acunetix

Developer Report

Acunetix Security Audit

19 June 2018

Generated by Acunetix

Scan of https://nxtstep.eastus.cloudapp.azure.com/stp-cogtwin/

Scan details

Scan information	
Start time	19/06/2018, 11:37:15
Start url	https://nxtstep.eastus.cloudapp.azure.com/stp-cogtwin/
Host	https://nxtstep.eastus.cloudapp.azure.com/stp-cogtwin/
Scan time	4 minutes, 33 seconds
Profile	Full Scan

Threat level

Acunetix Threat Level 2

One or more medium-severity type vulnerabilities have been discovered by the scanner. You should investigate each of these vulnerabilities to ensure they will not escalate to more severe problems.

Alerts distribution

Total alerts found	2
High	0
Medium	1
Low	1
Informational	0

Alerts summary

HTML form without CSRF protection

Classification	
CVSS2	Base Score: 2.6 Access Vector: Network_accessible Access Complexity: High Authentication: None Confidentiality Impact: None Integrity Impact: Partial Availability Impact: None Exploitability: Not_defined Remediation Level: Not_defined Report Confidence: Not_defined Availability Requirement: Not_defined Collateral Damage Potential: Not_defined Confidentiality Requirement: Not_defined Integrity Requirement: Not_defined Target Distribution: Not_defined
CVSS3	Base Score: 4.3 Attack Vector: Network Attack Complexity: Low Privileges Required: None User Interaction: Required Scope: Unchanged Confidentiality Impact: None Integrity Impact: Low Availability Impact: None
CWE	CWE-352

Affected items	Variation
Web Server	1

Clickjacking: X-Frame-Options header missing

Classification	
CVSS2	Base Score: 6.8 Access Vector: Network_accessible Access Complexity: Medium Authentication: None Confidentiality Impact: Partial Integrity Impact: Partial Availability Impact: Partial Exploitability: Not_defined Remediation Level: Not_defined Report Confidence: Not_defined Availability Requirement: Not_defined Collateral Damage Potential: Not_defined Confidentiality Requirement: Not_defined Integrity Requirement: Not_defined Target Distribution: Not_defined
CWE	CWE-693

Affected items	Variation
Web Server	1

Alerts details

HTML form without CSRF protection

Severity	**Medium**
Reported by module	/Crawler/12-Crawler_Form_NO_CSRF.js

Description

This alert requires manual confirmation

Cross-Site Request Forgery (CSRF, or XSRF) is a vulnerability wherein an attacker tricks a victim into making a request the victim did not intend to make. Therefore, with CSRF, an attacker abuses the trust a web application has with a victim's browser.

Acunetix found an HTML form with no apparent anti-CSRF protection implemented. Consult the 'Attack details' section for more information about the affected HTML form.

Impact

An attacker could use CSRF to trick a victim into accessing a website hosted by the attacker, or clicking a URL containing malicious or unauthorized requests.

CSRF is a type of 'confused deputy' attack which leverages the authentication and authorization of the victim when the forged request is being sent to the web server. Therefore, if a CSRF vulnerability could affect highly privileged users such as administrators full application compromise may be possible.

Recommendation

Verify if this form requires anti-CSRF protection and implement CSRF countermeasures if necessary.

The recommended and the most widely used technique for preventing CSRF attacks is know as an anti-CSRF token, also sometimes referred to as a synchronizer token. The characteristics of a well designed anti-CSRF system involve the following attributes.

- The anti-CSRF token should be unique for each user session
- The session should automatically expire after a suitable amount of time
- The anti-CSRF token should be a cryptographically random value of significant length
- The anti-CSRF token should be cryptographically secure, that is, generated by a strong Pseudo-Random Number Generator (PRNG) algorithm
- The anti-CSRF token is added as a hidden field for forms, or within URLs (only necessary if GET requests cause state changes, that is, GET requests are not idempotent)
- The server should reject the requested action if the anti-CSRF token fails validation

When a user submits a form or makes some other authenticated request that requires a Cookie, the anti-CSRF token should be included in the request. Then, the web application will then verify the existence and correctness of this token before processing the request. If the token is missing or incorrect, the request can be rejected.

References

What is Cross Site Reference Forgery (CSRF)? (https://www.acunetix.com/websitesecurity/csrf-attacks/)
Cross-Site Request Forgery (CSRF) Prevention Cheatsheet (https://www.owasp.org/index.php/Cross-Site_Request_Forgery_(CSRF)_Prevention_Cheat_Sheet)
The Cross-Site Request Forgery (CSRF/XSRF) FAQ (http://www.cgisecurity.com/csrf-faq.html)
Cross-site Request Forgery (https://en.wikipedia.org/wiki/Cross-site_request_forgery)

Affected items

Web Server
Details
Request headers

```
GET /stp-cogtwin/ HTTP/1.1
```

```
Accept: */*
Accept-Encoding: gzip,deflate
Host: nxtstep.eastus.cloudapp.azure.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.21 (KHTML, like Gecko)
Chrome/41.0.2228.0 Safari/537.21
Proxy-Connection: Keep-alive
```

Clickjacking: X-Frame-Options header missing

Severity	**Low**
Reported by module	/Scripts/PerServer/Clickjacking_X_Frame_Options.script

Description

Clickjacking (User Interface redress attack, UI redress attack, UI redressing) is a malicious technique of tricking a Web user into clicking on something different from what the user perceives they are clicking on, thus potentially revealing confidential information or taking control of their computer while clicking on seemingly innocuous web pages.

The server didn't return an **X-Frame-Options** header which means that this website could be at risk of a clickjacking attack. The X-Frame-Options HTTP response header can be used to indicate whether or not a browser should be allowed to render a page inside a frame or iframe. Sites can use this to avoid clickjacking attacks, by ensuring that their content is not embedded into other sites.

Impact

The impact depends on the affected web application.

Recommendation

Configure your web server to include an X-Frame-Options header. Consult Web references for more information about the possible values for this header.

References

The X-Frame-Options response header (https://developer.mozilla.org/en-US/docs/HTTP/X-Frame-Options)
Clickjacking (http://en.wikipedia.org/wiki/Clickjacking)
OWASP Clickjacking (https://www.owasp.org/index.php/Clickjacking)
Defending with Content Security Policy frame-ancestors directive (https://www.owasp.org/index.php/Clickjacking_Defense_Cheat_Sheet#Defending_with_Content_Security_Policy_frame-ancestors_directive)
Frame Buster Buster (http://stackoverflow.com/questions/958997/frame-buster-buster-buster-code-needed)

Affected items

Web Server
Details
Request headers

```
GET / HTTP/1.1
Accept: */*
Accept-Encoding: gzip,deflate
Host: nxtstep.eastus.cloudapp.azure.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.21 (KHTML, like Gecko)
Chrome/41.0.2228.0 Safari/537.21
Proxy-Connection: keep-alive
```

Scanned items (coverage report)

https://nxtstep.eastus.cloudapp.azure.com/
https://nxtstep.eastus.cloudapp.azure.com/stp-cogtwin
https://nxtstep.eastus.cloudapp.azure.com/stp-cogtwin/img
https://nxtstep.eastus.cloudapp.azure.com/stp-cogtwin/myform