COMMUNITY CONTENT

Home › Submit › 781769

Submit #781769: Casdoor v2.356.0 Open Redirect

Title Casdoor v2.356.0 Open Redirect

Description **Evidence:**

```go
for _, targetUri := range application.RedirectUris {
    targetUriRegex := regexp.MustCompile(targetUri)
    if targetUriRegex.MatchString(redirectUri) || strings.Contains(redirectUri, targetUri) {
        return true
    }
}
```

Two critical flaws:

1. `strings.Contains(redirectUri, targetUri)` is a substring match. If an application registers `https://app.example.com/callback`, then `https://evil.com/https://app.example.com/callback` passes validation.
2. `regexp.MustCompile(targetUri)` treats stored redirect URIs as regex patterns. A URI containing `.` matches any character. An entry like `https://example.com` matches `https://exampleXcom.evil.com`.

Attack scenario: An attacker constructs an OAuth authorization request with `redirect_uri=https://evil.com/?x=https://app.example.com/callback`. This passes the `strings.Contains` check. The OAuth code/token is then sent to `evil.com`.

Fix: Use strict equality or validated prefix matching. Never treat user-configured values as regex patterns. Compare parsed URL components (scheme, host, path) individually:

```go
func (application *Application) IsRedirectUriValid(redirectUri string) bool {
    parsedRedirect, err := url.Parse(redirectUri)
    if err != nil { return false }
    for _, targetUri := range application.RedirectUris {
        parsedTarget, err := url.Parse(targetUri)
        if err != nil { continue }
        if parsedRedirect.Scheme == parsedTarget.Scheme &&
            parsedRedirect.Host == parsedTarget.Host &&
            strings.HasPrefix(parsedRedirect.Path, parsedTarget.Path) {
            return true
        }
    }
    return false
}
```

Community Content

Submissions are made by VulDB community users. VulDB is *not responsible* for their content nor the links to external sources.

Please use the raw information shown and the links listed *with caution*. They might contain malicious and harmful actions, code or data.

The corresponding VulDB entries contain the moderated, verified, and normalized information provided within the raw submission.

Documentation

- Submission Policy
- Data Processing
- CVE Handling

User	Ghulfran Khan (UID 95493)
Submission	03/17/2026 02:23 PM (17 days ago)
Moderation	04/03/2026 09:26 AM (17 days later)
Status	Accepted
VulDB entry	[Casdoor 2.356.0 OAuth Authorization Request redirect_uri]
Points	17