net/http/httputil: NewSingleHostReverseProxy - omit X-Forwarded-For not working #53423

Closed **firefart** opened this issue on Jun 17 · 12 comments

Labels NeedsFix release-blocker Security

Milestone Go1.19

firefart commented on Jun 17 • edited

What version of Go are you using (`go version`)?

```
$ go version
go version go1.18 linux/amd64
```

Does this issue reproduce with the latest release?

yes

What operating system and processor architecture are you using (`go env`)?

▶ `go env` Output

What did you do?

In #38079 a check was added to not leak the IP if the `X-Forwarded-For` header is nil.

Doing this does not seem to work as it always return an empty string slice

```
r.Header["X-Forwarded-For"] = nil
fmt.Printf("%T", r.Header["X-Forwarded-For"])
```

Output:

```
[]string
```

What did you expect to see?

The header is not added

What did you see instead?

The IP header is always added as it seems impossible to set the value to `nil`

go/src/net/http/httputil/reverseproxy.go
Line 301 in 9068c68

```
301     omit := ok && prior == nil // Issue 38079: nil now means don't populate the header
```

 **firefart** changed the title ~~net/http/httputil: omit X-Forwarded-For not working~~ **net/http/httputil: NewSingleHostReverseProxy - omit X-Forwarded-For not working** on Jun 17

firefart commented on Jun 17 • edited ▾ Author

The actual issue lies in the `Header.Clone` method.

`ServeHTTP` calls `req.Clone(ctx)` over here:

go/src/net/http/httputil/reverseproxy.go
Line 246 in 9068c68

```
246     outreq := req.Clone(ctx)
```

`Clone` then calls the `Clone` method on the `Headers` map:

go/src/net/http/request.go
Line 380 in 9068c68

```
380     r2.Header = r.Header.Clone()
```

I added a debug output before and after the `r.Header.Clone()` call and this is the output:

```
http.Header{"X-Forwarded-For":[]string(nil)}
-- Clone()
http.Header{"X-Forwarded-For":[]string{}}
```

So it looks like the `Header.Clone()` method does not preserve `nil` values in headers which makes the check for nil invalid and thus the client ip is always added.

This can be a privacy issue because currently golang is leaking the original client ip to the end targets even if the header is set to nil to prevent that

seankhliao commented on Jun 17 — Member

Seems to be working? https://go.dev/play/p/ME0dFW6gT1g

seankhliao added the WaitingForInfo label on Jun 17

firefart commented on Jun 17 • edited — Author

Nope. Here is a short play to demonstrate the underlying issue:
https://go.dev/play/p/c8Oc19Te76u

```
http.Header{"X-Forwarded-For":[]string(nil)}
http.Header{"X-Forwarded-For":[]string{}}
```

The `Clone` converts the `nil` value to an empty slice

Here is also some pseudocode for an http handler that forwards the request using `httputil.NewSingleHostReverseProxy` .

```
func (app *application) proxyHandler(w http.ResponseWriter, r *http.Request) {
        host, port, err := net.SplitHostPort(r.Host)
        if err != nil {
                // no port present
                host = r.Host
                port = r.URL.Port()
        }

        ctx, cancel := context.WithTimeout(r.Context(), app.timeout)
        defer cancel()
        r = r.WithContext(ctx)

        if port != "" && port != "80" && port != "443" {
                host = net.JoinHostPort(host, port)
        }
        r.URL.Host = host
        r.Host = host

        if r.URL.Scheme == "" {
                switch port {
                case "":
                        r.URL.Scheme = "http"
                case "80":
                        r.URL.Scheme = "http"
```

```
                case "443":
                        r.URL.Scheme = "https"
                default:
                        r.URL.Scheme = "http"
                }
        }

        // needed so the ip will not be leaked
        r.Header["X-Forwarded-For"] = nil

        app.logger.Debugf("port: %+v", port)
        app.logger.Debugf("r.URL: %+v", r.URL)
        app.logger.Debugf("r.RequestURI: %+v", r.RequestURI)
        app.logger.Debugf("r.Host: %+v", r.Host)
        app.logger.Debugf("r.Header: %+v", r.Header)

        proxy := httputil.NewSingleHostReverseProxy(r.URL)
        proxy.FlushInterval = -1
        proxy.ModifyResponse = app.modifyResponse
        proxy.Transport = app.httpClient.tr.Clone()

        app.logger.Debugf("sending request %+v", r)

        proxy.ServeHTTP(w, r)
}
```

If you add the debug statements on the clone method mentioned in the comment, you can see that the header value is not nil after cloning

seankhliao commented on Jun 17 — Member

I understood #38079 to mean allowing control over `X-Forwarded-For` from within the proxy, ie using your own `Director`, not from the outside.

cc **@neild**

firefart commented on Jun 17 — Author

Maybe, but the comment does not say this exactly, it only mentions the header value needs to be nil

```
// If an X-Forwarded-For header already exists, the client IP is
// appended to the existing values. As a special case, if the header
// exists in the Request.Header map but has a nil value (such as when
// set by the Director func), the X-Forwarded-For header is
// not modified.
```

I also think the `Clone()` function not handling `nil` values is a bit inconsistent as it modifies the original request.

neild commented on Jun 17 — Contributor

Given that `ReverseProxy` assigns meaning to `nil` header values, `Header.Clone` should probably preserve nil-ness.

gopherbot commented on Jun 17

Change https://go.dev/cl/412857 mentions this issue: `net/http: preserve nil values in Header.Clone`

seankhliao added NeedsFix and removed WaitingForInfo labels on Jun 18

seankhliao added this to the **Go1.20** milestone on Jun 18

gopherbot closed this as completed in `b2cc0fe` on Jun 29

---

neild commented on Jun 29 — Contributor

This has been assigned CVE-2022-32148, since it can lead to unintended leakage of private information (the client IP address).

👍 1

neild commented on Jun 29 — Contributor

@gopherbot please open backport issues.

This was referenced on Jun 29

net/http/httputil: NewSingleHostReverseProxy - omit X-Forwarded-For not working [1.17 backport] #53620

Closed

net/http/httputil: NewSingleHostReverseProxy - omit X-Forwarded-For not working [1.18 backport] #53621

Closed

gopherbot commented on Jun 29

Backport issue(s) opened: #53620 (for 1.17), #53621 (for 1.18).

Remember to create the cherry-pick CL(s) as soon as the patch is submitted to master, according to https://go.dev/wiki/MinorReleases.

gopherbot commented on Jun 29

Change https://go.dev/cl/415221 mentions this issue: `[release-branch.go1.17] net/http: preserve nil values in Header.Clone`

gopherbot commented on Jun 29

Change https://go.dev/cl/415222 mentions this issue: `[release-branch.go1.18] net/http: preserve nil values in Header.Clone`

dmitshur modified the milestones: **Go1.20**, **Go1.19** on Jul 6

neild added the Security label on Jul 6

tatianab added the release-blocker label on Jul 8

gopherbot pushed a commit that referenced this issue on Jul 12

`[release-branch.go1.17] net/http: preserve nil values in Header.Clone` ... `ed2f33e`

gopherbot pushed a commit that referenced this issue on Jul 12

`[release-branch.go1.18] net/http: preserve nil values in Header.Clone` ... `ebea1e3`

bradfitz pushed a commit to tailscale/go that referenced this issue on Jul 13

`[release-branch.go1.18] net/http: preserve nil values in Header.Clone` ... `f54a012`

tatianab mentioned this issue on Jul 14

x/vulndb: potential Go vuln in std: CVE-2022-32148 golang/vulndb#520

Closed

jproberts pushed a commit to jproberts/go that referenced this issue on Aug 9

`net/http: preserve nil values in Header.Clone` ··· `4103516`

danbudris pushed a commit to danbudris/go that referenced this issue on Sep 9

`[release-branch.go1.17] net/http: preserve nil values in Header.Clone` ··· `0cf92aa`

danbudris pushed a commit to danbudris/go that referenced this issue on Sep 9

`[release-branch.go1.17] net/http: preserve nil values in Header.Clone` ··· `503ce82`

danbudris pushed a commit to danbudris/go that referenced this issue on Sep 12

`[release-branch.go1.17] net/http: preserve nil values in Header.Clone` ··· `e70280c`

danbudris pushed a commit to danbudris/go that referenced this issue on Sep 14

`[release-branch.go1.17] net/http: preserve nil values in Header.Clone` ··· `a966897`

rcrozean pushed a commit to rcrozean/go that referenced this issue on Oct 5

`net/http: preserve nil values in Header.Clone` ··· `b9c55bb`

rcrozean pushed a commit to rcrozean/go that referenced this issue on Oct 12

`net/http: preserve nil values in Header.Clone` ··· `45e00eb`

rcrozean pushed a commit to rcrozean/go that referenced this issue on Oct 12

`net/http: preserve nil values in Header.Clone` ··· `67da253`

Assignees

No one assigned

Labels

NeedsFix release-blocker Security

Projects

None yet

Milestone

Go1.19

Development

No branches or pull requests

6 participants