The pattern '/\domain.com' is not disallowed when redirecting, allowing for open redirect

High **starkers** published **GHSA-qqxw-m5fj-f7gv** on Jan 29, 2020

Package

No package listed

Affected versions

All

Patched versions

None

Description

Impact

An open redirect vulnerability has been found in `oauth2_proxy`. Anyone who uses `oauth2_proxy` may potentially be impacted.

For a context detectify have an in depth blog post about the potential impact of an open redirect. Particularly see the OAuth section.

tl;dr: People's authentication tokens could be silently harvested by an attacker. e.g:

```
facebook.com/oauth.php?clientid=123&state=abc&redirect_url=https://yourdomain.com/red.php?url%3dhttps://attacker.com/
```

Patches

@sauyon found the issue, and has submitted a patch.

```
diff --git a/oauthproxy.go b/oauthproxy.go
index 72ab580..f420df6 100644
--- a/oauthproxy.go
+++ b/oauthproxy.go
@@ -517,7 +517,7 @@ func (p *OAuthProxy) GetRedirect(req *http.Request) (redirect string, err error)
 // IsValidRedirect checks whether the redirect URL is whitelisted
 func (p *OAuthProxy) IsValidRedirect(redirect string) bool {
        switch {
-       case strings.HasPrefix(redirect, "/") && !strings.HasPrefix(redirect, "//"):
+       case strings.HasPrefix(redirect, "/") && !strings.HasPrefix(redirect, "//") && !strings.HasPrefix(redirect, "/\\"):
                return true
        case strings.HasPrefix(redirect, "http://") || strings.HasPrefix(redirect, "https://"):
                redirectURL, err := url.Parse(redirect)
```

This patch will be applied to the next release, which is scheduled for when this is publicly disclosed.

Workarounds

At this stage there is no work around.

References

- detectify's blog post

Severity

High

CVE ID

CVE-2020-5233

Weaknesses

No CWEs