanf(line_buf, "%s %d", header_field,  
&header_value);
```

Dangerous Functions\Path 26:

Severity Medium
Result State To Verify
Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=105>
Status New

The dangerous function, sscanf, was found in use at line 549 in TencentOS-tiny/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 579 | 579 |
| Object | sscanf | sscanf |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_number(const cJSON * const item, printbuffer * const output_buffer)

```
....  
579.             if ((sscanf((char*)number_buffer, "%lg", &test) != 1) ||  
!compare_double((double)test, d))
```

Dangerous Functions\Path 27:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=106>

Status New

The dangerous function, sscanf, was found in use at line 690 in TencentOS-tiny/iniparser.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 755 | 755 |
| Object | sscanf | sscanf |

Code Snippet

File Name TencentOS-tiny/iniparser.c

Method static line_status iniparser_line(

```
....  
755.             sscanf(line, "[%[^]]", key);
```

Dangerous Functions\Path 28:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=107>

Status New

The dangerous function, sscanf, was found in use at line 690 in TencentOS-tiny/iniparser.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 763 | 763 |
| Object | sscanf | sscanf |

Code Snippet

File Name TencentOS-tiny/iniparser.c
Method static line_status iniparser_line(

```
.....
763.         else if (equals && (sscanf (line, "%[^=] = \"%[^\"]\\\"", key,
value) == 2
```

Dangerous Functions\Path 29:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=108 |
| Status | New |

The dangerous function, sscanf, was found in use at line 690 in TencentOS-tiny/iniparser.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 764 | 764 |
| Object | sscanf | sscanf |

Code Snippet

File Name TencentOS-tiny/iniparser.c
Method static line_status iniparser_line(

```
.....
764.         || sscanf (line, "%[^=] = '%[^\\']'",
key, value) == 2
```

Dangerous Functions\Path 30:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=109 |
| Status | New |

The dangerous function, sscanf, was found in use at line 690 in TencentOS-tiny/iniparser.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| Source | Destination |
|--------|-------------|
|--------|-------------|

| | | |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 765 | 765 |
| Object | sscanf | sscanf |

Code Snippet

File Name TencentOS-tiny/iniparser.c
Method static line_status iniparser_line(

```
....
765.                                     ||  sscanf (line, "%[^] = %[^;#]",
key, value) == 2))
```

Dangerous Functions\Path 31:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=110 |
| Status | New |

The dangerous function, sscanf, was found in use at line 690 in TencentOS-tiny/iniparser.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 787 | 787 |
| Object | sscanf | sscanf |

Code Snippet

File Name TencentOS-tiny/iniparser.c
Method static line_status iniparser_line(

```
....
787.     else if (equals && (sscanf(line, "%[^] = %[^;#]", key, value)
== 2
```

Dangerous Functions\Path 32:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=111 |
| Status | New |

The dangerous function, sscanf, was found in use at line 690 in TencentOS-tiny/iniparser.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |

| | | |
|--------|--------|--------|
| Line | 788 | 788 |
| Object | sscanf | sscanf |

Code Snippet

File Name TencentOS-tiny/iniparser.c
Method static line_status iniparser_line(

```
....
788.                                     ||  sscanf(line, "%[^\n] %[=]", key, value)
== 2))
```

Dangerous Functions\Path 33:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=112 |
| Status | New |

The dangerous function, strcpy, was found in use at line 43 in TencentOS-tiny/bin_file_download_through_module.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|---|---|
| File | TencentOS-tiny/bin_file_download_through_module.c | TencentOS-tiny/bin_file_download_through_module.c |
| Line | 83 | 83 |
| Object | strcpy | strcpy |

Code Snippet

File Name TencentOS-tiny/bin_file_download_through_module.c
Method static int ota_url_parse(char *url, char *host, uint16_t *port, char *file)

```
....
83.      strcpy(file, url);
```

Dangerous Functions\Path 34:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=113 |
| Status | New |

The dangerous function, strcpy, was found in use at line 404 in TencentOS-tiny/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| Source | Destination |
|--------|-------------|
|--------|-------------|

| | | |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 414 | 414 |
| Object | strcpy | strcpy |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method cJSON_PUBLIC(char*) cJSON_SetValuestring(cJSON *object, const char *valuestring)

```
....  
414.         strcpy(object->valuestring, valuestring);
```

Dangerous Functions\Path 35:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=114>

Status New

The dangerous function, strcpy, was found in use at line 906 in TencentOS-tiny/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 928 | 928 |
| Object | strcpy | strcpy |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_string_ptr(const unsigned char * const input, printbuffer * const output_buffer)

```
....  
928.         strcpy((char*)output, "\"" + input + "\"");
```

Dangerous Functions\Path 36:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=115>

Status New

The dangerous function, strcpy, was found in use at line 1371 in TencentOS-tiny/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| Source | Destination |
|--------|-------------|
|--------|-------------|

| | | |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 1388 | 1388 |
| Object | strcpy | strcpy |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_value(const cJSON * const item, printbuffer * const output_buffer)

```
....  
1388.                strcpy((char*)output, "null");
```

Dangerous Functions\Path 37:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=116>

Status New

The dangerous function, strcpy, was found in use at line 1371 in TencentOS-tiny/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 1397 | 1397 |
| Object | strcpy | strcpy |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_value(const cJSON * const item, printbuffer * const output_buffer)

```
....  
1397.                strcpy((char*)output, "false");
```

Dangerous Functions\Path 38:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=117>

Status New

The dangerous function, strcpy, was found in use at line 1371 in TencentOS-tiny/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| Source | Destination |
|--------|-------------|
|--------|-------------|

| | | |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 1406 | 1406 |
| Object | strcpy | strcpy |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_value(const cJSON * const item, printbuffer * const output_buffer)

```
....  
1406.                strcpy((char*)output, "true");
```

Dangerous Functions\Path 39:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=118>

Status New

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

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

Code Snippet

File Name TencentOS-tiny/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 40:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=119>

Status New

The dangerous function, strcpy, was found in use at line 690 in TencentOS-tiny/iniparser.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |

| | | |
|--------|--------|--------|
| Line | 713 | 713 |
| Object | strcpy | strcpy |

Code Snippet

File Name TencentOS-tiny/iniparser.c
Method static line_status iniparser_line(

```
....
713.         strcpy(line, input_line);
```

Dangerous Functions\Path 41:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=120 |
| Status | New |

The dangerous function, strcpy, was found in use at line 850 in TencentOS-tiny/iniparser.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 997 | 997 |
| Object | strcpy | strcpy |

Code Snippet

File Name TencentOS-tiny/iniparser.c
Method dictionary * iniparser_load(const char * ininame)

```
....
997.         strcpy(full_line, prev_line);
```

Dangerous Functions\Path 42:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=121 |
| Status | New |

The dangerous function, strcpy, was found in use at line 850 in TencentOS-tiny/iniparser.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 1000 | 1000 |
| Object | strcpy | strcpy |

Code Snippet

File Name TencentOS-tiny/iniparser.c
Method dictionary * iniparser_load(const char * ininame)

```
....
1000.          strcpy(full_line + prev_line_len, line);
```

Dangerous Functions\Path 43:

Severity Medium
Result State To Verify
Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=122>
Status New

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

| | Source | Destination |
|--------|---|---|
| File | TencentOS-tiny/bin_file_download_through_module.c | TencentOS-tiny/bin_file_download_through_module.c |
| Line | 261 | 261 |
| Object | strlen | strlen |

Code Snippet

File Name TencentOS-tiny/bin_file_download_through_module.c
Method void application_entry(void *arg)

```
....
261.          if (tos_sal_module_send(fd, (void *)http_request,
strlen(http_request)) < 0) {
```

Dangerous Functions\Path 44:

Severity Medium
Result State To Verify
Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=123>
Status New

The dangerous function, strlen, was found in use at line 43 in TencentOS-tiny/bin_file_download_through_module.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|------|---|---|
| File | TencentOS-tiny/bin_file_download_through_module.c | TencentOS-tiny/bin_file_download_through_module.c |

| | | |
|--------|--------|--------|
| | c | c |
| Line | 52 | 52 |
| Object | strlen | strlen |

Code Snippet

File Name TencentOS-tiny/bin_file_download_through_module.c

Method static int ota_url_parse(char *url, char *host, uint16_t *port, char *file)

```
....  
52.      if (strncmp(url, HTTP_URL_PREFIX, strlen(HTTP_URL_PREFIX)) !=  
0) {
```

Dangerous Functions\Path 45:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=124>

Status New

The dangerous function, strlen, was found in use at line 43 in TencentOS-tiny/bin_file_download_through_module.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|---|---|
| File | TencentOS-tiny/bin_file_download_through_module.c | TencentOS-tiny/bin_file_download_through_module.c |
| Line | 56 | 56 |
| Object | strlen | strlen |

Code Snippet

File Name TencentOS-tiny/bin_file_download_through_module.c

Method static int ota_url_parse(char *url, char *host, uint16_t *port, char *file)

```
....  
56.      url += strlen(HTTP_URL_PREFIX);
```

Dangerous Functions\Path 46:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=125>

Status New

The dangerous function, strlen, was found in use at line 43 in TencentOS-tiny/bin_file_download_through_module.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|---|---|
| File | TencentOS-tiny/bin_file_download_through_module.c | TencentOS-tiny/bin_file_download_through_module.c |
| Line | 80 | 80 |
| Object | strlen | strlen |

Code Snippet

File Name TencentOS-tiny/bin_file_download_through_module.c

Method static int ota_url_parse(char *url, char *host, uint16_t *port, char *file)

```
....
80.         if (strlen(url) > FILE_MAX) {
```

Dangerous Functions\Path 47:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=126>

Status New

The dangerous function, strlen, was found in use at line 174 in TencentOS-tiny/bin_file_download_through_module.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|---|---|
| File | TencentOS-tiny/bin_file_download_through_module.c | TencentOS-tiny/bin_file_download_through_module.c |
| Line | 189 | 189 |
| Object | strlen | strlen |

Code Snippet

File Name TencentOS-tiny/bin_file_download_through_module.c

Method static int ota_http_response_parse(int fd)

```
....
189.         if (strncmp("HTTP/", line_buf, strlen("HTTP/")) == 0) {
```

Dangerous Functions\Path 48:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=127>

Status New

The dangerous function, strlen, was found in use at line 174 in TencentOS-tiny/bin_file_download_through_module.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|---|---|
| File | TencentOS-tiny/bin_file_download_through_module.c | TencentOS-tiny/bin_file_download_through_module.c |
| Line | 196 | 196 |
| Object | strlen | strlen |

Code Snippet

File Name TencentOS-tiny/bin_file_download_through_module.c
Method static int ota_http_response_parse(int fd)

```
....
196.          } else if (strcmp("Content-Length:", line_buf,
strlen("Content-Length:")) == 0) {
```

Dangerous Functions\Path 49:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=128 |
| Status | New |

The dangerous function, strlen, was found in use at line 192 in TencentOS-tiny/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 202 | 202 |
| Object | strlen | strlen |

Code Snippet

File Name TencentOS-tiny/cJSON.c
Method static unsigned char* cJSON_strdup(const unsigned char* string, const internal_hooks * const hooks)

```
....
202.          length = strlen((const char*)string) + sizeof("");
```

Dangerous Functions\Path 50:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=129 |

Status New

The dangerous function, strlen, was found in use at line 404 in TencentOS-tiny/cJSON.c file. Such functions may expose information and allow an attacker to get full control over the host machine.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 412 | 412 |
| Object | strlen | strlen |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method cJSON_PUBLIC(char*) cJSON_SetValuestring(cJSON *object, const char *valuestring)

```

...
412.         if (strlen(valuestring) <= strlen(object->valuestring))

```

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=1050050&projectid=50040&pathid=156>

Status New

The variable declared in after_end at TencentOS-tiny/cJSON.c in line 309 is not initialized when it is used by after_end at TencentOS-tiny/cJSON.c in line 309.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 312 | 381 |
| Object | after_end | after_end |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool parse_number(cJSON * const item, parse_buffer * const input_buffer)

```
....
312.      unsigned char *after_end = NULL;
....
381.      input_buffer->offset += (size_t)(after_end - number_c_string);
```

Use of Zero Initialized Pointer\Path 2:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=157 |
| Status | New |

The variable declared in current_item at TencentOS-tiny/cJSON.c in line 1445 is not initialized when it is used by prev at TencentOS-tiny/cJSON.c in line 1445.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 1448 | 1499 |
| Object | current_item | prev |

Code Snippet

File Name TencentOS-tiny/cJSON.c
 Method static cJSON_bool parse_array(cJSON * const item, parse_buffer * const input_buffer)

```
....
1448.      cJSON *current_item = NULL;
....
1499.      new_item->prev = current_item;
```

Use of Zero Initialized Pointer\Path 3:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=158 |
| Status | New |

The variable declared in current_item at TencentOS-tiny/cJSON.c in line 1605 is not initialized when it is used by prev at TencentOS-tiny/cJSON.c in line 1605.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 1608 | 1657 |
| Object | current_item | prev |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool parse_object(cJSON * const item, parse_buffer * const input_buffer)

```
....
1608.         cJSON *current_item = NULL;
....
1657.             new_item->prev = current_item;
```

Use of Zero Initialized Pointer\Path 4:

Severity Medium
 Result State To Verify
 Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=159>
 Status New

The variable declared in p at TencentOS-tiny/cJSON.c in line 2629 is not initialized when it is used by prev at TencentOS-tiny/cJSON.c in line 1932.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 2633 | 1935 |
| Object | p | prev |

Code Snippet

File Name TencentOS-tiny/cJSON.c
 Method cJSON_PUBLIC(cJSON *) cJSON_CreateDoubleArray(const double *numbers, int count)

```
....
2633.         cJSON *p = NULL;
```

File Name TencentOS-tiny/cJSON.c
 Method static void suffix_object(cJSON *prev, cJSON *item)

```
....
1935.         item->prev = prev;
```

Use of Zero Initialized Pointer\Path 5:

Severity Medium
 Result State To Verify
 Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=160>
 Status New

The variable declared in p at TencentOS-tiny/cJSON.c in line 2589 is not initialized when it is used by prev at TencentOS-tiny/cJSON.c in line 1932.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 2593 | 1935 |
| Object | p | prev |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method cJSON_PUBLIC(cJSON *) cJSON_CreateFloatArray(const float *numbers, int count)

```
....
2593.      cJSON *p = NULL;
```



File Name TencentOS-tiny/cJSON.c

Method static void suffix_object(cJSON *prev, cJSON *item)

```
....
1935.      item->prev = prev;
```

Use of Zero Initialized Pointer\Path 6:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=161>

Status New

The variable declared in p at TencentOS-tiny/cJSON.c in line 2549 is not initialized when it is used by prev at TencentOS-tiny/cJSON.c in line 1932.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 2553 | 1935 |
| Object | p | prev |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method cJSON_PUBLIC(cJSON *) cJSON_CreateIntArray(const int *numbers, int count)

```
....
2553.      cJSON *p = NULL;
```



File Name TencentOS-tiny/cJSON.c

Method static void suffix_object(cJSON *prev, cJSON *item)


```
....
1935.         item->prev = prev;
```

Use of Zero Initialized Pointer\Path 7:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=162 |
| Status | New |

The variable declared in p at TencentOS-tiny/cJSON.c in line 2669 is not initialized when it is used by prev at TencentOS-tiny/cJSON.c in line 1932.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 2673 | 1935 |
| Object | p | prev |

Code Snippet

File Name TencentOS-tiny/cJSON.c
Method cJSON_PUBLIC(cJSON *) cJSON_CreateStringArray(const char *const *strings, int count)

```
....
2673.         cJSON *p = NULL;
```

File Name TencentOS-tiny/cJSON.c
Method static void suffix_object(cJSON *prev, cJSON *item)

```
....
1935.         item->prev = prev;
```

Use of Zero Initialized Pointer\Path 8:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=163 |
| Status | New |

The variable declared in p at TencentOS-tiny/cJSON.c in line 2549 is not initialized when it is used by p at TencentOS-tiny/cJSON.c in line 2549.

| | Source | Destination |
|------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |

| | | |
|--------|------|------|
| Line | 2553 | 2579 |
| Object | p | p |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method cJSON_PUBLIC(cJSON *) cJSON_CreateIntArray(const int *numbers, int count)

```
....
2553.      cJSON *p = NULL;
....
2579.      p = n;
```

Use of Zero Initialized Pointer\Path 9:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=164>

Status New

The variable declared in p at TencentOS-tiny/cJSON.c in line 2629 is not initialized when it is used by next at TencentOS-tiny/cJSON.c in line 1932.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 2633 | 1934 |
| Object | p | next |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method cJSON_PUBLIC(cJSON *) cJSON_CreateDoubleArray(const double *numbers, int count)

```
....
2633.      cJSON *p = NULL;
```

File Name TencentOS-tiny/cJSON.c

Method static void suffix_object(cJSON *prev, cJSON *item)

```
....
1934.      prev->next = item;
```

Use of Zero Initialized Pointer\Path 10:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=165>

Status New

The variable declared in p at TencentOS-tiny/cJSON.c in line 2589 is not initialized when it is used by next at TencentOS-tiny/cJSON.c in line 1932.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 2593 | 1934 |
| Object | p | next |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method cJSON_PUBLIC(cJSON *) cJSON_CreateFloatArray(const float *numbers, int count)

```
....
2593.     cJSON *p = NULL;
```

File Name TencentOS-tiny/cJSON.c

Method static void suffix_object(cJSON *prev, cJSON *item)

```
....
1934.     prev->next = item;
```

Use of Zero Initialized Pointer\Path 11:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=166>

Status New

The variable declared in p at TencentOS-tiny/cJSON.c in line 2549 is not initialized when it is used by next at TencentOS-tiny/cJSON.c in line 1932.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 2553 | 1934 |
| Object | p | next |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method cJSON_PUBLIC(cJSON *) cJSON_CreateIntArray(const int *numbers, int count)

```
....
2553.     cJSON *p = NULL;
```

File Name TencentOS-tiny/cJSON.c
Method static void suffix_object(cJSON *prev, cJSON *item)

```
....
1934.         prev->next = item;
```

Use of Zero Initialized Pointer\Path 12:

Severity Medium
Result State To Verify
Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=167>
Status New

The variable declared in p at TencentOS-tiny/cJSON.c in line 2669 is not initialized when it is used by next at TencentOS-tiny/cJSON.c in line 1932.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 2673 | 1934 |
| Object | p | next |

Code Snippet

File Name TencentOS-tiny/cJSON.c
Method cJSON_PUBLIC(cJSON *) cJSON_CreateStringArray(const char *const *strings, int count)

```
....
2673.         cJSON *p = NULL;
```

File Name TencentOS-tiny/cJSON.c
Method static void suffix_object(cJSON *prev, cJSON *item)

```
....
1934.         prev->next = item;
```

Use of Zero Initialized Pointer\Path 13:

Severity Medium
Result State To Verify
Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=168>
Status New

The variable declared in p at TencentOS-tiny/cJSON.c in line 2589 is not initialized when it is used by p at TencentOS-tiny/cJSON.c in line 2589.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 2593 | 2619 |
| Object | p | p |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method cJSON_PUBLIC(cJSON *) cJSON_CreateFloatArray(const float *numbers, int count)

```
....  
2593.      cJSON *p = NULL;  
....  
2619.      p = n;
```

Use of Zero Initialized Pointer\Path 14:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=169>

Status New

The variable declared in p at TencentOS-tiny/cJSON.c in line 2629 is not initialized when it is used by p at TencentOS-tiny/cJSON.c in line 2629.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 2633 | 2659 |
| Object | p | p |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method cJSON_PUBLIC(cJSON *) cJSON_CreateDoubleArray(const double *numbers, int count)

```
....  
2633.      cJSON *p = NULL;  
....  
2659.      p = n;
```

Use of Zero Initialized Pointer\Path 15:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=170>

Status New

The variable declared in p at TencentOS-tiny/cJSON.c in line 2669 is not initialized when it is used by p at TencentOS-tiny/cJSON.c in line 2669.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 2673 | 2699 |
| Object | p | p |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method cJSON_PUBLIC(cJSON *) cJSON_CreateStringArray(const char *const *strings, int count)

```
....  
2673.         cJSON *p = NULL;  
....  
2699.         p = n;
```

Use of Zero Initialized Pointer\Path 16:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=171>

Status New

The variable declared in a_element at TencentOS-tiny/cJSON.c in line 2990 is not initialized when it is used by a_element at TencentOS-tiny/cJSON.c in line 2990.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 3074 | 3079 |
| Object | a_element | a_element |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method cJSON_PUBLIC(cJSON_bool) cJSON_Compare(const cJSON * const a, const cJSON * const b, const cJSON_bool case_sensitive)

```
....  
3074.         cJSON *a_element = NULL;  
....  
3079.         b_element = get_object_item(b, a_element->string,  
case_sensitive);
```

Use of Zero Initialized Pointer\Path 17:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=171>

[40&pathid=172](#)

Status New

The variable declared in current_section at TencentOS-tiny/iniparser.c in line 850 is not initialized when it is used by section at TencentOS-tiny/iniparser.c in line 850.

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 854 | 1023 |
| Object | current_section | section |

Code Snippet

File Name TencentOS-tiny/iniparser.c

Method dictionary * iniparser_load(const char * ininame)

```
....  
854.      char *current_section = NULL;  
....  
1023.      section = current_section;
```

Use of Zero Initialized Pointer\Path 18:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=173>

Status New

The variable declared in key at TencentOS-tiny/iniparser.c in line 850 is not initialized when it is used by section at TencentOS-tiny/iniparser.c in line 850.

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 908 | 1023 |
| Object | key | section |

Code Snippet

File Name TencentOS-tiny/iniparser.c

Method dictionary * iniparser_load(const char * ininame)

```
....  
908.      key = NULL;  
....  
1023.      section = current_section;
```

Use of Zero Initialized Pointer\Path 19:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=173>

[40&pathid=174](#)**Status** New

The variable declared in `val` at TencentOS-tiny/iniparser.c in line 850 is not initialized when it is used by `section` at TencentOS-tiny/iniparser.c in line 850.

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 914 | 1023 |
| Object | val | section |

Code Snippet**File Name** TencentOS-tiny/iniparser.c**Method** dictionary * iniparser_load(const char * ininame)

```
....  
914.                val = NULL;  
....  
1023.               section = current_section;
```

Use of Zero Initialized Pointer\Path 20:**Severity** Medium**Result State** To Verify**Online Results** <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=175>**Status** New

The variable declared in `full_line` at TencentOS-tiny/iniparser.c in line 850 is not initialized when it is used by `prev_line` at TencentOS-tiny/iniparser.c in line 850.

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 1065 | 972 |
| Object | full_line | prev_line |

Code Snippet**File Name** TencentOS-tiny/iniparser.c**Method** dictionary * iniparser_load(const char * ininame)

```
....  
1065.               full_line = NULL;  
....  
972.               prev_line = full_line;
```

Use of Zero Initialized Pointer\Path 21:**Severity** Medium**Result State** To Verify**Online Results** <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=175>

[40&pathid=176](#)

Status New

The variable declared in full_line at TencentOS-tiny/iniparser.c in line 850 is not initialized when it is used by prev_line at TencentOS-tiny/iniparser.c in line 850.

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 857 | 972 |
| Object | full_line | prev_line |

Code Snippet

File Name TencentOS-tiny/iniparser.c

Method dictionary * iniparser_load(const char * ininame)

```
....
857.      char* full_line = NULL;
....
972.      prev_line = full_line;
```

Use of Zero Initialized Pointer\Path 22:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=177>

Status New

The variable declared in buffer at TencentOS-tiny/cJSON.c in line 443 is not initialized when it is used by newbuffer at TencentOS-tiny/cJSON.c in line 443.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 501 | 496 |
| Object | buffer | newbuffer |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
501.      p->buffer = NULL;
....
496.      newbuffer = (unsigned char*)p->hooks.reallocate(p->buffer,
newsize);
```

Use of Zero Initialized Pointer\Path 23:

Severity Medium

Result State To Verify

Online Results <http://WIN->

| | |
|--------|--|
| | BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=178 |
| Status | New |

The variable declared in buffer at TencentOS-tiny/cJSON.c in line 443 is not initialized when it is used by newbuffer at TencentOS-tiny/cJSON.c in line 443.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 514 | 496 |
| Object | buffer | newbuffer |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
514.         p->buffer = NULL;
....
496.         newbuffer = (unsigned char*)p->hooks.reallocate(p->buffer,
newsize);
```

Use of Zero Initialized Pointer\Path 24:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=179 |
| Status | New |

The variable declared in buffer at TencentOS-tiny/cJSON.c in line 443 is not initialized when it is used by output_pointer at TencentOS-tiny/cJSON.c in line 1718.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 501 | 1750 |
| Object | buffer | output_pointer |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
501.         p->buffer = NULL;
```



File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_object(const cJSON * const item, printbuffer * const output_buffer)

```
....
1750.                output_pointer = ensure(output_buffer, output_buffer-
>depth);
```

Use of Zero Initialized Pointer\Path 25:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=180 |
| Status | New |

The variable declared in buffer at TencentOS-tiny/cJSON.c in line 443 is not initialized when it is used by output_pointer at TencentOS-tiny/cJSON.c in line 1718.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 514 | 1750 |
| Object | buffer | output_pointer |

Code Snippet

File Name TencentOS-tiny/cJSON.c
Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
514.                p->buffer = NULL;
```

File Name TencentOS-tiny/cJSON.c
Method static cJSON_bool print_object(const cJSON * const item, printbuffer * const output_buffer)

```
....
1750.                output_pointer = ensure(output_buffer, output_buffer-
>depth);
```

Use of Zero Initialized Pointer\Path 26:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=181 |
| Status | New |

The variable declared in buffer at TencentOS-tiny/cJSON.c in line 443 is not initialized when it is used by output at TencentOS-tiny/cJSON.c in line 906.

| Source | Destination |
|--------|-------------|
|--------|-------------|

| | | |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 501 | 959 |
| Object | buffer | output |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
501.                p->buffer = NULL;
```

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_string_ptr(const unsigned char * const input, printbuffer * const output_buffer)

```
....
959.                output = ensure(output_buffer, output_length +
sizeof("\"\""));
```

Use of Zero Initialized Pointer\Path 27:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=182>

Status New

The variable declared in buffer at TencentOS-tiny/cJSON.c in line 443 is not initialized when it is used by newbuffer at TencentOS-tiny/cJSON.c in line 443.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 501 | 509 |
| Object | buffer | newbuffer |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
501.                p->buffer = NULL;
....
509.                newbuffer = (unsigned char*)p->hooks.allocate(newsize);
```

Use of Zero Initialized Pointer\Path 28:

Severity Medium

Result State To Verify

Online Results <http://WIN->

| | |
|--------|--|
| | BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=183 |
| Status | New |

The variable declared in buffer at TencentOS-tiny/cJSON.c in line 443 is not initialized when it is used by newbuffer at TencentOS-tiny/cJSON.c in line 443.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 514 | 509 |
| Object | buffer | newbuffer |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
514.         p->buffer = NULL;
....
509.         newbuffer = (unsigned char*)p->hooks.allocate(newsize);
```

Use of Zero Initialized Pointer\Path 29:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=184 |
| Status | New |

The variable declared in buffer at TencentOS-tiny/cJSON.c in line 443 is not initialized when it is used by output_pointer at TencentOS-tiny/cJSON.c in line 1718.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 501 | 1811 |
| Object | buffer | output_pointer |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
501.         p->buffer = NULL;
```



File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_object(const cJSON * const item, printbuffer * const output_buffer)

```
....
1811.         output_pointer = ensure(output_buffer, output_buffer->format
? (output_buffer->depth + 1) : 2);
```

Use of Zero Initialized Pointer\Path 30:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=185 |
| Status | New |

The variable declared in buffer at TencentOS-tiny/cJSON.c in line 443 is not initialized when it is used by output_pointer at TencentOS-tiny/cJSON.c in line 1718.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 514 | 1811 |
| Object | buffer | output_pointer |

Code Snippet

File Name TencentOS-tiny/cJSON.c
Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
514.         p->buffer = NULL;
```

File Name TencentOS-tiny/cJSON.c
Method static cJSON_bool print_object(const cJSON * const item, printbuffer * const output_buffer)

```
....
1811.         output_pointer = ensure(output_buffer, output_buffer->format
? (output_buffer->depth + 1) : 2);
```

Use of Zero Initialized Pointer\Path 31:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=186 |
| Status | New |

The variable declared in buffer at TencentOS-tiny/cJSON.c in line 443 is not initialized when it is used by output_pointer at TencentOS-tiny/cJSON.c in line 1718.

| Source | Destination |
|--------|-------------|
|--------|-------------|

| | | |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 514 | 1791 |
| Object | buffer | output_pointer |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
514.         p->buffer = NULL;
```

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_object(const cJSON * const item, printbuffer * const output_buffer)

```
....
1791.         output_pointer = ensure(output_buffer, length + 1);
```

Use of Zero Initialized Pointer\Path 32:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=187>

Status New

The variable declared in buffer at TencentOS-tiny/cJSON.c in line 443 is not initialized when it is used by output_pointer at TencentOS-tiny/cJSON.c in line 1718.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 501 | 1791 |
| Object | buffer | output_pointer |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
501.         p->buffer = NULL;
```

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_object(const cJSON * const item, printbuffer * const output_buffer)

```
....
1791.          output_pointer = ensure(output_buffer, length + 1);
```

Use of Zero Initialized Pointer\Path 33:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=188 |
| Status | New |

The variable declared in buffer at TencentOS-tiny/cJSON.c in line 443 is not initialized when it is used by buffer at TencentOS-tiny/cJSON.c in line 529.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 501 | 536 |
| Object | buffer | buffer |

Code Snippet

File Name TencentOS-tiny/cJSON.c
Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
501.          p->buffer = NULL;
```

File Name TencentOS-tiny/cJSON.c
Method static void update_offset(printbuffer * const buffer)

```
....
536.          buffer_pointer = buffer->buffer + buffer->offset;
```

Use of Zero Initialized Pointer\Path 34:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=189 |
| Status | New |

The variable declared in buffer at TencentOS-tiny/cJSON.c in line 443 is not initialized when it is used by buffer at TencentOS-tiny/cJSON.c in line 529.

| | Source | Destination |
|------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 514 | 536 |

| | | |
|--------|--------|--------|
| Object | buffer | buffer |
|--------|--------|--------|

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
514.          p->buffer = NULL;
```



File Name TencentOS-tiny/cJSON.c

Method static void update_offset(printbuffer * const buffer)

```
....
536.          buffer_pointer = buffer->buffer + buffer->offset;
```

Use of Zero Initialized Pointer\Path 35:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=190>

Status New

The variable declared in buffer at TencentOS-tiny/cJSON.c in line 443 is not initialized when it is used by output_pointer at TencentOS-tiny/cJSON.c in line 1718.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 514 | 1770 |
| Object | buffer | output_pointer |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
514.          p->buffer = NULL;
```



File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_object(const cJSON * const item, printbuffer * const output_buffer)

```
....
1770.          output_pointer = ensure(output_buffer, length);
```

Use of Zero Initialized Pointer\Path 36:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=191 |
| Status | New |

The variable declared in buffer at TencentOS-tiny/cJSON.c in line 443 is not initialized when it is used by output_pointer at TencentOS-tiny/cJSON.c in line 1718.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 501 | 1770 |
| Object | buffer | output_pointer |

Code Snippet

File Name TencentOS-tiny/cJSON.c
Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
501.         p->buffer = NULL;
```



File Name TencentOS-tiny/cJSON.c
Method static cJSON_bool print_object(const cJSON * const item, printbuffer * const output_buffer)

```
....
1770.         output_pointer = ensure(output_buffer, length);
```

Use of Zero Initialized Pointer\Path 37:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=192 |
| Status | New |

The variable declared in buffer at TencentOS-tiny/cJSON.c in line 443 is not initialized when it is used by output_pointer at TencentOS-tiny/cJSON.c in line 1543.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 514 | 1592 |
| Object | buffer | output_pointer |

Code Snippet

File Name TencentOS-tiny/cJSON.c
Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
514.          p->buffer = NULL;
```

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_array(const cJSON * const item, printbuffer * const output_buffer)

```
....
1592.        output_pointer = ensure(output_buffer, 2);
```

Use of Zero Initialized Pointer\Path 38:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=193>

Status New

The variable declared in buffer at TencentOS-tiny/cJSON.c in line 443 is not initialized when it is used by output_pointer at TencentOS-tiny/cJSON.c in line 1543.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 501 | 1592 |
| Object | buffer | output_pointer |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
501.          p->buffer = NULL;
```

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_array(const cJSON * const item, printbuffer * const output_buffer)

```
....
1592.        output_pointer = ensure(output_buffer, 2);
```

Use of Zero Initialized Pointer\Path 39:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=194>

Status New

The variable declared in buffer at TencentOS-tiny/cJSON.c in line 443 is not initialized when it is used by output_pointer at TencentOS-tiny/cJSON.c in line 1543.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 514 | 1576 |
| Object | buffer | output_pointer |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
514.         p->buffer = NULL;
```



File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_array(const cJSON * const item, printbuffer * const output_buffer)

```
....
1576.         output_pointer = ensure(output_buffer, length + 1);
```

Use of Zero Initialized Pointer\Path 40:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=195>

Status New

The variable declared in buffer at TencentOS-tiny/cJSON.c in line 443 is not initialized when it is used by output_pointer at TencentOS-tiny/cJSON.c in line 1543.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 501 | 1576 |
| Object | buffer | output_pointer |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static unsigned char* ensure(printbuffer * const p, size_t needed)

```
....
501.         p->buffer = NULL;
```

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_array(const cJSON * const item, printbuffer * const output_buffer)

```
....
1576.                output_pointer = ensure(output_buffer, length + 1);
```

Use of Zero Initialized Pointer\Path 41:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=196>

Status New

The variable declared in valustring at TencentOS-tiny/cJSON.c in line 1605 is not initialized when it is used by current_item at TencentOS-tiny/cJSON.c in line 1605.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 1672 | 1671 |
| Object | valustring | current_item |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool parse_object(cJSON * const item, parse_buffer * const input_buffer)

```
....
1672.                current_item->valustring = NULL;
....
1671.                current_item->string = current_item->valustring;
```

Use of Zero Initialized Pointer\Path 42:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=197>

Status New

The variable declared in prev at TencentOS-tiny/cJSON.c in line 2191 is not initialized when it is used by next at TencentOS-tiny/cJSON.c in line 257.

| | Source | Destination |
|------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 2221 | 262 |

| Object | prev | next |
|--------|------|------|
|--------|------|------|

Code Snippet

File Name TencentOS-tiny/cJSON.c
Method cJSON_PUBLIC(cJSON *) cJSON_DetachItemViaPointer(cJSON *parent, cJSON *const item)

```
....
2221.         item->prev = NULL;
```

File Name TencentOS-tiny/cJSON.c
Method cJSON_PUBLIC(void) cJSON_Delete(cJSON *item)

```
....
262.         next = item->next;
```

Use of Zero Initialized Pointer\Path 43:

Severity Medium
Result State To Verify
Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=198>
Status New

The variable declared in next at TencentOS-tiny/cJSON.c in line 2191 is not initialized when it is used by next at TencentOS-tiny/cJSON.c in line 257.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 2222 | 262 |
| Object | next | next |

Code Snippet

File Name TencentOS-tiny/cJSON.c
Method cJSON_PUBLIC(cJSON *) cJSON_DetachItemViaPointer(cJSON *parent, cJSON *const item)

```
....
2222.         item->next = NULL;
```

File Name TencentOS-tiny/cJSON.c
Method cJSON_PUBLIC(void) cJSON_Delete(cJSON *item)

```
....
262.         next = item->next;
```

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=1050050&projectid=50040&pathid=27 |
| Status | New |

The size of the buffer used by *create_reference in cJSON, at line 1939 of TencentOS-tiny/cJSON.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *create_reference passes to cJSON, at line 1939 of TencentOS-tiny/cJSON.c, to overwrite the target buffer.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 1953 | 1953 |
| Object | cJSON | cJSON |

Code Snippet

File Name TencentOS-tiny/cJSON.c
Method static cJSON *create_reference(const cJSON *item, const internal_hooks * const hooks)

```
....
1953.     memcpy(reference, item, sizeof(cJSON));
```

Buffer Overflow boundcpy WrongSizeParam\Path 2:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=28 |
| Status | New |

The size of the buffer used by *BF_crypt in Namespace368826186, at line 649 of TencentOS-tiny/crypt_blowfish.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *BF_crypt passes to Namespace368826186, at line 649 of TencentOS-tiny/crypt_blowfish.c, to overwrite the target buffer.

| | Source | Destination |
|--------|---------------------------------|---------------------------------|
| File | TencentOS-tiny/crypt_blowfish.c | TencentOS-tiny/crypt_blowfish.c |
| Line | 698 | 698 |
| Object | Namespace368826186 | Namespace368826186 |

Code Snippet

File Name TencentOS-tiny/crypt_blowfish.c

Method static char *BF_crypt(const char *key, const char *setting,

```
....
698.         memcpy(data.ctx.S, BF_init_state.S, sizeof(data.ctx.S));
```

Buffer Overflow boundcpy WrongSizeParam\Path 3:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=29>

Status New

The size of the buffer used by *_crypt_blowfish_rn in Namespace368826186, at line 816 of TencentOS-tiny/crypt_blowfish.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *_crypt_blowfish_rn passes to Namespace368826186, at line 816 of TencentOS-tiny/crypt_blowfish.c, to overwrite the target buffer.

| | Source | Destination |
|--------|---------------------------------|---------------------------------|
| File | TencentOS-tiny/crypt_blowfish.c | TencentOS-tiny/crypt_blowfish.c |
| Line | 845 | 845 |
| Object | Namespace368826186 | Namespace368826186 |

Code Snippet

File Name TencentOS-tiny/crypt_blowfish.c

Method char *_crypt_blowfish_rn(const char *key, const char *setting,

```
....
845.         memcpy(buf.s, test_setting, sizeof(buf.s));
```

Buffer Overflow boundcpy WrongSizeParam\Path 4:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=30>

Status New

The size of the buffer used by lv_img_buf_get_px_color in lv_color_t, at line 78 of TencentOS-tiny/lv_draw_img.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that lv_img_buf_get_px_color passes to lv_color_t, at line 78 of TencentOS-tiny/lv_draw_img.c, to overwrite the target buffer.

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 103 | 103 |
| Object | lv_color_t | lv_color_t |

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c

Method lv_color_t lv_img_buf_get_px_color(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y, const lv_style_t * style)

```
....
103.         memcpy(&p_color, &buf_u8[px], sizeof(lv_color_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 5:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=31>

Status New

The size of the buffer used by *cJSON_New_Item in cJSON, at line 245 of TencentOS-tiny/cJSON.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *cJSON_New_Item passes to cJSON, at line 245 of TencentOS-tiny/cJSON.c, to overwrite the target buffer.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 250 | 250 |
| Object | cJSON | cJSON |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static cJSON *cJSON_New_Item(const internal_hooks * const hooks)

```
....
250.         memset(node, '\0', sizeof(cJSON));
```

Buffer Overflow boundcpy WrongSizeParam\Path 6:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=32>

Status New

The size of the buffer used by *_crypt_blowfish_rn in Namespace368826186, at line 816 of TencentOS-tiny/crypt_blowfish.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that *_crypt_blowfish_rn passes to Namespace368826186, at line 816 of TencentOS-tiny/crypt_blowfish.c, to overwrite the target buffer.

| | Source | Destination |
|--------|---------------------------------|---------------------------------|
| File | TencentOS-tiny/crypt_blowfish.c | TencentOS-tiny/crypt_blowfish.c |
| Line | 852 | 852 |
| Object | Namespace368826186 | Namespace368826186 |

Code Snippet

File Name TencentOS-tiny/crypt_blowfish.c

Method char *_crypt_blowfish_rn(const char *key, const char *setting,

```
....  
852.          memset(buf.o, 0x55, sizeof(buf.o));
```

Buffer Overflow boundcpy WrongSizeParam\Path 7:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=33>

Status New

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

| | Source | Destination |
|--------|--------------------------------|--------------------------------|
| File | TencentOS-tiny/lv_draw_basic.c | TencentOS-tiny/lv_draw_basic.c |
| Line | 568 | 568 |
| Object | length | length |

Code Snippet

File Name TencentOS-tiny/lv_draw_basic.c

Method static void sw_mem_blend(lv_color_t * dest, const lv_color_t * src, uint32_t length, lv_opa_t opa)

```
....  
568.          memcpy(dest, src, length * sizeof(lv_color_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 8:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=34>

Status New

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

| | Source | Destination |
|--------|--------------------------------|--------------------------------|
| File | TencentOS-tiny/lv_draw_basic.c | TencentOS-tiny/lv_draw_basic.c |
| Line | 568 | 568 |
| Object | lv_color_t | lv_color_t |

Code Snippet**File Name** TencentOS-tiny/lv_draw_basic.c**Method** static void sw_mem_blend(lv_color_t * dest, const lv_color_t * src, uint32_t length, lv_opa_t opa)

```
....  
568.          memcpy(dest, src, length * sizeof(lv_color_t));
```

Buffer Overflow boundcpy WrongSizeParam\Path 9:**Severity** Medium**Result State** To Verify**Online Results** <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=35>**Status** New

The size of the buffer used by cJSON_strdup in length, at line 192 of TencentOS-tiny/cJSON.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that cJSON_strdup passes to length, at line 192 of TencentOS-tiny/cJSON.c, to overwrite the target buffer.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 208 | 208 |
| Object | length | length |

Code Snippet**File Name** TencentOS-tiny/cJSON.c**Method** static unsigned char* cJSON_strdup(const unsigned char* string, const internal_hooks * const hooks)

```
....  
208.          memcpy(copy, string, length);
```

Buffer Overflow boundcpy WrongSizeParam\Path 10:**Severity** Medium**Result State** To Verify**Online Results** <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=36>**Status** New

The size of the buffer used by print_string_ptr in output_length, at line 906 of TencentOS-tiny/cJSON.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that print_string_ptr passes to output_length, at line 906 of TencentOS-tiny/cJSON.c, to overwrite the target buffer.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 969 | 969 |
| Object | output_length | output_length |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_string_ptr(const unsigned char * const input, printbuffer * const output_buffer)

```
....
969.         memcpy(output + 1, input, output_length);
```

Buffer Overflow boundcpy WrongSizeParam\Path 11:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=37>

Status New

The size of the buffer used by print_value in raw_length, at line 1371 of TencentOS-tiny/cJSON.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that print_value passes to raw_length, at line 1371 of TencentOS-tiny/cJSON.c, to overwrite the target buffer.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 1426 | 1426 |
| Object | raw_length | raw_length |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static cJSON_bool print_value(const cJSON * const item, printbuffer * const output_buffer)

```
....
1426.         memcpy(output, item->valuelstring, raw_length);
```

Buffer Overflow boundcpy WrongSizeParam\Path 12:

Severity Medium

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=38>

Status New

The size of the buffer used by sw_color_fill in copy_size, at line 585 of TencentOS-tiny/lv_draw_basic.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that sw_color_fill passes to copy_size, at line 585 of TencentOS-tiny/lv_draw_basic.c, to overwrite the target buffer.

| | Source | Destination |
|--------|--------------------------------|--------------------------------|
| File | TencentOS-tiny/lv_draw_basic.c | TencentOS-tiny/lv_draw_basic.c |
| Line | 616 | 616 |
| Object | copy_size | copy_size |

Code Snippet

File Name TencentOS-tiny/lv_draw_basic.c
Method static void sw_color_fill(lv_color_t * mem, lv_coord_t mem_width, const lv_area_t * fill_area, lv_color_t color,

```
....  
616.                memcpy(&mem[fill_area->x1], mem_first, copy_size);
```

Buffer Overflow boundcpy WrongSizeParam\Path 13:

Severity Medium
Result State To Verify
Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=39>
Status New

The size of the buffer used by lv_img_buf_set_px_color in px_size, at line 213 of TencentOS-tiny/lv_draw_img.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that lv_img_buf_set_px_color passes to px_size, at line 213 of TencentOS-tiny/lv_draw_img.c, to overwrite the target buffer.

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 220 | 220 |
| Object | px_size | px_size |

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c
Method void lv_img_buf_set_px_color(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y, lv_color_t c)

```
....  
220.                memcpy(&buf_u8[px], &c, px_size);
```

Buffer Overflow boundcpy WrongSizeParam\Path 14:

Severity Medium
Result State To Verify
Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=40>
Status New

The size of the buffer used by ota_http_response_line_parse in buf_len, at line 93 of TencentOS-tiny/bin_file_download_through_module.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ota_http_response_line_parse passes to buf_len, at line 93 of TencentOS-tiny/bin_file_download_through_module.c, to overwrite the target buffer.

| | Source | Destination |
|------|---|---|
| File | TencentOS-tiny/bin_file_download_through_module.c | TencentOS-tiny/bin_file_download_through_module.c |

| | | |
|--------|---------|---------|
| Line | 99 | 99 |
| Object | buf_len | buf_len |

Code Snippet

File Name TencentOS-tiny/bin_file_download_through_module.c
 Method static http_parse_status_t ota_http_response_line_parse(int fd, char *line_buf, size_t buf_len)

```
....
99.     memset(line_buf, 0, buf_len);
```

Buffer Overflow boundcpy WrongSizeParam\Path 15:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=41 |
| Status | New |

The size of the buffer used by ota_url_parse in len, at line 43 of TencentOS-tiny/bin_file_download_through_module.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ota_url_parse passes to len, at line 43 of TencentOS-tiny/bin_file_download_through_module.c, to overwrite the target buffer.

| | Source | Destination |
|--------|---|---|
| File | TencentOS-tiny/bin_file_download_through_module.c | TencentOS-tiny/bin_file_download_through_module.c |
| Line | 64 | 64 |
| Object | len | len |

Code Snippet

File Name TencentOS-tiny/bin_file_download_through_module.c
 Method static int ota_url_parse(char *url, char *host, uint16_t *port, char *file)

```
....
64.     strncpy(host, url, len);
```

Buffer Overflow boundcpy WrongSizeParam\Path 16:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=42 |
| Status | New |

The size of the buffer used by ota_url_parse in len, at line 43 of TencentOS-tiny/bin_file_download_through_module.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ota_url_parse passes to len, at line 43 of TencentOS-tiny/bin_file_download_through_module.c, to overwrite the target buffer.

| | Source | Destination |
|--------|---|---|
| File | TencentOS-tiny/bin_file_download_through_module.c | TencentOS-tiny/bin_file_download_through_module.c |
| Line | 74 | 74 |
| Object | len | len |

Code Snippet

File Name TencentOS-tiny/bin_file_download_through_module.c
 Method static int ota_url_parse(char *url, char *host, uint16_t *port, char *file)

```
....
74.         strncpy(the_port, url, 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=1050050&projectid=50040&pathid=199 |
| Status | New |

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

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

Code Snippet

File Name TencentOS-tiny/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=1050050&projectid=50040&pathid=200 |
| Status | New |

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

| | Source | Destination |
|--------|-----------------------------------|-----------------------------------|
| File | <code>TencentOS-tiny/eap.c</code> | <code>TencentOS-tiny/eap.c</code> |
| Line | 1450 | 1450 |
| Object | <code>secret</code> | <code>lwip_md5_update</code> |

Code Snippet

File Name `TencentOS-tiny/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=1050050&projectid=50040&pathid=201 |
| Status | New |

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

| | Source | Destination |
|--------|-----------------------------------|-----------------------------------|
| File | <code>TencentOS-tiny/eap.c</code> | <code>TencentOS-tiny/eap.c</code> |
| Line | 1877 | 1877 |
| Object | <code>secret_len</code> | <code>lwip_md5_update</code> |

Code Snippet

File Name `TencentOS-tiny/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=1050050&projectid=50040&pathid=201 |

[40&pathid=202](#)

Status New

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

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

Code Snippet

File Name TencentOS-tiny/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=1050050&projectid=50040&pathid=203>

Status New

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

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

Code Snippet

File Name TencentOS-tiny/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=1050050&projectid=50040&pathid=204>

Status New

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

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

Code Snippet

File Name TencentOS-tiny/eap.c
Method pncrypt_setkey(int timeoffs)

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

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=1050050&projectid=50040&pathid=25 |
| Status | New |

The pointer opa_table at TencentOS-tiny/lv_draw_img.c in line 149 is being used after it has been freed.

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 187 | 187 |
| Object | opa_table | opa_table |

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c
Method lv_opa_t lv_img_buf_get_px_alpha(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y)

```
....
187.          return opa_table[px_opa];
```

Buffer Overflow AddressOfLocalVarReturned\Path 2:

Severity Medium

| | |
|----------------|---|
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=26 |
| Status | New |

The pointer opa_table at TencentOS-tiny/lv_draw_img.c in line 149 is being used after it has been freed.

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 197 | 197 |
| Object | opa_table | opa_table |

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c
Method lv_opa_t lv_img_buf_get_px_alpha(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y)

```
....
197.         return opa_table[px_opa];
```

Divide By Zero

Query Path:

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

Description

Divide By Zero\Path 1:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=21 |
| Status | New |

The application performs an illegal operation in color_mix_2_alpha, in TencentOS-tiny/lv_draw_basic.c. In line 660, the program attempts to divide by alpha_res, which might be evaluate to 0 (zero) at time of division. This value could be a hard-coded zero value, or received from external, untrusted input alpha_res in color_mix_2_alpha of TencentOS-tiny/lv_draw_basic.c, at line 660.

| | Source | Destination |
|--------|--------------------------------|--------------------------------|
| File | TencentOS-tiny/lv_draw_basic.c | TencentOS-tiny/lv_draw_basic.c |
| Line | 699 | 699 |
| Object | alpha_res | alpha_res |

Code Snippet

File Name TencentOS-tiny/lv_draw_basic.c
Method static inline lv_color_t color_mix_2_alpha(lv_color_t bg_color, lv_opa_t bg_opa, lv_color_t fg_color, lv_opa_t fg_opa)

```
....
699.          lv_opa_t ratio = (uint16_t)((uint16_t)fg_opa * 255) /
alpha_res;
```

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=1050050&projectid=50040&pathid=43 |
| Status | New |

The function pl in TencentOS-tiny/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 | TencentOS-tiny/eap.c | TencentOS-tiny/eap.c |
| Line | 1211 | 1211 |
| Object | pl | pl |

Code Snippet

File Name TencentOS-tiny/eap.c
Method name_of_pn_file()

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

Memory Leak

Query Path:

CPP\Cx\CPP Medium Threat\Memory Leak Version:1

[Categories](#)

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

[Description](#)

Memory Leak\Path 1:

| | |
|----------------|---|
| Severity | Medium |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=155 |
| Status | New |

| | Source | Destination |
|------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |

| | | |
|--------|------|------|
| Line | 438 | 438 |
| Object | keys | keys |

Code Snippet

File Name TencentOS-tiny/iniparser.c

Method char ** iniparser_getseckey(dictionary * d, char * s)

```
....
438.         keys = (char**) malloc(nkeys * sizeof(char*));
```

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=1050050&projectid=50040&pathid=56>

Status New

| | Source | Destination |
|--------|----------------------|----------------------|
| File | TencentOS-tiny/can.c | TencentOS-tiny/can.c |
| Line | 143 | 143 |
| Object | mb | mb |

Code Snippet

File Name TencentOS-tiny/can.c

Method void CAN_SetRxFilterMask(uint32_t instance, uint32_t mb, uint32_t mask)

```
....
143.         CANBase[instance]->RXIMR[mb] = CAN_ID_EXT(mask);
```

Unchecked Array Index\Path 2:

Severity Low

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=57>

Status New

| | Source | Destination |
|------|----------------------|----------------------|
| File | TencentOS-tiny/can.c | TencentOS-tiny/can.c |

| | | |
|--------|-----|-----|
| Line | 147 | 147 |
| Object | mb | mb |

Code Snippet

File Name TencentOS-tiny/can.c

Method void CAN_SetRxFilterMask(uint32_t instance, uint32_t mb, uint32_t mask)

```
....  
147.          CANBase[instance]->RXIMR[mb] = CAN_ID_STD(mask);
```

Unchecked Array Index\Path 3:

Severity Low

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=58>

Status New

| | Source | Destination |
|--------|----------------------|----------------------|
| File | TencentOS-tiny/can.c | TencentOS-tiny/can.c |
| Line | 378 | 378 |
| Object | instance | instance |

Code Snippet

File Name TencentOS-tiny/can.c

Method void CAN_CallbackInstall(uint32_t instance, CAN_CallbackType AppCBFun)

```
....  
378.          CAN_CallbackTable[instance] = AppCBFun;
```

Unchecked Array Index\Path 4:

Severity Low

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=59>

Status New

| | Source | Destination |
|--------|--------------------------|--------------------------|
| File | TencentOS-tiny/fsl_csi.c | TencentOS-tiny/fsl_csi.c |
| Line | 636 | 636 |
| Object | instance | instance |

Code Snippet

File Name TencentOS-tiny/fsl_csi.c

Method status_t CSI_TransferCreateHandle(CSI_Type *base,

```
....  
636.         s_csiHandle[instance] = handle;
```

Unchecked Array Index\Path 5:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=60 |
| Status | New |

| | Source | Destination |
|--------|---------------------------|---------------------------|
| File | TencentOS-tiny/fsl_pint.c | TencentOS-tiny/fsl_pint.c |
| Line | 131 | 131 |
| Object | intr | intr |

Code Snippet

File Name TencentOS-tiny/fsl_pint.c
Method void PINT_PinInterruptConfig(PINT_Type *base, pint_pin_int_t intr, pint_pin_enable_t enable, pint_cb_t callback)

```
....  
131.         s_pintCallback[intr] = callback;
```

Unchecked Array Index\Path 6:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=61 |
| Status | New |

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 453 | 453 |
| Object | i | i |

Code Snippet

File Name TencentOS-tiny/iniparser.c
Method char ** iniparser_getseckey(dictionary * d, char * s)

```
....  
453.         keys[i] = d->key[j];
```

Unchecked Array Index\Path 7:

| | |
|--------------|-----------|
| Severity | Low |
| Result State | To Verify |

| | |
|----------------|---|
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=62 |
| Status | New |

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 231 | 231 |
| Object | px | px |

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c

Method void lv_img_buf_set_px_color(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y, lv_color_t c)

```
....
231.          buf_u8[px]  = buf_u8[px] & ~(1 << (7 - bit));
```

Unchecked Array Index\Path 8:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=63 |
| Status | New |

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 232 | 232 |
| Object | px | px |

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c

Method void lv_img_buf_set_px_color(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y, lv_color_t c)

```
....
232.          buf_u8[px]  = buf_u8[px] | ((c.full & 0x1) << (7 - bit));
```

Unchecked Array Index\Path 9:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=64 |
| Status | New |

| Source | Destination |
|--------|-------------|
|--------|-------------|

| | | |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 240 | 240 |
| Object | px | px |

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c
Method void lv_img_buf_set_px_color(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y, lv_color_t c)

```
....
240.          buf_u8[px] = buf_u8[px] & ~(3 << (6 - bit));
```

Unchecked Array Index\Path 10:

Severity Low
Result State To Verify
Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=65>
Status New

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 241 | 241 |
| Object | px | px |

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c
Method void lv_img_buf_set_px_color(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y, lv_color_t c)

```
....
241.          buf_u8[px] = buf_u8[px] | ((c.full & 0x3) << (6 - bit));
```

Unchecked Array Index\Path 11:

Severity Low
Result State To Verify
Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=66>
Status New

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 248 | 248 |
| Object | px | px |

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c
Method void lv_img_buf_set_px_color(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y, lv_color_t c)

```
....
248.          buf_u8[px]  = buf_u8[px] & ~(0xF << (4 - bit));
```

Unchecked Array Index\Path 12:

Severity Low
Result State To Verify
Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=67>
Status New

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 249 | 249 |
| Object | px | px |

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c
Method void lv_img_buf_set_px_color(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y, lv_color_t c)

```
....
249.          buf_u8[px]  = buf_u8[px] | ((c.full & 0xF) << (4 - bit));
```

Unchecked Array Index\Path 13:

Severity Low
Result State To Verify
Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=68>
Status New

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 253 | 253 |
| Object | px | px |

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c
Method void lv_img_buf_set_px_color(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y, lv_color_t c)

```
....
253.          buf_u8[px]  = c.full;
```

Unchecked Array Index\Path 14:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=69 |
| Status | New |

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 277 | 277 |
| Object | px | px |

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c
Method void lv_img_buf_set_px_alpha(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y, lv_opa_t opa)

```
....  
277.          buf_u8[px]  = buf_u8[px] & ~(1 << (7 - bit));
```

Unchecked Array Index\Path 15:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=70 |
| Status | New |

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 278 | 278 |
| Object | px | px |

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c
Method void lv_img_buf_set_px_alpha(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y, lv_opa_t opa)

```
....  
278.          buf_u8[px]  = buf_u8[px] | ((opa & 0x1) << (7 - bit));
```

Unchecked Array Index\Path 16:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=71 |

| | |
|--------|-----|
| Status | New |
|--------|-----|

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 284 | 284 |
| Object | px | px |

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c

Method void lv_img_buf_set_px_alpha(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y, lv_opa_t opa)

```
....  
284.          buf_u8[px]  = buf_u8[px] & ~(3 << (6 - bit));
```

Unchecked Array Index\Path 17:

Severity Low

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=72>

Status New

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 285 | 285 |
| Object | px | px |

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c

Method void lv_img_buf_set_px_alpha(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y, lv_opa_t opa)

```
....  
285.          buf_u8[px]  = buf_u8[px] | ((opa & 0x3) << (6 - bit));
```

Unchecked Array Index\Path 18:

Severity Low

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=73>

Status New

| | Source | Destination |
|------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 292 | 292 |

| | | |
|--------|----|----|
| Object | px | px |
|--------|----|----|

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c

Method void lv_img_buf_set_px_alpha(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y, lv_opa_t opa)

```
....
292.          buf_u8[px]  = buf_u8[px] & ~(0xF << (4 - bit));
```

Unchecked Array Index\Path 19:

Severity Low

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=74>

Status New

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 293 | 293 |
| Object | px | px |

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c

Method void lv_img_buf_set_px_alpha(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y, lv_opa_t opa)

```
....
293.          buf_u8[px]  = buf_u8[px] | ((opa & 0xF) << (4 - bit));
```

Unchecked Array Index\Path 20:

Severity Low

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=75>

Status New

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/lv_draw_img.c | TencentOS-tiny/lv_draw_img.c |
| Line | 296 | 296 |
| Object | px | px |

Code Snippet

File Name TencentOS-tiny/lv_draw_img.c

Method void lv_img_buf_set_px_alpha(lv_img_dsc_t * dsc, lv_coord_t x, lv_coord_t y, lv_opa_t opa)

```
.....  
296.          buf_u8[px]  = opa;
```

Unchecked Array Index\Path 21:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=76 |
| Status | New |

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/RegionAU915.c | TencentOS-tiny/RegionAU915.c |
| Line | 738 | 738 |
| Object | cntChannelMask | cntChannelMask |

Code Snippet

File Name TencentOS-tiny/RegionAU915.c
Method uint8_t RegionAU915LinkAdrReq(LinkAdrReqParams_t* linkAdrReq, int8_t* drOut, int8_t* txPowOut, uint8_t* nbRepOut, uint8_t* nbBytesParsed)

```
.....  
738.          channelsMask[cntChannelMask] |= 0x00FF;
```

Unchecked Array Index\Path 22:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=77 |
| Status | New |

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/RegionAU915.c | TencentOS-tiny/RegionAU915.c |
| Line | 745 | 745 |
| Object | cntChannelMask | cntChannelMask |

Code Snippet

File Name TencentOS-tiny/RegionAU915.c
Method uint8_t RegionAU915LinkAdrReq(LinkAdrReqParams_t* linkAdrReq, int8_t* drOut, int8_t* txPowOut, uint8_t* nbRepOut, uint8_t* nbBytesParsed)

```
.....  
745.          channelsMask[cntChannelMask] |= 0xFF00;
```

Unchecked Array Index\Path 23:

| | |
|----------|-----|
| Severity | Low |
|----------|-----|

| | |
|----------------|---|
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=78 |
| Status | New |

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/RegionAU915.c | TencentOS-tiny/RegionAU915.c |
| Line | 758 | 758 |
| Object | cntChannelMask | cntChannelMask |

Code Snippet

File Name TencentOS-tiny/RegionAU915.c

Method uint8_t RegionAU915LinkAdrReq(LinkAdrReqParams_t* linkAdrReq, int8_t* drOut, int8_t* txPowOut, uint8_t* nbRepOut, uint8_t* nbBytesParsed)

```
....
758.                                channelsMask[cntChannelMask] &= 0xFF00;
```

Unchecked Array Index\Path 24:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=79 |
| Status | New |

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/RegionAU915.c | TencentOS-tiny/RegionAU915.c |
| Line | 765 | 765 |
| Object | cntChannelMask | cntChannelMask |

Code Snippet

File Name TencentOS-tiny/RegionAU915.c

Method uint8_t RegionAU915LinkAdrReq(LinkAdrReqParams_t* linkAdrReq, int8_t* drOut, int8_t* txPowOut, uint8_t* nbRepOut, uint8_t* nbBytesParsed)

```
....
765.                                channelsMask[cntChannelMask] &= 0x00FF;
```

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=1050050&projectid=50040&pathid=205 |
| Status | New |

| | Source | Destination |
|--------|----------------------|----------------------|
| File | TencentOS-tiny/eap.c | TencentOS-tiny/eap.c |
| Line | 1366 | 1366 |
| Object | BinaryExpr | BinaryExpr |

Code Snippet

File Name TencentOS-tiny/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,
```

Improper Resource Access Authorization\Path 2:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=206 |
| Status | New |

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 706 | 706 |
| Object | fprintf | fprintf |

Code Snippet

File Name TencentOS-tiny/iniparser.c

Method static line_status iniparser_line(

```
....  
706. fprintf(stderr, "iniparser: memory alloc error\n");
```

Improper Resource Access Authorization\Path 3:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=207 |
| Status | New |

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 733 | 733 |
| Object | fprintf | fprintf |

Code Snippet

File Name TencentOS-tiny/iniparser.c

Method static line_status iniparser_line(

```
....  
733.          fprintf(stderr, "iniparser: memory alloc error\n");
```

Improper Resource Access Authorization\Path 4:

Severity Low

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=208>

Status New

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 875 | 875 |
| Object | fprintf | fprintf |

Code Snippet

File Name TencentOS-tiny/iniparser.c

Method dictionary * iniparser_load(const char * ininame)

```
....  
875.          fprintf(stderr, "iniparser: cannot open %s\n", ininame);
```

Improper Resource Access Authorization\Path 5:

Severity Low

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=209>

Status New

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 931 | 931 |
| Object | fprintf | fprintf |

Code Snippet

File Name TencentOS-tiny/iniparser.c
Method dictionary * iniparser_load(const char * ininame)

```
....  
931.                fprintf(stderr,
```

Improper Resource Access Authorization\Path 6:

Severity Low
Result State To Verify
Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=210>
Status New

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 987 | 987 |
| Object | fprintf | fprintf |

Code Snippet

File Name TencentOS-tiny/iniparser.c
Method dictionary * iniparser_load(const char * ininame)

```
....  
987.                fprintf(stderr,
```

Improper Resource Access Authorization\Path 7:

Severity Low
Result State To Verify
Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=211>
Status New

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 1036 | 1036 |
| Object | fprintf | fprintf |

Code Snippet

File Name TencentOS-tiny/iniparser.c
Method dictionary * iniparser_load(const char * ininame)

```
....  
1036.               fprintf(stderr,
```

Improper Resource Access Authorization\Path 8:

Severity Low

| | |
|----------------|---|
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=212 |
| Status | New |

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 1049 | 1049 |
| Object | fprintf | fprintf |

Code Snippet

File Name TencentOS-tiny/iniparser.c

Method dictionary * iniparser_load(const char * ininame)

```
....  
1049.                fprintf(stderr, "iniparser: syntax error in %s  
(%d):\n",
```

Improper Resource Access Authorization\Path 9:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=213 |
| Status | New |

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 1052 | 1052 |
| Object | fprintf | fprintf |

Code Snippet

File Name TencentOS-tiny/iniparser.c

Method dictionary * iniparser_load(const char * ininame)

```
....  
1052.                fprintf(stderr, "-> %s\n", full_line);
```

Improper Resource Access Authorization\Path 10:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=214 |
| Status | New |

| Source | Destination |
|--------|-------------|
|--------|-------------|

| | | |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 1070 | 1070 |
| Object | fprintf | fprintf |

Code Snippet

File Name TencentOS-tiny/iniparser.c
Method dictionary * iniparser_load(const char * ininame)

```
....
1070.                fprintf(stderr, "iniparser: memory allocation
failure\n");
```

Unchecked Return Value

Query Path:

CPP\Cx\CPP Low Visibility\Unchecked Return Value Version:1

Categories

NIST SP 800-53: SI-11 Error Handling (P2)

Description

Unchecked Return Value\Path 1:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=10 |
| Status | New |

The application_entry method calls the sprintf function, at line 220 of TencentOS-tiny/bin_file_download_through_module.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 | TencentOS-tiny/bin_file_download_through_module.c | TencentOS-tiny/bin_file_download_through_module.c |
| Line | 243 | 243 |
| Object | sprintf | sprintf |

Code Snippet

File Name TencentOS-tiny/bin_file_download_through_module.c
Method void application_entry(void *arg)

```
....
243.                sprintf(port, "%d", port_num);
```

Unchecked Return Value\Path 2:

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

| | |
|--------|--|
| | BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=11 |
| Status | New |

The ota_http_request_construct method calls the snprintf function, at line 88 of TencentOS-tiny/bin_file_download_through_module.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 | TencentOS-tiny/bin_file_download_through_module.c | TencentOS-tiny/bin_file_download_through_module.c |
| Line | 90 | 90 |
| Object | snprintf | snprintf |

Code Snippet

File Name TencentOS-tiny/bin_file_download_through_module.c
 Method static void ota_http_request_construct(char *http_request, char *host, uint16_t port, char *file)

```
....
90.      snprintf(http_request, HTTP_REQUEST_MAX, HTTP_REQUEST_TEMPLATE,
file, host, port);
```

Unchecked Return Value\Path 3:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=12 |
| Status | New |

The CJSON_PUBLIC method calls the sprintf function, at line 125 of TencentOS-tiny/cJSON.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 | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 128 | 128 |
| Object | sprintf | sprintf |

Code Snippet

File Name TencentOS-tiny/cJSON.c
 Method CJSON_PUBLIC(const char*) cJSON_Version(void)

```
....
128.      sprintf(version, "%i.%i.%i", CJSON_VERSION_MAJOR,
CJSON_VERSION_MINOR, CJSON_VERSION_PATCH);
```

Unchecked Return Value\Path 4:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=13 |
| Status | New |

The `print_string_ptr` method calls the `sprintf` function, at line 906 of `TencentOS-tiny/cJSON.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 | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 1015 | 1015 |
| Object | sprintf | sprintf |

Code Snippet

File Name TencentOS-tiny/cJSON.c
Method static cJSON_bool print_string_ptr(const unsigned char * const input, printbuffer * const output_buffer)

```
....  
1015.                sprintf((char*)output_pointer, "u%04x",  
*input_pointer);
```

Unchecked Return Value\Path 5:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=14 |
| Status | New |

The `iniparser_dumpsection_ini` method calls the `snprintf` function, at line 316 of `TencentOS-tiny/iniparser.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 | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 336 | 336 |
| Object | snprintf | snprintf |

Code Snippet

File Name TencentOS-tiny/iniparser.c
Method void iniparser_dumpsection_ini(dictionary * d, char * s, FILE *f)

```
....  
336.                snprintf(keym, secsize, "%s:", s);
```

Unchecked Return Value\Path 6:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=15 |
| Status | New |

The `iniparser_getsecnkeys` method calls the `snprintf` function, at line 377 of `TencentOS-tiny/iniparser.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 | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 391 | 391 |
| Object | snprintf | snprintf |

Code Snippet

File Name TencentOS-tiny/iniparser.c
Method `int iniparser_getsecnkeys(dictionary * d, char * s)`

```
....  
391.      snprintf(keym, secksize, "%s:", s);
```

Unchecked Return Value\Path 7:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=16 |
| Status | New |

The `iniparser_getseckey` method calls the `snprintf` function, at line 421 of `TencentOS-tiny/iniparser.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 | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 442 | 442 |
| Object | snprintf | snprintf |

Code Snippet

File Name TencentOS-tiny/iniparser.c
Method `char ** iniparser_getseckey(dictionary * d, char * s)`

```
....  
442.      snprintf(keym, secksize, "%s:", s);
```

Unchecked Return Value\Path 8:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=17 |
| Status | New |

The `iniparser_load` method calls the `snprintf` function, at line 850 of `TencentOS-tiny/iniparser.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 | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 1041 | 1041 |
| Object | snprintf | snprintf |

Code Snippet

File Name TencentOS-tiny/iniparser.c
Method dictionary * iniparser_load(const char * ininame)

```
....  
1041.             snprintf(seckey, seckey_size, "%s:%s", section,  
key);
```

Unchecked Return Value\Path 9:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=18 |
| Status | New |

The `iniparser_getseckey` method calls the `keys` function, at line 421 of `TencentOS-tiny/iniparser.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 | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 438 | 438 |
| Object | keys | keys |

Code Snippet

File Name TencentOS-tiny/iniparser.c
Method char ** iniparser_getseckey(dictionary * d, char * s)

```
....  
438.             keys = (char**) malloc(nkeys * sizeof(char*));
```


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=1050050&projectid=50040&pathid=44 |
| Status | New |

The variable declared in null at TencentOS-tiny/cJSON.c in line 1042 is not initialized when it is used by content at TencentOS-tiny/cJSON.c in line 1042.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 1046 | 1044 |
| Object | null | content |

Code Snippet

File Name TencentOS-tiny/cJSON.c
Method static parse_buffer *buffer_skip_whitespace(parse_buffer * const buffer)

```

....
1046.         return NULL;
....
1044.         if ((buffer == NULL) || (buffer->content == NULL))

```

NULL Pointer Dereference\Path 2:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=45 |
| Status | New |

The variable declared in null at TencentOS-tiny/cJSON.c in line 1068 is not initialized when it is used by content at TencentOS-tiny/cJSON.c in line 1042.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 1072 | 1044 |
| Object | null | content |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method static parse_buffer *skip_utf8_bom(parse_buffer * const buffer)

```
....
1072.         return NULL;
```

File Name TencentOS-tiny/cJSON.c

Method static parse_buffer *buffer_skip_whitespace(parse_buffer * const buffer)

```
....
1044.         if ((buffer == NULL) || (buffer->content == NULL))
```

NULL Pointer Dereference\Path 3:

Severity Low

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=46>

Status New

The variable declared in newitem at TencentOS-tiny/cJSON.c in line 2710 is not initialized when it is used by valueint at TencentOS-tiny/cJSON.c in line 2710.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 2712 | 2730 |
| Object | newitem | valueint |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method cJSON_PUBLIC(cJSON *) cJSON_Duplicate(const cJSON *item, cJSON_bool recurse)

```
....
2712.         cJSON *newitem = NULL;
....
2730.         newitem->valueint = item->valueint;
```

NULL Pointer Dereference\Path 4:

Severity Low

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=47>

Status New

The variable declared in newitem at TencentOS-tiny/cJSON.c in line 2710 is not initialized when it is used by type at TencentOS-tiny/cJSON.c in line 2710.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 2712 | 2729 |
| Object | newitem | type |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method cJSON_PUBLIC(cJSON *) cJSON_Duplicate(const cJSON *item, cJSON_bool recurse)

```
....  
2712.      cJSON *newitem = NULL;  
....  
2729.      newitem->type = item->type & (~cJSON_IsReference);
```

NULL Pointer Dereference\Path 5:

Severity Low

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=48>

Status New

The variable declared in newitem at TencentOS-tiny/cJSON.c in line 2710 is not initialized when it is used by valuedouble at TencentOS-tiny/cJSON.c in line 2710.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 2712 | 2731 |
| Object | newitem | valuedouble |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method cJSON_PUBLIC(cJSON *) cJSON_Duplicate(const cJSON *item, cJSON_bool recurse)

```
....  
2712.      cJSON *newitem = NULL;  
....  
2731.      newitem->valuedouble = item->valuedouble;
```

NULL Pointer Dereference\Path 6:

Severity Low

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=49>

Status New

The variable declared in newitem at TencentOS-tiny/cJSON.c in line 2710 is not initialized when it is used by valuestring at TencentOS-tiny/cJSON.c in line 2710.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 2712 | 2734 |
| Object | newitem | valuestring |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method cJSON_PUBLIC(cJSON *) cJSON_Duplicate(const cJSON *item, cJSON_bool recurse)

```
....
2712.         cJSON *newitem = NULL;
....
2734.         newitem->valuestring = (char*)cJSON_strdup((unsigned
char*)item->valuestring, &global_hooks);
```

NULL Pointer Dereference\Path 7:

Severity Low

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=50>

Status New

The variable declared in newitem at TencentOS-tiny/cJSON.c in line 2710 is not initialized when it is used by valuestring at TencentOS-tiny/cJSON.c in line 2710.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 2712 | 2735 |
| Object | newitem | valuestring |

Code Snippet

File Name TencentOS-tiny/cJSON.c

Method cJSON_PUBLIC(cJSON *) cJSON_Duplicate(const cJSON *item, cJSON_bool recurse)

```
....
2712.         cJSON *newitem = NULL;
....
2735.         if (!newitem->valuestring)
```

NULL Pointer Dereference\Path 8:

Severity Low

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=50>

| | |
|--------|---|
| Status | 40&pathid=51 New |
|--------|---|

The variable declared in newitem at TencentOS-tiny/cJSON.c in line 2710 is not initialized when it is used by string at TencentOS-tiny/cJSON.c in line 2710.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 2712 | 2742 |
| Object | newitem | string |

Code Snippet

File Name TencentOS-tiny/cJSON.c
Method cJSON_PUBLIC(cJSON *) cJSON_Duplicate(const cJSON *item, cJSON_bool recurse)

```
....
2712.      cJSON *newitem = NULL;
....
2742.      newitem->string = (item->type&cJSON_StringIsConst) ?
item->string : (char*)cJSON_strdup((unsigned char*)item->string,
&global_hooks);
```

NULL Pointer Dereference\Path 9:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=52 |
| Status | New |

The variable declared in newitem at TencentOS-tiny/cJSON.c in line 2710 is not initialized when it is used by string at TencentOS-tiny/cJSON.c in line 2710.

| | Source | Destination |
|--------|------------------------|------------------------|
| File | TencentOS-tiny/cJSON.c | TencentOS-tiny/cJSON.c |
| Line | 2712 | 2743 |
| Object | newitem | string |

Code Snippet

File Name TencentOS-tiny/cJSON.c
Method cJSON_PUBLIC(cJSON *) cJSON_Duplicate(const cJSON *item, cJSON_bool recurse)

```
....
2712.      cJSON *newitem = NULL;
....
2743.      if (!newitem->string)
```

Potential Off by One Error in Loops

Query Path:

CPP\Cx\CPP Heuristic\Potential Off by One Error in Loops Version:1

Categories

PCI DSS v3.2: PCI DSS (3.2) - 6.5.1 - Injection flaws - particularly SQL injection

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

OWASP Top 10 2017: A1-Injection

Description

Potential Off by One Error in Loops\Path 1:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=22 |
| Status | New |

The buffer allocated by <= in TencentOS-tiny/RegionAU915.c at line 529 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/RegionAU915.c | TencentOS-tiny/RegionAU915.c |
| Line | 544 | 544 |
| Object | <= | <= |

Code Snippet

File Name TencentOS-tiny/RegionAU915.c
Method void RegionAU915ApplyCFList(ApplyCFListParams_t* applyCFList)

```
....
544.      for( uint8_t chMaskItr = 0, cntPayload = 0; chMaskItr <= 4;
chMaskItr++, cntPayload+=2 )
```

Potential Off by One Error in Loops\Path 2:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=23 |
| Status | New |

The buffer allocated by <= in TencentOS-tiny/RegionAU915.c at line 673 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

| | Source | Destination |
|--------|------------------------------|------------------------------|
| File | TencentOS-tiny/RegionAU915.c | TencentOS-tiny/RegionAU915.c |
| Line | 729 | 729 |
| Object | <= | <= |

Code Snippet

File Name TencentOS-tiny/RegionAU915.c

Method uint8_t RegionAU915LinkAdrReq(LinkAdrReqParams_t* linkAdrReq, int8_t* drOut, int8_t* txPowOut, uint8_t* nbRepOut, uint8_t* nbBytesParsed)

```
....
729.                for( uint8_t i = 0; i <= 7; i++ )
```

Potential Off by One Error in Loops\Path 3:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=24 |
| Status | New |

The buffer allocated by <= in TencentOS-tiny/timeutils.c at line 68 does not correctly account for the actual size of the value, resulting in an incorrect allocation that is off by one.

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/timeutils.c | TencentOS-tiny/timeutils.c |
| Line | 131 | 131 |
| Object | <= | <= |

Code Snippet

File Name TencentOS-tiny/timeutils.c
Method void timeutils_seconds_since_2000_to_struct_time(mp_uint_t t, timeutils_struct_time_t *tm) {

```
....
131.                for (month = 0; days_in_month[month] <= days; month++) {
```

Use of Sizeof On a Pointer Type

Query Path:

CPP\Cx\CPP Low Visibility\Use of Sizeof On a Pointer Type Version:1

[Description](#)

Use of Sizeof On a Pointer Type\Path 1:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=19 |
| Status | New |

| | Source | Destination |
|--------|----------------------------|----------------------------|
| File | TencentOS-tiny/iniparser.c | TencentOS-tiny/iniparser.c |
| Line | 438 | 438 |
| Object | sizeof | sizeof |

Code Snippet

File Name TencentOS-tiny/iniparser.c

Method char ** iniparser_getseckey(dictionary * d, char * s)

```
....
438.      keys = (char**) malloc(nkeys * sizeof(char*));
```

Use of Sizeof On a Pointer Type\Path 2:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=20 |
| Status | New |

| | Source | Destination |
|--------|----------------------------------|----------------------------------|
| File | TencentOS-tiny/rsa_alt_helpers.c | TencentOS-tiny/rsa_alt_helpers.c |
| Line | 86 | 86 |
| Object | sizeof | sizeof |

Code Snippet

File Name TencentOS-tiny/rsa_alt_helpers.c
Method int mbedtls_rsa_deduce_primes(mbedtls_mpi const *N,

```
....
86.      const size_t num_primes = sizeof( primes ) / sizeof( *primes );
```

Potential Precision Problem

Query Path:

CPP\Cx\CPP Buffer Overflow\Potential Precision Problem Version:0

Categories

NIST SP 800-53: SI-10 Information Input Validation (P1)

OWASP Top 10 2017: A1-Injection

Description

Potential Precision Problem\Path 1:

| | |
|----------------|---|
| Severity | Low |
| Result State | To Verify |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=53 |
| Status | New |

The size of the buffer used by ota_http_response_parse in "%s %d", at line 174 of TencentOS-tiny/bin_file_download_through_module.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ota_http_response_parse passes to "%s %d", at line 174 of TencentOS-tiny/bin_file_download_through_module.c, to overwrite the target buffer.

| | Source | Destination |
|------|---|---|
| File | TencentOS-tiny/bin_file_download_through_module.c | TencentOS-tiny/bin_file_download_through_module.c |

| | | |
|--------|---------|---------|
| Line | 190 | 190 |
| Object | "%s %d" | "%s %d" |

Code Snippet

File Name TencentOS-tiny/bin_file_download_through_module.c

Method static int ota_http_response_parse(int fd)

```
....
190.          sscanf(line_buf, "%s %d", header_field,
&header_value);
```

Potential Precision Problem\Path 2:

Severity Low

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=54>

Status New

The size of the buffer used by ota_http_response_parse in "%s %d", at line 174 of TencentOS-tiny/bin_file_download_through_module.c, is not properly verified before writing data to the buffer. This can enable a buffer overflow attack, using the source buffer that ota_http_response_parse passes to "%s %d", at line 174 of TencentOS-tiny/bin_file_download_through_module.c, to overwrite the target buffer.

| | Source | Destination |
|--------|---|---|
| File | TencentOS-tiny/bin_file_download_through_module.c | TencentOS-tiny/bin_file_download_through_module.c |
| Line | 197 | 197 |
| Object | "%s %d" | "%s %d" |

Code Snippet

File Name TencentOS-tiny/bin_file_download_through_module.c

Method static int ota_http_response_parse(int fd)

```
....
197.          sscanf(line_buf, "%s %d", header_field,
&header_value);
```

Sizeof Pointer Argument

Query Path:

CPP\Cx\CPP Low Visibility\Sizeof Pointer Argument Version:0

[Description](#)

Sizeof Pointer Argument\Path 1:

Severity Low

Result State To Verify

Online Results <http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050050&projectid=50040&pathid=55>

Status New

| | Source | Destination |
|--------|---------------------------------|---------------------------------|
| File | TencentOS-tiny/crypt_blowfish.c | TencentOS-tiny/crypt_blowfish.c |
| Line | 868 | 868 |
| Object | ai | sizeof |

Code Snippet

File Name TencentOS-tiny/crypt_blowfish.c

Method char *_crypt_blowfish_rn(const char *key, const char *setting,

```
....  
868.                !memcmp(ai, yi, sizeof(ai));
```

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=1050050&projectid=50040&pathid=215>

Status New

The open_pn_file method in TencentOS-tiny/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 | TencentOS-tiny/eap.c | TencentOS-tiny/eap.c |
| Line | 1231 | 1231 |
| Object | open | open |

Code Snippet

File Name TencentOS-tiny/eap.c

Method open_pn_file(modebits)

```
....  
1231.        fd = open(path, modebits, S_IRUSR | S_IWUSR);
```

Buffer Overflow LongString

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In its 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 (strlen(inputString, MAX_INPUT_SIZE) < sizeof(buffer))
    {
        strncpy(buffer, inputString, sizeof(buffer));
    }
}
```

Buffer Overflow StrcpyStrcat

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 its most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.

Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
 - Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
 - Consistently apply tests for the size of buffers.
 - Do not return variable addresses outside the scope of their variables.
-

Source Code Examples

Buffer Overflow OutOfBound

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In its most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.

Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
 - Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
 - Consistently apply tests for the size of buffers.
 - Do not return variable addresses outside the scope of their variables.
-

Source Code Examples

Divide By Zero

Risk

What might happen

When a program divides a number by zero, an exception will be raised. If this exception is not handled by the application, unexpected results may occur, including crashing the application. This can be considered a DoS (Denial of Service) attack, if an external user has control of the value of the denominator or can cause this error to occur.

Cause

How does it happen

The program receives an unexpected value, and uses it for division without filtering, validation, or verifying that the value is not zero. The application does not explicitly handle this error or prevent division by zero from occurring.

General Recommendations

How to avoid it

- Before dividing by an unknown value, validate the number and explicitly ensure it does not evaluate to zero.
 - Validate all untrusted input from all sources, in particular verifying that it is not zero before dividing with it.
 - Verify output of methods, calculations, dictionary lookups, and so on, and ensure it is not zero before dividing with the result.
 - Ensure divide-by-zero errors are caught and handled appropriately.
-

Source Code Examples

Java

Divide by Zero

```
public float getAverage(HttpServletRequest req) {  
    int total = Integer.parseInt(req.getParameter("total"));  
    int count = Integer.parseInt(req.getParameter("count"));  
  
    return total / count;  
}
```

Checked Division

```
public float getAverage(HttpServletRequest req) {  
    int total = Integer.parseInt(req.getParameter("total"));  
    int count = Integer.parseInt(req.getParameter("count"));
```

```
if (count > 0)
    return total / count;
else
    return 0;
}
```

Buffer Overflow AddressOfLocalVarReturned

Risk

What might happen

A use after free error will cause code to use an area of memory previously assigned with a specific value, which has since been freed and may have been overwritten by another value. This error will likely cause unexpected behavior, memory corruption and crash errors. In some cases where the freed and used section of memory is used to determine execution flow, and the error can be induced by an attacker, this may result in execution of malicious code.

Cause

How does it happen

Pointers to variables allow code to have an address with a set size to a dynamically allocated variable. Eventually, the pointer's destination may become free - either explicitly in code, such as when programmatically freeing this variable, or implicitly, such as when a local variable is returned - once it is returned, the variable's scope is released. Once freed, this memory will be re-used by the application, overwritten with new data. At this point, dereferencing this pointer will potentially resolve newly written and unexpected data.

General Recommendations

How to avoid it

- Do not return local variables or pointers
 - Review code to ensure no flow allows use of a pointer after it has been explicitly freed
-

Source Code Examples

CPP

Use of Variable after It was Freed

```
free(input);  
printf("%s", input);
```

Use of Pointer to Local Variable That Was Freed On Return

```
int* func1()  
{  
    int i;  
    i = 1;  
    return &i;  
}  
  
void func2()
```



```
{  
    int j;  
    j = 5;  
}  
  
//..  
int * i = func1();  
printf("%d\r\n", *i); // Output could be 1 or Segmentation Fault  
func2();  
printf("%d\r\n", *i); // Output is 5, which is j's value, as func2() overwrote data in  
the stack  
//..
```

Buffer Overflow boundcpy WrongSizeParam

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In its 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

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

```
int main(int argc, char* argv[])
{
    printf("%s", argv[1]); // Second parameter is not a formattable string
    return 0;
}
```

Failure to Release Memory Before Removing Last Reference ('Memory Leak')

Weakness ID: 401 (*Weakness Base*)

Status: Draft

Description

Description Summary

The software does not sufficiently track and release allocated memory after it has been used, which slowly consumes remaining memory.

Extended Description

This is often triggered by improper handling of malformed data or unexpectedly interrupted sessions.

Terminology Notes

"memory leak" has sometimes been used to describe other kinds of issues, e.g. for information leaks in which the contents of memory are inadvertently leaked (CVE-2003-0400 is one such example of this terminology conflict).

Time of Introduction

- Architecture and Design
- Implementation

Applicable Platforms

Languages

C

C++

Modes of Introduction

Memory leaks have two common and sometimes overlapping causes:

- Error conditions and other exceptional circumstances
- Confusion over which part of the program is responsible for freeing the memory

Common Consequences

| Scope | Effect |
|--------------|---|
| Availability | Most memory leaks result in general software reliability problems, but if an attacker can intentionally trigger a memory leak, the attacker might be able to launch a denial of service attack (by crashing or hanging the program) or take advantage of other unexpected program behavior resulting from a low memory condition. |

Likelihood of Exploit

Medium

Demonstrative Examples

Example 1

The following C function leaks a block of allocated memory if the call to read() fails to return the expected number of bytes:

(*Bad Code*)

Example Language: C

```
char* getBlock(int fd) {
char* buf = (char*) malloc(BLOCK_SIZE);
if (!buf) {
return NULL;
}
if (read(fd, buf, BLOCK_SIZE) != BLOCK_SIZE) {

return NULL;
}
```



```
return buf;
}
```

Example 2

Here the problem is that every time a connection is made, more memory is allocated. So if one just opened up more and more connections, eventually the machine would run out of memory.

(Bad Code)

Example Language: C

```
bar connection(){
foo = malloc(1024);
return foo;
}

endConnection(bar foo) {

free(foo);
}

int main() {

while(1) //thread 1
//On a connection
foo=connection(); //thread 2
//When the connection ends
endConnection(foo)
}
```

Observed Examples

| Reference | Description |
|-------------------------------|--|
| CVE-2005-3119 | Memory leak because function does not free() an element of a data structure. |
| CVE-2004-0427 | Memory leak when counter variable is not decremented. |
| CVE-2002-0574 | Memory leak when counter variable is not decremented. |
| CVE-2005-3181 | Kernel uses wrong function to release a data structure, preventing data from being properly tracked by other code. |
| CVE-2004-0222 | Memory leak via unknown manipulations as part of protocol test suite. |
| CVE-2001-0136 | Memory leak via a series of the same command. |

Potential Mitigations

Pre-design: Use a language or compiler that performs automatic bounds checking.

Phase: Architecture and Design

Use an abstraction library to abstract away risky APIs. Not a complete solution.

Pre-design through Build: The Boehm-Demers-Weiser Garbage Collector or valgrind can be used to detect leaks in code. This is not a complete solution as it is not 100% effective.

Relationships

| Nature | Type | ID | Name | View(s) this relationship pertains to |
|---------|----------------|-----|--|--|
| ChildOf | Weakness Class | 398 | Indicator of Poor Code Quality | Seven Pernicious Kingdoms (primary)700 |
| ChildOf | Category | 399 | Resource Management Errors | Development Concepts (primary)699 |
| ChildOf | Category | 633 | Weaknesses that Affect Memory | Resource-specific Weaknesses (primary)631 |
| ChildOf | Category | 730 | OWASP Top Ten 2004 Category A9 - Denial of Service | Weaknesses in OWASP Top Ten (2004) (primary)711 |
| ChildOf | Weakness Base | 772 | Missing Release of Resource after Effective | Research Concepts (primary)1000 |

| | | | | |
|-----------|----------------|-----|---|---|
| MemberOf | View | 630 | Lifetime Weaknesses Examined by SAMATE | Weaknesses Examined by SAMATE (primary) 630 Research Concepts1000 |
| CanFollow | Weakness Class | 390 | Detection of Error Condition Without Action | |

Relationship Notes

This is often a resultant weakness due to improper handling of malformed data or early termination of sessions.

Affected Resources

- Memory

Functional Areas

- Memory management

Taxonomy Mappings

| Mapped Taxonomy Name | Node ID | Fit | Mapped Node Name |
|-----------------------|---------|-------------------|----------------------------|
| PLOVER | | | Memory leak |
| 7 Pernicious Kingdoms | | | Memory Leak |
| CLASP | | | Failure to deallocate data |
| OWASP Top Ten 2004 | A9 | CWE More Specific | Denial of Service |

White Box Definitions

A weakness where the code path has:

1. start statement that allocates dynamically allocated memory resource
2. end statement that loses identity of the dynamically allocated memory resource creating situation where dynamically allocated memory resource is never relinquished

Where "loses" is defined through the following scenarios:

1. identity of the dynamic allocated memory resource never obtained
2. the statement assigns another value to the data element that stored the identity of the dynamically allocated memory resource and there are no aliases of that data element
3. identity of the dynamic allocated memory resource obtained but never passed on to function for memory resource release
4. the data element that stored the identity of the dynamically allocated resource has reached the end of its scope at the statement and there are no aliases of that data element

References

J. Whittaker and H. Thompson. "How to Break Software Security". Addison Wesley. 2003.

Content History

| Submissions | | | |
|-------------------|---|---------------|------------------|
| Submission Date | Submitter | Organization | Source |
| | PLOVER | | Externally Mined |
| Modifications | | | |
| Modification Date | Modifier | Organization | Source |
| 2008-07-01 | Eric Dalci | Cigital | External |
| | updated Time of Introduction | | |
| 2008-08-01 | | KDM Analytics | External |
| | added/updated white box definitions | | |
| 2008-08-15 | | Veracode | External |
| | Suggested OWASP Top Ten 2004 mapping | | |
| 2008-09-08 | CWE Content Team | MITRE | Internal |
| | updated Applicable Platforms, Common Consequences, Relationships, Other Notes, References, Relationship Notes, Taxonomy Mappings, Terminology Notes | | |
| 2008-10-14 | CWE Content Team | MITRE | Internal |
| | updated Description | | |
| 2009-03-10 | CWE Content Team | MITRE | Internal |
| | updated Other Notes | | |
| 2009-05-27 | CWE Content Team | MITRE | Internal |
| | updated Name | | |
| 2009-07-17 | KDM Analytics | | External |
| | Improved the White Box Definition | | |

| | | | | |
|----------------------|--|-------|----------|--|
| 2009-07-27 | CWE Content Team | MITRE | Internal | |
| | updated White Box Definitions | | | |
| 2009-10-29 | CWE Content Team | MITRE | Internal | |
| | updated Modes of Introduction, Other Notes | | | |
| 2010-02-16 | CWE Content Team | MITRE | Internal | |
| | updated Relationships | | | |
| Previous Entry Names | | | | |
| Change Date | Previous Entry Name | | | |
| 2008-04-11 | Memory Leak | | | |
| 2009-05-27 | Failure to Release Memory Before Removing Last Reference (aka 'Memory Leak') | | | |

[BACK TO TOP](#)

Use of Zero Initialized Pointer

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
 - Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
 - Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.
-

Source Code Examples

CPP

Explicit NULL Dereference

```
char * input = NULL;
printf("%s", input);
```

Implicit NULL Dereference

```
char * input;
printf("%s", input);
```

Java

Explicit Null Dereference

```
Object o = null;
out.println(o.getClass());
```



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 obsolete since it is relatively easy to break. These obsolete 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 function. These functions often return error codes in the result, or share other status codes with its caller. The application simply ignores this result value, losing this vital information.

General Recommendations

How to avoid it

- Always check the result of any called function that returns a value, and verify the result is an expected value.
 - Ensure the calling function responds to all possible return values.
 - Expect runtime errors and handle them gracefully. Explicitly define a mechanism for handling unexpected errors.
-

Source Code Examples

CPP

Unchecked Memory Allocation

```
buff = (char*) malloc(size);
strncpy(buff, source, size);
```

Safer Memory Allocation

```
buff = (char*) malloc(size+1);
if (buff==NULL) exit(1);

strncpy(buff, source, size);
buff[size] = '\0';
```


Use of sizeof() on a Pointer Type

Weakness ID: 467 (*Weakness Variant*)

Status: Draft

Description

Description Summary

The code calls sizeof() on a malloced pointer type, which always returns the wordsize/8. This can produce an unexpected result if the programmer intended to determine how much memory has been allocated.

Time of Introduction

Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

| Scope | Effect |
|-----------|---|
| Integrity | This error can often cause one to allocate a buffer that is much smaller than what is needed, leading to resultant weaknesses such as buffer overflows. |

Likelihood of Exploit

High

Demonstrative Examples

Example 1

Care should be taken to ensure sizeof returns the size of the data structure itself, and not the size of the pointer to the data structure.

In this example, sizeof(foo) returns the size of the pointer.

(Bad Code)

Example Languages: C and C++

```
double *foo;
...
foo = (double *)malloc(sizeof(foo));
```

In this example, sizeof(*foo) returns the size of the data structure and not the size of the pointer.

(Good Code)

Example Languages: C and C++

```
double *foo;
...
foo = (double *)malloc(sizeof(*foo));
```

Example 2

This example defines a fixed username and password. The AuthenticateUser() function is intended to accept a username and a password from an untrusted user, and check to ensure that it matches the username and password. If the username and password match, AuthenticateUser() is intended to indicate that authentication succeeded.

(Bad Code)

/ Ignore CWE-259 (hard-coded password) and CWE-309 (use of password system for authentication) for this example. */*

```
char *username = "admin";
char *pass = "password";

int AuthenticateUser(char *inUser, char *inPass) {
```

```
printf("Sizeof username = %d\n", sizeof(username));
printf("Sizeof pass = %d\n", sizeof(pass));

if (strcmp(username, inUser, sizeof(username))) {
printf("Auth failure of username using sizeof\n");
return(AUTH_FAIL);
}
/* Because of CWE-467, the sizeof returns 4 on many platforms and architectures. */
if (! strcmp(pass, inPass, sizeof(pass))) {
printf("Auth success of password using sizeof\n");
return(AUTH_SUCCESS);
}
else {
printf("Auth fail of password using sizeof\n");
return(AUTH_FAIL);
}
}

int main (int argc, char **argv)
{
int authResult;

if (argc < 3) {
ExitError("Usage: Provide a username and password");
}
authResult = AuthenticateUser(argv[1], argv[2]);
if (authResult != AUTH_SUCCESS) {
ExitError("Authentication failed");
}
else {
DoAuthenticatedTask(argv[1]);
}
}
```

In `AuthenticateUser()`, because `sizeof()` is applied to a parameter with an array type, the `sizeof()` call might return 4 on many modern architectures. As a result, the `strcmp()` call only checks the first four characters of the input password, resulting in a partial comparison (CWE-187), leading to improper authentication (CWE-287).

Because of the partial comparison, any of these passwords would still cause authentication to succeed for the "admin" user:

(Attack)

```
pass5
passABCDEFGH
passWORD
```

Because only 4 characters are checked, this significantly reduces the search space for an attacker, making brute force attacks more feasible.

The same problem also applies to the username, so values such as "adminXYZ" and "administrator" will succeed for the username.

Potential Mitigations

Phase: Implementation

Use expressions such as "`sizeof(*pointer)`" instead of "`sizeof(pointer)`", unless you intend to run `sizeof()` on a pointer type to gain some platform independence or if you are allocating a variable on the stack.

Other Notes

The use of `sizeof()` on a pointer can sometimes generate useful information. An obvious case is to find out the wordsize on a platform. More often than not, the appearance of `sizeof(pointer)` indicates a bug.

Weakness Ordinalities

| Ordinality | Description |
|------------|--|
| Primary | <i>(where the weakness exists independent of other weaknesses)</i> |

Relationships

| Nature | Type | ID | Name | View(s) this relationship pertains to |
|------------|----------------|-----|---|---|
| ChildOf | Category | 465 | Pointer Issues | Development Concepts (primary)699 |
| ChildOf | Weakness Class | 682 | Incorrect Calculation | Research Concepts (primary)1000 |
| ChildOf | Category | 737 | CERT C Secure Coding Section 03 - Expressions (EXP) | Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734 |
| ChildOf | Category | 740 | CERT C Secure Coding Section 06 - Arrays (ARR) | Weaknesses Addressed by the CERT C Secure Coding Standard734 |
| CanPrecede | Weakness Base | 131 | Incorrect Calculation of Buffer Size | Research Concepts1000 |

Taxonomy Mappings

| Mapped Taxonomy Name | Node ID | Fit | Mapped Node Name |
|----------------------|---------|-----|--|
| CLASP | | | Use of sizeof() on a pointer type |
| CERT C Secure Coding | ARR01-C | | Do not apply the sizeof operator to a pointer when taking the size of an array |
| CERT C Secure Coding | EXP01-C | | Do not take the size of a pointer to determine the size of the pointed-to type |

White Box Definitions

A weakness where code path has:

1. end statement that passes an identity of a dynamically allocated memory resource to a sizeof operator
2. start statement that allocates the dynamically allocated memory resource

References

Robert Seacord. "EXP01-A. Do not take the sizeof a pointer to determine the size of a type".
<https://www.securecoding.cert.org/confluence/display/seccode/EXP01-A.+Do+not+take+the+sizeof+a+pointer+to+determine+the+size+of+a+type>.

Content History

| Submissions | | | |
|-------------------|---|---------------|------------------|
| Submission Date | Submitter | Organization | Source |
| | CLASP | | Externally Mined |
| Modifications | | | |
| Modification Date | Modifier | Organization | Source |
| 2008-07-01 | Eric Dalci | Cigital | External |
| | updated Time of Introduction | | |
| 2008-08-01 | | KDM Analytics | External |
| | added/updated white box definitions | | |
| 2008-09-08 | CWE Content Team | MITRE | Internal |
| | updated Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities | | |
| 2008-11-24 | CWE Content Team | MITRE | Internal |
| | updated Relationships, Taxonomy Mappings | | |
| 2009-03-10 | CWE Content Team | MITRE | Internal |
| | updated Demonstrative Examples | | |
| 2009-12-28 | CWE Content Team | MITRE | Internal |
| | updated Demonstrative Examples | | |
| 2010-02-16 | CWE Content Team | MITRE | Internal |
| | updated Relationships | | |

[BACK TO TOP](#)

Potential Off by One Error in Loops

Risk

What might happen

An off by one error may result in overwriting or over-reading of unintended memory; in most cases, this can result in unexpected behavior and even application crashes. In other cases, where allocation can be controlled by an attacker, a combination of variable assignment and an off by one error can result in execution of malicious code.

Cause

How does it happen

Often when designating variables to memory, a calculation error may occur when determining size or length that is off by one.

For example in loops, when allocating an array of size 2, its cells are counted as 0,1 - therefore, if a For loop iterator on the array is incorrectly set with the start condition `i=0` and the continuation condition `i<=2`, three cells will be accessed instead of 2, and an attempt will be made to write or read cell [2], which was not originally allocated, resulting in potential corruption of memory outside the bounds of the originally assigned array.

Another example occurs when a null-byte terminated string, in the form of a character array, is copied without its terminating null-byte. Without the null-byte, the string representation is unterminated, resulting in certain functions to over-read memory as they expect the missing null terminator.

General Recommendations

How to avoid it

- Always ensure that a given iteration boundary is correct:
 - With array iterations, consider that arrays begin with cell 0 and end with cell `n-1`, for a size `n` array.
 - With character arrays and null-byte terminated string representations, consider that the null byte is required and should not be overwritten or ignored; ensure functions in use are not vulnerable to off-by-one, specifically for instances where null-bytes are automatically appended after the buffer, instead of in place of its last character.
 - Where possible, use safe functions that manage memory and are not prone to off-by-one errors.
-

Source Code Examples

CPP

Off-By-One in For Loop

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
for (int i = 0; i <= 5; i++)
{
    ptr[i] = i * 2 + 1; // ptr[5] will be set, but is out of bounds
}
```

```
}
```

Proper Iteration in For Loop

```
int *ptr;
ptr = (int*)malloc(5 * sizeof(int));
for (int i = 0; i < 5; i++)
{
    ptr[i] = i * 2 + 1; // ptr[0-4] are well defined
}
```

Off-By-One in strncat

```
strncat(buf, input, sizeof(buf) - strlen(buf)); // actual value should be sizeof(buf) -  
strlen(buf)-1 - this form will overwrite the terminating nullbyte
```

NULL Pointer Dereference

Risk

What might happen

A null pointer dereference is likely to cause a run-time exception, a crash, or other unexpected behavior.

Cause

How does it happen

Variables which are declared without being assigned will implicitly retain a null value until they are assigned. The null value can also be explicitly set to a variable, to ensure clear out its contents. Since null is not really a value, it may not have object variables and methods, and any attempt to access contents of a null object, instead of verifying it is set beforehand, will result in a null pointer dereference exception.

General Recommendations

How to avoid it

- For any variable that is created, ensure all logic flows between declaration and use assign a non-null value to the variable first.
 - Enforce null checks on any received variable or object before it is dereferenced, to ensure it does not contain a null assigned to it elsewhere.
 - Consider the need to assign null values in order to overwrite initialized variables. Consider reassigning or releasing these variables instead.
-

Source Code Examples

Potential Precision Problem

Risk

What might happen

Buffer overflow attacks, in their various forms, could allow an attacker to control certain areas of memory. Typically, this is used to overwrite data on the stack necessary for the program to function properly, such as code and memory addresses, though other forms of this attack exist. Exploiting this vulnerability can generally lead to system crashes, infinite loops, or even execution of arbitrary code.

Cause

How does it happen

Buffer Overflows can manifest in numerous different variations. In its most basic form, the attack controls a buffer, which is then copied to a smaller buffer without size verification. Because the attacker's source buffer is larger than the program's target buffer, the attacker's data overwrites whatever is next on the stack, allowing the attacker to control program structures.

Alternatively, the vulnerability could be the result of improper bounds checking; exposing internal memory addresses outside of their valid scope; allowing the attacker to control the size of the target buffer; or various other forms.

General Recommendations

How to avoid it

- Always perform proper bounds checking before copying buffers or strings.
 - Prefer to use safer functions and structures, e.g. safe string classes over `char*`, `strncpy` over `strcpy`, and so on.
 - Consistently apply tests for the size of buffers.
 - Do not return variable addresses outside the scope of their variables.
-

Source Code Examples

Use of sizeof() on a Pointer Type

Weakness ID: 467 (*Weakness Variant*)

Status: Draft

Description

Description Summary

The code calls sizeof() on a malloced pointer type, which always returns the wordsize/8. This can produce an unexpected result if the programmer intended to determine how much memory has been allocated.

Time of Introduction

Implementation

Applicable Platforms

Languages

C

C++

Common Consequences

| Scope | Effect |
|-----------|---|
| Integrity | This error can often cause one to allocate a buffer that is much smaller than what is needed, leading to resultant weaknesses such as buffer overflows. |

Likelihood of Exploit

High

Demonstrative Examples

Example 1

Care should be taken to ensure sizeof returns the size of the data structure itself, and not the size of the pointer to the data structure.

In this example, sizeof(foo) returns the size of the pointer.

(*Bad Code*)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(foo));
```

In this example, sizeof(*foo) returns the size of the data structure and not the size of the pointer.

(*Good Code*)

Example Languages: **C and C++**

```
double *foo;
...
foo = (double *)malloc(sizeof(*foo));
```

Example 2

This example defines a fixed username and password. The AuthenticateUser() function is intended to accept a username and a password from an untrusted user, and check to ensure that it matches the username and password. If the username and password match, AuthenticateUser() is intended to indicate that authentication succeeded.

(*Bad Code*)

/ Ignore CWE-259 (hard-coded password) and CWE-309 (use of password system for authentication) for this example. */*

```
char *username = "admin";
char *pass = "password";

int AuthenticateUser(char *inUser, char *inPass) {
```



```
printf("Sizeof username = %d\n", sizeof(username));
printf("Sizeof pass = %d\n", sizeof(pass));

if (strcmp(username, inUser, sizeof(username))) {
printf("Auth failure of username using sizeof\n");
return(AUTH_FAIL);
}
/* Because of CWE-467, the sizeof returns 4 on many platforms and architectures. */
if (! strcmp(pass, inPass, sizeof(pass))) {
printf("Auth success of password using sizeof\n");
return(AUTH_SUCCESS);
}
else {
printf("Auth fail of password using sizeof\n");
return(AUTH_FAIL);
}
}

int main (int argc, char **argv)
{
int authResult;

if (argc < 3) {
ExitError("Usage: Provide a username and password");
}
authResult = AuthenticateUser(argv[1], argv[2]);
if (authResult != AUTH_SUCCESS) {
ExitError("Authentication failed");
}
else {
DoAuthenticatedTask(argv[1]);
}
}
```

In `AuthenticateUser()`, because `sizeof()` is applied to a parameter with an array type, the `sizeof()` call might return 4 on many modern architectures. As a result, the `strcmp()` call only checks the first four characters of the input password, resulting in a partial comparison (CWE-187), leading to improper authentication (CWE-287).

Because of the partial comparison, any of these passwords would still cause authentication to succeed for the "admin" user:

(Attack)

```
pass5
passABCDEFGH
passWORD
```

Because only 4 characters are checked, this significantly reduces the search space for an attacker, making brute force attacks more feasible.

The same problem also applies to the username, so values such as "adminXYZ" and "administrator" will succeed for the username.

Potential Mitigations

Phase: Implementation

Use expressions such as "`sizeof(*pointer)`" instead of "`sizeof(pointer)`", unless you intend to run `sizeof()` on a pointer type to gain some platform independence or if you are allocating a variable on the stack.

Other Notes

The use of `sizeof()` on a pointer can sometimes generate useful information. An obvious case is to find out the wordsize on a platform. More often than not, the appearance of `sizeof(pointer)` indicates a bug.

Weakness Ordinalities

| Ordinality | Description |
|------------|--|
| Primary | <i>(where the weakness exists independent of other weaknesses)</i> |

Relationships

| Nature | Type | ID | Name | View(s) this relationship pertains to |
|------------|----------------|-----|---|---|
| ChildOf | Category | 465 | Pointer Issues | Development Concepts (primary)699 |
| ChildOf | Weakness Class | 682 | Incorrect Calculation | Research Concepts (primary)1000 |
| ChildOf | Category | 737 | CERT C Secure Coding Section 03 - Expressions (EXP) | Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734 |
| ChildOf | Category | 740 | CERT C Secure Coding Section 06 - Arrays (ARR) | Weaknesses Addressed by the CERT C Secure Coding Standard734 |
| CanPrecede | Weakness Base | 131 | Incorrect Calculation of Buffer Size | Research Concepts1000 |

Taxonomy Mappings

| Mapped Taxonomy Name | Node ID | Fit | Mapped Node Name |
|----------------------|---------|-----|--|
| CLASP | | | Use of sizeof() on a pointer type |
| CERT C Secure Coding | ARR01-C | | Do not apply the sizeof operator to a pointer when taking the size of an array |
| CERT C Secure Coding | EXP01-C | | Do not take the size of a pointer to determine the size of the pointed-to type |

White Box Definitions

A weakness where code path has:

1. end statement that passes an identity of a dynamically allocated memory resource to a sizeof operator
2. start statement that allocates the dynamically allocated memory resource

References

Robert Seacord. "EXP01-A. Do not take the sizeof a pointer to determine the size of a type".
<https://www.securecoding.cert.org/confluence/display/seccode/EXP01-A.+Do+not+take+the+sizeof+a+pointer+to+determine+the+size+of+a+type>.

Content History

| Submissions | | | |
|-------------------|---|---------------|------------------|
| Submission Date | Submitter | Organization | Source |
| | CLASP | | Externally Mined |
| Modifications | | | |
| Modification Date | Modifier | Organization | Source |
| 2008-07-01 | Eric Dalci updated Time of Introduction | Cigital | External |
| 2008-08-01 | added/updated white box definitions | KDM Analytics | External |
| 2008-09-08 | CWE Content Team updated Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities | MITRE | Internal |
| 2008-11-24 | CWE Content Team updated Relationships, Taxonomy Mappings | MITRE | Internal |
| 2009-03-10 | CWE Content Team updated Demonstrative Examples | MITRE | Internal |
| 2009-12-28 | CWE Content Team updated Demonstrative Examples | MITRE | Internal |
| 2010-02-16 | CWE Content Team updated Relationships | MITRE | Internal |

[BACK TO TOP](#)

Improper Validation of Array Index

Weakness ID: 129 (*Weakness Base*)

Status: Draft

Description

Description Summary

The product uses untrusted input when calculating or using an array index, but the product does not validate or incorrectly validates the index to ensure the index references a valid position within the array.

Alternate Terms

out-of-bounds array index

index-out-of-range

array index underflow

Time of Introduction

Implementation

Applicable Platforms

Languages

C: (*Often*)

C++: (*Often*)

Language-independent

Common Consequences

| Scope | Effect |
|--|--|
| Integrity Availability | Unchecked array indexing will very likely result in the corruption of relevant memory and perhaps instructions, leading to a crash, if the values are outside of the valid memory area. |
| Integrity | If the memory corrupted is data, rather than instructions, the system will continue to function with improper values. |
| Confidentiality Integrity | Unchecked array indexing can also trigger out-of-bounds read or write operations, or operations on the wrong objects; i.e., "buffer overflows" are not always the result. This may result in the exposure or modification of sensitive data. |
| Integrity | If the memory accessible by the attacker can be effectively controlled, it may be possible to execute arbitrary code, as with a standard buffer overflow and possibly without the use of large inputs if a precise index can be controlled. |
| Integrity Availability Confidentiality | A single fault could allow either an overflow (CWE-788) or underflow (CWE-786) of the array index. What happens next will depend on the type of operation being performed out of bounds, but can expose sensitive information, cause a system crash, or possibly lead to arbitrary code execution. |

Likelihood of Exploit

High

Detection Methods

Automated Static Analysis

This weakness can often be detected using automated static analysis tools. Many modern tools use data flow analysis or constraint-based techniques to minimize the number of false positives.

Automated static analysis generally does not account for environmental considerations when reporting out-of-bounds memory operations. This can make it difficult for users to determine which warnings should be investigated first. For example, an analysis tool might report array index errors that originate from command line arguments in a program that is not expected to run with setuid or other special privileges.

Effectiveness: High

This is not a perfect solution, since 100% accuracy and coverage are not feasible.

Automated Dynamic Analysis

This weakness can be detected using dynamic tools and techniques that interact with the software using large test suites with many diverse inputs, such as fuzz testing (fuzzing), robustness testing, and fault injection. The software's operation may slow down, but it should not become unstable, crash, or generate incorrect results.

Black Box

Black box methods might not get the needed code coverage within limited time constraints, and a dynamic test might not produce any noticeable side effects even if it is successful.

Demonstrative Examples

Example 1

The following C/C++ example retrieves the sizes of messages for a pop3 mail server. The message sizes are retrieved from a socket that returns in a buffer the message number and the message size, the message number (num) and size (size) are extracted from the buffer and the message size is placed into an array using the message number for the array index.

(Bad Code)

Example Language: C

```
/* capture the sizes of all messages */
int getsizes(int sock, int count, int *sizes) {
    ...
    char buf[BUFFER_SIZE];
    int ok;
    int num, size;

    // read values from socket and added to sizes array
    while ((ok = gen_recv(sock, buf, sizeof(buf))) == 0)
    {

        // continue read from socket until buf only contains '.'
        if (DOTLINE(buf))
            break;
        else if (sscanf(buf, "%d %d", &num, &size) == 2)
            sizes[num - 1] = size;
        }
    ...
}
```

In this example the message number retrieved from the buffer could be a value that is outside the allowable range of indices for the array and could possibly be a negative number. Without proper validation of the value to be used for the array index an array overflow could occur and could potentially lead to unauthorized access to memory addresses and system crashes. The value of the array index should be validated to ensure that it is within the allowable range of indices for the array as in the following code.

(Good Code)

Example Language: C

```
/* capture the sizes of all messages */
int getsizes(int sock, int count, int *sizes) {
    ...
    char buf[BUFFER_SIZE];
    int ok;
    int num, size;

    // read values from socket and added to sizes array
    while ((ok = gen_recv(sock, buf, sizeof(buf))) == 0)
    {

        // continue read from socket until buf only contains '.'
        if (DOTLINE(buf))
```

```
break;
else if (sscanf(buf, "%d %d", &num, &size) == 2) {
if (num > 0 && num <= (unsigned)count)
sizes[num - 1] = size;
else
/* warn about possible attempt to induce buffer overflow */
report(stderr, "Warning: ignoring bogus data for message sizes returned by server.\n");
}
}
...
}
```

Example 2

In the code snippet below, an unchecked integer value is used to reference an object in an array.

(Bad Code)

Example Language: Java

```
public String getValue(int index) {
return array[index];
}
```

If index is outside of the range of the array, this may result in an `ArrayIndexOutOfBoundsException` 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 cause the application to fail. The integer value used for the array index should be validated to ensure that it is within the allowable range of indices for the array as in the following code.

(Good Code)

Example Language: Java

// Method called from servlet to obtain product information

```
public String displayProductSummary(int index) {

String productSummary = new String("");
```

```
try {
String productSummary = getProductSummary(index);

} catch (Exception ex) {...}

return productSummary;
}

public String getProductSummary(int index) {
String productSummary = "";

if ((index >= 0) && (index < MAX_PRODUCTS)) {
productSummary = products[index];
}
else {
System.err.println("index is out of bounds");
throw new IndexOutOfBoundsException();
}

return productSummary;
}
```

An alternative in Java would be to use one of the collection objects such as `ArrayList` that will automatically generate an exception if an attempt is made to access an array index that is out of bounds.

(Good Code)

Example Language: Java

```
ArrayList productArray = new ArrayList(MAX_PRODUCTS);
...
try {
productSummary = (String) productArray.get(index);
} catch (IndexOutOfBoundsException ex) {...}
```

Observed Examples

| Reference | Description |
|-------------------------------|---|
| CVE-2005-0369 | large ID in packet used as array index |
| CVE-2001-1009 | negative array index as argument to POP LIST command |
| CVE-2003-0721 | Integer signedness error leads to negative array index |
| CVE-2004-1189 | product does not properly track a count and a maximum number, which can lead to resultant array index overflow. |
| CVE-2007-5756 | chain: device driver for packet-capturing software allows access to an unintended IOCTL with resultant array index error. |

Potential Mitigations

Phase: Architecture and Design

Strategies: Input Validation; Libraries or Frameworks

Use an input validation framework such as Struts or the OWASP ESAPI Validation API. If you use Struts, be mindful of weaknesses covered by the CWE-101 category.

Phase: Architecture and Design

For any security checks that are performed on the client side, ensure that these checks are duplicated on the server side, in order to avoid CWE-602. Attackers can bypass the client-side checks by modifying values after the checks have been performed, or by changing the client to remove the client-side checks entirely. Then, these modified values would be submitted to the server.

Even though client-side checks provide minimal benefits with respect to server-side security, they are still useful. First, they can support intrusion detection. If the server receives input that should have been rejected by the client, then it may be an indication of an attack. Second, client-side error-checking can provide helpful feedback to the user about the expectations for valid input. Third, there may be a reduction in server-side processing time for accidental input errors, although this is typically a small savings.

Phase: Requirements

Strategy: Language Selection

Use a language with features that can automatically mitigate or eliminate out-of-bounds indexing errors.

For example, Ada allows the programmer to constrain the values of a variable and languages such as Java and Ruby will allow the programmer to handle exceptions when an out-of-bounds index is accessed.

Phase: Implementation

Strategy: Input Validation

Assume all input is malicious. Use an "accept known good" input validation strategy (i.e., use a whitelist). Reject any input that does not strictly conform to specifications, or transform it into something that does. Use a blacklist to reject any unexpected inputs and detect potential attacks.

When accessing a user-controlled array index, use a stringent range of values that are within the target array. Make sure that you do not allow negative values to be used. That is, verify the minimum as well as the maximum of the range of acceptable values.

Phase: Implementation

Be especially careful to validate your input when you invoke code that crosses language boundaries, such as from an interpreted language to native code. This could create an unexpected interaction between the language boundaries. Ensure that you are not violating any of the expectations of the language with which you are interfacing. For example, even though Java may not be susceptible to buffer overflows, providing a large argument in a call to native code might trigger an overflow.

Weakness Ordinalities

| Ordinality | Description |
|------------|--|
| Resultant | The most common condition situation leading to unchecked array indexing is the use of loop index variables as buffer indexes. If the end condition for the loop is subject to a flaw, the index can grow or shrink unbounded, therefore causing a buffer overflow or underflow. Another common situation leading to this condition is the use of a function's return value, or the resulting value of a calculation directly as an index in to a buffer. |

Relationships

| Nature | Type | ID | Name | View(s) this relationship pertains to |
|------------|------------------|-----|--|--|
| ChildOf | Weakness Class | 20 | Improper Input Validation | Development Concepts (primary)699 Research Concepts (primary)1000 |
| ChildOf | Category | 189 | Numeric Errors | Development Concepts699 |
| ChildOf | Category | 633 | Weaknesses that Affect Memory | Resource-specific Weaknesses (primary)631 |
| ChildOf | Category | 738 | CERT C Secure Coding Section 04 - Integers (INT) | Weaknesses Addressed by the CERT C Secure Coding Standard (primary)734 |
| ChildOf | Category | 740 | CERT C Secure Coding Section 06 - Arrays (ARR) | Weaknesses Addressed by the CERT C Secure Coding Standard734 |
| ChildOf | Category | 802 | 2010 Top 25 - Risky Resource Management | Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800 |
| CanPrecede | Weakness Class | 119 | Failure to Constrain Operations within the Bounds of a Memory Buffer | Research Concepts1000 |
| CanPrecede | Weakness Variant | 789 | Uncontrolled Memory Allocation | Research Concepts1000 |
| PeerOf | Weakness Base | 124 | Buffer Underwrite ('Buffer Underflow') | Research Concepts1000 |

Theoretical Notes

An improperly validated array index might lead directly to the always-incorrect behavior of "access of array using out-of-bounds index."

Affected Resources

Memory

f Causal Nature

Explicit

Taxonomy Mappings

| Mapped Taxonomy Name | Node ID | Fit | Mapped Node Name |
|----------------------|---------|-----|---|
| CLASP | | | Unchecked array indexing |
| PLOVER | | | INDEX - Array index overflow |
| CERT C Secure Coding | ARR00-C | | Understand how arrays work |
| CERT C Secure Coding | ARR30-C | | Guarantee that array indices are within the valid range |
| CERT C Secure Coding | ARR38-C | | Do not add or subtract an integer to a pointer if the resulting value does not refer to a valid array element |
| CERT C Secure Coding | INT32-C | | Ensure that operations on signed integers do not result in overflow |

Related Attack Patterns

| CAPEC-ID | Attack Pattern Name | (CAPEC Version: 1.5) |
|---------------------|---------------------|----------------------|
| 100 | Overflow Buffers | |

References

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 5, "Array Indexing Errors" Page 144. 2nd Edition. Microsoft. 2002.

Content History

| Submissions | | | |
|----------------------|---|--------------|------------------|
| Submission Date | Submitter | Organization | Source |
| | CLASP | | Externally Mined |
| Modifications | | | |
| Modification Date | Modifier | Organization | Source |
| 2008-07-01 | Sean Eidemiller | Cigital | External |
| | added/updated demonstrative examples | | |
| 2008-09-08 | CWE Content Team | MITRE | Internal |
| | updated Alternate Terms, Applicable Platforms, Common Consequences, Relationships, Other Notes, Taxonomy Mappings, Weakness Ordinalities | | |
| 2008-11-24 | CWE Content Team | MITRE | Internal |
| | updated Relationships, Taxonomy Mappings | | |
| 2009-01-12 | CWE Content Team | MITRE | Internal |
| | updated Common Consequences | | |
| 2009-10-29 | CWE Content Team | MITRE | Internal |
| | updated Description, Name, Relationships | | |
| 2009-12-28 | CWE Content Team | MITRE | Internal |
| | updated Applicable Platforms, Common Consequences, Observed Examples, Other Notes, Potential Mitigations, Theoretical Notes, Weakness Ordinalities | | |
| 2010-02-16 | CWE Content Team | MITRE | Internal |
| | updated Applicable Platforms, Demonstrative Examples, Detection Factors, Likelihood of Exploit, Potential Mitigations, References, Related Attack Patterns, Relationships | | |
| 2010-04-05 | CWE Content Team | MITRE | Internal |
| | updated Related Attack Patterns | | |
| Previous Entry Names | | | |
| Change Date | Previous Entry Name | | |
| 2009-10-29 | Unchecked Array Indexing | | |

[BACK TO TOP](#)

Improper Access Control (Authorization)

Weakness ID: 285 (*Weakness Class*)

Status: Draft

Description

Description Summary

The software does not perform or incorrectly performs access control checks across all potential execution paths.

Extended Description

When access control checks are not applied consistently - or not at all - users are able to access data or perform actions that they should not be allowed to perform. This can lead to a wide range of problems, including information leaks, denial of service, and arbitrary code execution.

Alternate Terms

AuthZ:

"AuthZ" is typically used as an abbreviation of "authorization" within the web application security community. It is also distinct from "AuthC," which is an abbreviation of "authentication." The use of "Auth" as an abbreviation is discouraged, since it could be used for either authentication or authorization.

Time of Introduction

- Architecture and Design
- Implementation
- Operation

Applicable Platforms

Languages

Language-independent

Technology Classes

Web-Server: (*Often*)

Database-Server: (*Often*)

Modes of Introduction

A developer may introduce authorization weaknesses because of a lack of understanding about the underlying technologies. For example, a developer may assume that attackers cannot modify certain inputs such as headers or cookies.

Authorization weaknesses may arise when a single-user application is ported to a multi-user environment.

Common Consequences

| Scope | Effect |
|-----------------|---|
| Confidentiality | An attacker could read sensitive data, either by reading the data directly from a data store that is not properly restricted, or by accessing insufficiently-protected, privileged functionality to read the data. |
| Integrity | An attacker could modify sensitive data, either by writing the data directly to a data store that is not properly restricted, or by accessing insufficiently-protected, privileged functionality to write the data. |
| Integrity | An attacker could gain privileges by modifying or reading critical data directly, or by accessing insufficiently-protected, privileged functionality. |

Likelihood of Exploit

High

Detection Methods

Automated Static Analysis

Automated static analysis is useful for detecting commonly-used idioms for authorization. A tool may be able to analyze related configuration files, such as .htaccess in Apache web servers, or detect the usage of commonly-used authorization libraries.

Generally, automated static analysis tools have difficulty detecting custom authorization schemes. In addition, the software's design may include some functionality that is accessible to any user and does not require an authorization check; an automated technique that detects the absence of authorization may report false positives.

Effectiveness: Limited

Automated Dynamic Analysis

Automated dynamic analysis may find many or all possible interfaces that do not require authorization, but manual analysis is required to determine if the lack of authorization violates business logic

Manual Analysis

This weakness can be detected using tools and techniques that require manual (human) analysis, such as penetration testing, threat modeling, and interactive tools that allow the tester to record and modify an active session.

Specifically, manual static analysis is useful for evaluating the correctness of custom authorization mechanisms.

Effectiveness: Moderate

These may be more effective than strictly automated techniques. This is especially the case with weaknesses that are related to design and business rules. However, manual efforts might not achieve desired code coverage within limited time constraints.

Demonstrative Examples

Example 1

The following program could be part of a bulletin board system that allows users to send private messages to each other. This program intends to authenticate the user before deciding whether a private message should be displayed. Assume that `LookupMessageObject()` ensures that the `$id` argument is numeric, constructs a filename based on that id, and reads the message details from that file. Also assume that the program stores all private messages for all users in the same directory.

(Bad Code)

Example Language: Perl

```
sub DisplayPrivateMessage {
my($id) = @_ ;
my $Message = LookupMessageObject($id);
print "From: " . encodeHTML($Message->{from}) . "<br>\n";
print "Subject: " . encodeHTML($Message->{subject}) . "\n";
print "<hr>\n";
print "Body: " . encodeHTML($Message->{body}) . "\n";
}

my $q = new CGI;
# For purposes of this example, assume that CWE-309 and
# CWE-523 do not apply.
if (! AuthenticateUser($q->param('username'), $q->param('password'))) {
ExitError("invalid username or password");
}

my $id = $q->param('id');
DisplayPrivateMessage($id);
```

While the program properly exits if authentication fails, it does not ensure that the message is addressed to the user. As a result, an authenticated attacker could provide any arbitrary identifier and read private messages that were intended for other users. One way to avoid this problem would be to ensure that the "to" field in the message object matches the username of the authenticated user.

Observed Examples

| Reference | Description |
|-------------------------------|--|
| CVE-2009-3168 | Web application does not restrict access to admin scripts, allowing authenticated users to reset administrative passwords. |

| | |
|-------------------------------|---|
| CVE-2009-2960 | Web application does not restrict access to admin scripts, allowing authenticated users to modify passwords of other users. |
| CVE-2009-3597 | Web application stores database file under the web root with insufficient access control (CWE-219), allowing direct request. |
| CVE-2009-2282 | Terminal server does not check authorization for guest access. |
| CVE-2009-3230 | Database server does not use appropriate privileges for certain sensitive operations. |
| CVE-2009-2213 | Gateway uses default "Allow" configuration for its authorization settings. |
| CVE-2009-0034 | Chain: product does not properly interpret a configuration option for a system group, allowing users to gain privileges. |
| CVE-2008-6123 | Chain: SNMP product does not properly parse a configuration option for which hosts are allowed to connect, allowing unauthorized IP addresses to connect. |
| CVE-2008-5027 | System monitoring software allows users to bypass authorization by creating custom forms. |
| CVE-2008-7109 | Chain: reliance on client-side security (CWE-602) allows attackers to bypass authorization using a custom client. |
| CVE-2008-3424 | Chain: product does not properly handle wildcards in an authorization policy list, allowing unintended access. |
| CVE-2009-3781 | Content management system does not check access permissions for private files, allowing others to view those files. |
| CVE-2008-4577 | ACL-based protection mechanism treats negative access rights as if they are positive, allowing bypass of intended restrictions. |
| CVE-2008-6548 | Product does not check the ACL of a page accessed using an "include" directive, allowing attackers to read unauthorized files. |
| CVE-2007-2925 | Default ACL list for a DNS server does not set certain ACLs, allowing unauthorized DNS queries. |
| CVE-2006-6679 | Product relies on the X-Forwarded-For HTTP header for authorization, allowing unintended access by spoofing the header. |
| CVE-2005-3623 | OS kernel does not check for a certain privilege before setting ACLs for files. |
| CVE-2005-2801 | Chain: file-system code performs an incorrect comparison (CWE-697), preventing defaults ACLs from being properly applied. |
| CVE-2001-1155 | Chain: product does not properly check the result of a reverse DNS lookup because of operator precedence (CWE-783), allowing bypass of DNS-based access restrictions. |

Potential Mitigations

Phase: Architecture and Design

Divide your application into anonymous, normal, privileged, and administrative areas. Reduce the attack surface by carefully mapping roles with data and functionality. Use role-based access control (RBAC) to enforce the roles at the appropriate boundaries.

Note that this approach may not protect against horizontal authorization, i.e., it will not protect a user from attacking others with the same role.

Phase: Architecture and Design

Ensure that you perform access control checks related to your business logic. These checks may be different than the access control checks that you apply to more generic resources such as files, connections, processes, memory, and database records. For example, a database may restrict access for medical records to a specific database user, but each record might only be intended to be accessible to the patient and the patient's doctor.

Phase: Architecture and Design

Strategy: Libraries or Frameworks

Use a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness

easier to avoid.

For example, consider using authorization frameworks such as the JAAS Authorization Framework and the OWASP ESAPI Access Control feature.

Phase: Architecture and Design

For web applications, make sure that the access control mechanism is enforced correctly at the server side on every page. Users should not be able to access any unauthorized functionality or information by simply requesting direct access to that page.

One way to do this is to ensure that all pages containing sensitive information are not cached, and that all such pages restrict access to requests that are accompanied by an active and authenticated session token associated with a user who has the required permissions to access that page.

Phases: System Configuration; Installation

Use the access control capabilities of your operating system and server environment and define your access control lists accordingly. Use a "default deny" policy when defining these ACLs.

Relationships

| Nature | Type | ID | Name | View(s) this relationship pertains to |
|----------|------------------|-----|---|--|
| ChildOf | Category | 254 | Security Features | Seven Pernicious Kingdoms (primary)700 |
| ChildOf | Weakness Class | 284 | Access Control (Authorization) Issues | Development Concepts (primary)699 Research Concepts (primary)1000 |
| ChildOf | Category | 721 | OWASP Top Ten 2007 Category A10 - Failure to Restrict URL Access | Weaknesses in OWASP Top Ten (2007) (primary)629 |
| ChildOf | Category | 723 | OWASP Top Ten 2004 Category A2 - Broken Access Control | Weaknesses in OWASP Top Ten (2004) (primary)711 |
| ChildOf | Category | 753 | 2009 Top 25 - Porous Defenses | Weaknesses in the 2009 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)750 |
| ChildOf | Category | 803 | 2010 Top 25 - Porous Defenses | Weaknesses in the 2010 CWE/SANS Top 25 Most Dangerous Programming Errors (primary)800 |
| ParentOf | Weakness Variant | 219 | Sensitive Data Under Web Root | Research Concepts (primary)1000 |
| ParentOf | Weakness Base | 551 | Incorrect Behavior Order: Authorization Before Parsing and Canonicalization | Development Concepts (primary)699 Research Concepts1000 |
| ParentOf | Weakness Class | 638 | Failure to Use Complete Mediation | Research Concepts1000 |
| ParentOf | Weakness Base | 804 | Guessable CAPTCHA | Development Concepts (primary)699 Research Concepts (primary)1000 |

Taxonomy Mappings

| Mapped Taxonomy Name | Node ID | Fit | Mapped Node Name |
|-----------------------|---------|-------------------|--------------------------------|
| 7 Pernicious Kingdoms | | | Missing Access Control |
| OWASP Top Ten 2007 | A10 | CWE More Specific | Failure to Restrict URL Access |
| OWASP Top Ten 2004 | A2 | CWE More Specific | Broken Access Control |

Related Attack Patterns

| CAPEC-ID | Attack Pattern Name | (CAPEC Version: 1.5) |
|--------------------|--|----------------------|
| 1 | Accessing Functionality Not Properly Constrained by ACLs | |
| 13 | Subverting Environment Variable Values | |

| | |
|---------------------|---|
| 17 | Accessing, Modifying or Executing Executable Files |
| 87 | Forceful Browsing |
| 39 | Manipulating Opaque Client-based Data Tokens |
| 45 | Buffer Overflow via Symbolic Links |
| 51 | Poison Web Service Registry |
| 59 | Session Credential Falsification through Prediction |
| 60 | Reusing Session IDs (aka Session Replay) |
| 77 | Manipulating User-Controlled Variables |
| 76 | Manipulating Input to File System Calls |
| 104 | Cross Zone Scripting |

References

NIST. "Role Based Access Control and Role Based Security". <<http://csrc.nist.gov/groups/SNS/rbac/>>.

[REF-11] M. Howard and D. LeBlanc. "Writing Secure Code". Chapter 4, "Authorization" Page 114; Chapter 6, "Determining Appropriate Access Control" Page 171. 2nd Edition. Microsoft. 2002.

Content History

| Submissions | | | |
|----------------------|---|--------------|------------------|
| Submission Date | Submitter | Organization | Source |
| | 7 Pernicious Kingdoms | | Externally Mined |
| Modifications | | | |
| Modification Date | Modifier | Organization | Source |
| 2008-07-01 | Eric Dalci | Cigital | External |
| | updated Time of Introduction | | |
| 2008-08-15 | | Veracode | External |
| | Suggested OWASP Top Ten 2004 mapping | | |
| 2008-09-08 | CWE Content Team | MITRE | Internal |
| | updated Relationships, Other Notes, Taxonomy Mappings | | |
| 2009-01-12 | CWE Content Team | MITRE | Internal |
| | updated Common Consequences, Description, Likelihood of Exploit, Name, Other Notes, Potential Mitigations, References, Relationships | | |
| 2009-03-10 | CWE Content Team | MITRE | Internal |
| | updated Potential Mitigations | | |
| 2009-05-27 | CWE Content Team | MITRE | Internal |
| | updated Description, Related Attack Patterns | | |
| 2009-07-27 | CWE Content Team | MITRE | Internal |
| | updated Relationships | | |
| 2009-10-29 | CWE Content Team | MITRE | Internal |
| | updated Type | | |
| 2009-12-28 | CWE Content Team | MITRE | Internal |
| | updated Applicable Platforms, Common Consequences, Demonstrative Examples, Detection Factors, Modes of Introduction, Observed Examples, Relationships | | |
| 2010-02-16 | CWE Content Team | MITRE | Internal |
| | updated Alternate Terms, Detection Factors, Potential Mitigations, References, Relationships | | |
| 2010-04-05 | CWE Content Team | MITRE | Internal |
| | updated Potential Mitigations | | |
| Previous Entry Names | | | |
| Change Date | Previous Entry Name | | |
| 2009-01-12 | Missing or Inconsistent Access Control | | |

[BACK TO TOP](#)

TOCTOU

Risk

What might happen

At best, a Race Condition may cause errors in accuracy, overridden 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 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 |