

proxychains-ng Scan Report

| | |
|-----------------------|---|
| Project Name | proxychains-ng |
| Scan Start | Friday, June 21, 2024 10:48:23 PM |
| Preset | Checkmarx Default |
| Scan Time | 00h:01m:07s |
| Lines Of Code Scanned | 188 |
| Files Scanned | 1 |
| Report Creation Time | Friday, June 21, 2024 10:51:18 PM |
| Online Results | http://WIN-BA8RD5TJ8IG/CxWebClient/ViewerMain.aspx?scanid=1050057&projectid=50047 |
| Team | CxServer |
| Checkmarx Version | 8.7.0 |
| Scan Type | Full |
| Source Origin | LocalPath |
| Density | 5/1000 (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 | 0 | 0 |
| A2-Broken Authentication | App. Specific | EASY | COMMON | AVERAGE | SEVERE | App. Specific | 0 | 0 |
| A3-Sensitive Data Exposure | App. Specific | AVERAGE | WIDESPREAD | AVERAGE | SEVERE | App. Specific | 0 | 0 |
| 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 | 0 | 0 |
| 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 | 0 | 0 |
| A10-Unvalidated Redirects and Forwards | USERS BROWSERS | AVERAGE | WIDESPREAD | DIFFICULT | MODERATE | AFFECTED DATA AND FUNCTIONS | 0 | 0 |

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

Scan Summary - PCI DSS v3.2

| Category | Issues Found | Best Fix Locations |
|---|--------------|--------------------|
| PCI DSS (3.2) - 6.5.1 - Injection flaws - particularly SQL injection | 0 | 0 |
| PCI DSS (3.2) - 6.5.2 - Buffer overflows | 0 | 0 |
| 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. | 0 | 0 |
| Identification And Authentication* | Organizations must identify information system users, processes acting on behalf of users, or devices and authenticate (or verify) the identities of those users, processes, or devices, as a prerequisite to allowing access to organizational information systems. | 0 | 0 |
| 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) | 0 | 0 |
| AC-4 Information Flow Enforcement (P1) | 0 | 0 |
| AC-6 Least Privilege (P1) | 0 | 0 |
| AU-9 Protection of Audit Information (P1) | 0 | 0 |
| CM-6 Configuration Settings (P2) | 0 | 0 |
| IA-5 Authenticator Management (P1) | 0 | 0 |
| IA-6 Authenticator Feedback (P2) | 0 | 0 |
| IA-8 Identification and Authentication (Non-Organizational Users) (P1) | 0 | 0 |
| SC-12 Cryptographic Key Establishment and Management (P1) | 0 | 0 |
| SC-13 Cryptographic Protection (P1) | 0 | 0 |
| SC-17 Public Key Infrastructure Certificates (P1) | 0 | 0 |
| SC-18 Mobile Code (P2) | 0 | 0 |
| SC-23 Session Authenticity (P1)* | 0 | 0 |
| SC-28 Protection of Information at Rest (P1) | 0 | 0 |
| SC-4 Information in Shared Resources (P1) | 0 | 0 |
| SC-5 Denial of Service Protection (P1)* | 0 | 0 |
| SC-8 Transmission Confidentiality and Integrity (P1) | 0 | 0 |
| SI-10 Information Input Validation (P1)* | 0 | 0 |
| SI-11 Error Handling (P2)* | 0 | 0 |
| SI-15 Information Output Filtering (P0) | 0 | 0 |
| SI-16 Memory Protection (P1) | 0 | 0 |

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

Scan Summary - OWASP Mobile Top 10 2016

| Category | Description | Issues Found | Best Fix Locations |
|------------------------------|--|--------------|--------------------|
| M1-Improper Platform Usage | This category covers misuse of a platform feature or failure to use platform security controls. It might include Android intents, platform permissions, misuse of TouchID, the Keychain, or some other security control that is part of the mobile operating system. There are several ways that mobile apps can experience this risk. | 0 | 0 |
| M2-Insecure Data Storage | This category covers insecure data storage and unintended data leakage. | 0 | 0 |
| M3-Insecure Communication | This category covers poor handshaking, incorrect SSL versions, weak negotiation, cleartext communication of sensitive assets, etc. | 0 | 0 |
| M4-Insecure Authentication | This category captures notions of authenticating the end user or bad session management. This can include: -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

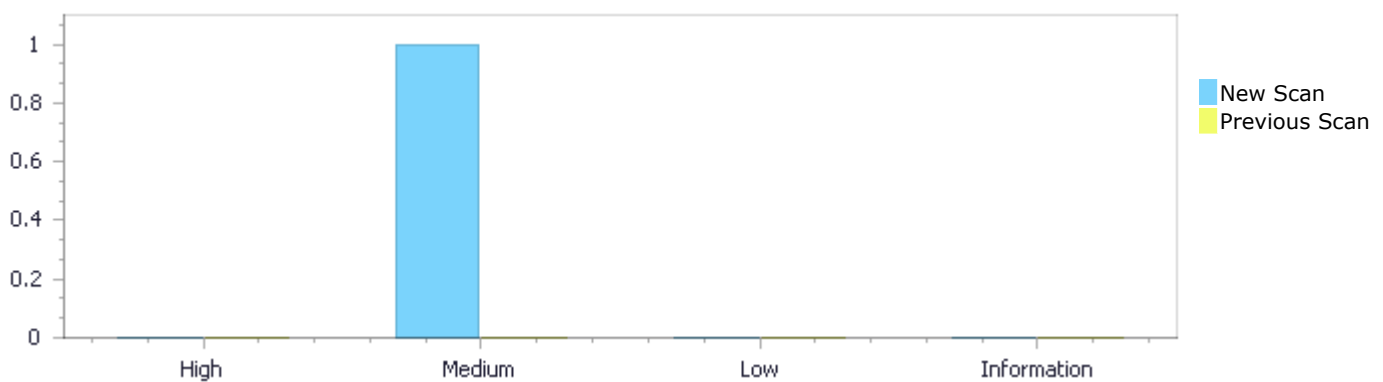
| Category | Issues Found | Best Fix Locations |
|------------|--------------|--------------------|
| Must audit | 0 | 0 |
| Check | 0 | 0 |
| Optional | 0 | 0 |

Results Distribution By Status

First scan of the project

| | High | Medium | Low | Information | Total |
|------------------|------|--------|-----|-------------|-------|
| New Issues | 0 | 1 | 0 | 0 | 1 |
| Recurrent Issues | 0 | 0 | 0 | 0 | 0 |
| Total | 0 | 1 | 0 | 0 | 1 |

| | | | | | |
|--------------|---|---|---|---|---|
| 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 | 0 | 1 | 0 | 0 | 1 |
| Urgent | 0 | 0 | 0 | 0 | 0 |
| Proposed Not Exploitable | 0 | 0 | 0 | 0 | 0 |
| Total | 0 | 1 | 0 | 0 | 1 |

Result Summary

| Vulnerability Type | Occurrences | Severity |
|---|-------------|----------|
| Wrong Size t Allocation | 1 | Medium |

10 Most Vulnerable Files

High and Medium Vulnerabilities

| File Name | Issues Found |
|--------------------------|--------------|
| proxychains-ng/hsearch.c | 1 |

Scan Results Details

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=1050057&projectid=50047&pathid=1 |
| Status | New |

The function newsize in proxychains-ng/hsearch.c at line 66 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 | proxychains-ng/hsearch.c | proxychains-ng/hsearch.c |
| Line | 77 | 77 |
| Object | newsize | newsize |

Code Snippet

File Name proxychains-ng/hsearch.c
Method static int resize(struct htab *htab, size_t nel)

```
....
77.    htab->elems = calloc(newsize, sizeof *htab->elems);
```

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

C++

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);
```


Scanned Languages

| Language | Hash Number | Change Date |
|----------|------------------|-------------|
| CPP | 4541647240435660 | 6/19/2024 |
| Common | 0105849645654507 | 6/19/2024 |