IOTHREAT

Security Report
iothreat.com

Executive Summary

IOTHREAT conducted an automated penetration testing and vulnerability assessment of your web application hosted at the following web address: https://www.iothreat.com on January 17, 2022.

The security review covered the following topics:

1. Web Application Vulnerabilities
2. Compromised Accounts
3. Open Ports and Services
4. Malware Detections
5. Email Security
6. SSL Certificate
7. Subdomain Discovery

During the security review, several security issues were detected, the complete details of which, including supporting evidence, and mitigation recommendations, are detailed below.

Web Application Vulnerabilities

We have detected 8 vulnerabilities. Review the list of detected vulnerabilities, and apply the suggested mitigation recommendations.

As a security best practice, we recommend configuring a Web Application Firewall (WAF) to protect your web application from multiple attacks (e.g., SQL Injection, Cross-Site Scripting, and many more).

1. CSP: Wildcard Directive

Severity:

Medium

Description:

The following directives either allow wildcard sources (or ancestors), are not defined, or are overly broadly defined: script-src, style-src, img-src, connects-src, frame-src, font-src, media-src, object-src, manifest-src, worker-src, prefetch-src, form-actionThe directive(s): form-action are among the directives that do not fallback to default-src, missing/excluding them is the same as allowing anything.

Mitigation:
Ensure that your web server, application server, load balancer, etc. is properly configured to set the Content-Security-Policy header.

Affected URLs:
1. https://www.iothreat.com/
HTTP Method: GET
Parameter:
Evidence: frame-ancestors 'self'

2. https://www.iothreat.com/?Email-2=foo-bar%40example.com
HTTP Method: GET
Parameter:
Evidence: frame-ancestors 'self'

3. https://www.iothreat.com/about
HTTP Method: GET
Parameter:
Evidence: frame-ancestors 'self'

4. https://www.iothreat.com/blog
HTTP Method: GET
Parameter:
Evidence: frame-ancestors 'self'

5. https://www.iothreat.com/careers
HTTP Method: GET

Parameter:
Evidence: frame-ancestors 'self'

6. https://www.iothreat.com/contact
HTTP Method: GET
Parameter:
Evidence: frame-ancestors 'self'

7. https://www.iothreat.com/pricing
HTTP Method: GET
Parameter:
Evidence: frame-ancestors 'self'

8. https://www.iothreat.com/utility-pages/privacy-policy
HTTP Method: GET
Parameter:
Evidence: frame-ancestors 'self'

9. https://www.iothreat.com/utility-pages/sign-in
HTTP Method: GET
Parameter:
Evidence: frame-ancestors 'self'

10. https://www.iothreat.com/utility-pages/terms-conditions
HTTP Method: GET
Parameter:
Evidence: frame-ancestors 'self'

2. Absence of Anti-CSRF Tokens

Severity:
Low

Description:

No Anti-CSRF tokens were found in a HTML submission form.A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf.CSRF attacks are effective in a number of situations, including: * The victim has an active session on the target site. * The victim is authenticated via HTTP auth on the target site. * The victim is on the same local network as the target site.CSRF has primarily been used to perform an action against a target site using the victim's privileges, but recent techniques have been discovered to disclose information by gaining access to the response. The risk of information disclosure is dramatically increased when the target site is vulnerable to XSS, because XSS can be used as a platform for CSRF, allowing the attack to operate within the bounds of the same-origin policy.

Mitigation:

Phase: Architecture and DesignUse a vetted library or framework that does not allow this weakness to occur or provides constructs that make this weakness easier to avoid.For example, use anti-CSRF packages such as the OWASP CSRFGuard.Phase: ImplementationEnsure that your application is free of cross-site scripting issues, because most CSRF defenses can be bypassed using attacker-controlled script.Phase: Architecture and DesignGenerate a unique nonce for each form, place the nonce into the form, and verify the nonce upon receipt of the form. Be sure that the nonce is not predictable (CWE-330).Note that this can be bypassed using XSS.Identify especially dangerous operations. When the user performs a dangerous operation, send a separate confirmation request to ensure that the user intended to perform that operation.Note that this can be bypassed using XSS.Use the ESAPI Session Management control.This control includes a component for CSRF.Do not use the GET method for any request that triggers a state change.Phase: ImplementationCheck the HTTP Referer header to see if the request originated from an expected page. This could break legitimate functionality, because users or proxies may have disabled sending the Referer for privacy reasons.

Affected URLs:

1. https://www.iothreat.com/
HTTP Method: GET
Parameter:
Evidence: <form id="email-form" name="email-form" data-name="Email Form" redirect="https://checkup.iothreat.com/" data-redirect="https://checkup.iothreat.com/" method="get" class="home-v3-newsletter-form-wrapper">

2. https://www.iothreat.com/
HTTP Method: GET
Parameter:
Evidence: <form id="email-form" name="email-form" data-name="Email Form" redirect="https://checkup.iothreat.com/" data-redirect="https://checkup.iothreat.com/" method="get" class="cta-v5-newsletter-form-wrapper">

3. https://www.iothreat.com/
HTTP Method: GET
Parameter:
Evidence: <form id="email-form" name="email-form" data-name="Email Form" redirect="https://checkup.iothreat.com/" data-redirect="https://checkup.iothreat.com/" method="get" class="cta-v5-newsletter-form-wrapper">

4. https://www.iothreat.com/
HTTP Method: GET
Parameter:
Evidence: <form id="email-form" name="email-form" data-name="Email Form" redirect="https://checkup.iothreat.com/" data-redirect="https://checkup.iothreat.com/" method="get" class="cta-v5-newsletter-form-wrapper">

5. https://www.iothreat.com/
HTTP Method: GET
Parameter:

Evidence: <form id="email-form" name="email-form" data-name="Email Form" redirect="https://checkup.iothreat.com/" data-redirect="https://checkup.iothreat.com/" method="get" class="cta-v5-newsletter-form-wrapper">

6. https://www.iothreat.com/
HTTP Method: GET
Parameter:
Evidence: <form id="email-form" name="email-form" data-name="Email Form" redirect="https://checkup.iothreat.com/" data-redirect="https://checkup.iothreat.com/" method="get" class="fotter-newsletter-form-wrapper">

3. Cross-Domain JavaScript Source File Inclusion

Severity:
Low

Description:
The page includes one or more script files from a third-party domain.

Mitigation:
Ensure JavaScript source files are loaded from only trusted sources, and the sources can't be controlled by end users of the application.

Affected URLs:
1. https://www.iothreat.com/
HTTP Method: GET
Parameter: //cdnjs.cloudflare.com/ajax/libs/placeholders/3.0.2/placeholders.min.js
Evidence: <script src="//cdnjs.cloudflare.com/ajax/libs/placeholders/3.0.2/placeholders.min.js"></script>

2. https://www.iothreat.com/
HTTP Method: GET
Parameter: https://cdnjs.cloudflare.com/ajax/libs/html5shiv/3.7.3/html5shiv.min.js
Evidence: <script src="https://cdnjs.cloudflare.com/ajax/libs/html5shiv/3.7.3/html5shiv.min.js"

type="text/javascript"></script>

3. https://www.iothreat.com/
HTTP Method: GET
Parameter: https://uploads-ssl.webflow.com/60f4ba9be544e212f137162f/js/webflow.bf12808f6.js
Evidence: <script src="https://uploads-ssl.webflow.com/60f4ba9be544e212f137162f/js/webflow.bf12808f6.js" type="text/javascript"></script>

4. https://www.iothreat.com/
HTTP Method: GET
Parameter: https://www.googletagmanager.com/gtag/js?id=G-WF2QDYMLZ4
Evidence: <script async src="https://www.googletagmanager.com/gtag/js?id=G-WF2QDYMLZ4"></script>

5. https://www.iothreat.com/about
HTTP Method: GET
Parameter: //cdnjs.cloudflare.com/ajax/libs/placeholders/3.0.2/placeholders.min.js
Evidence: <script src="//cdnjs.cloudflare.com/ajax/libs/placeholders/3.0.2/placeholders.min.js"></script>

6. https://www.iothreat.com/about
HTTP Method: GET
Parameter: https://cdnjs.cloudflare.com/ajax/libs/html5shiv/3.7.3/html5shiv.min.js
Evidence: <script src="https://cdnjs.cloudflare.com/ajax/libs/html5shiv/3.7.3/html5shiv.min.js" type="text/javascript"></script>

7. https://www.iothreat.com/about
HTTP Method: GET
Parameter: https://uploads-ssl.webflow.com/60f4ba9be544e212f137162f/js/webflow.bf12808f6.js
Evidence: <script src="https://uploads-ssl.webflow.com/60f4ba9be544e212f137162f/js/webflow.bf12808f6.js"

type="text/javascript"></script>

8. https://www.iothreat.com/about
HTTP Method: GET
Parameter: https://www.googletagmanager.com/gtag/js?id=G-WF2QDYMLZ4
Evidence: <script async src="https://www.googletagmanager.com/gtag/js?id=G-WF2QDYMLZ4"></script>

4. Incomplete or No Cache-control Header Set

Severity:
Low

Description:
The cache-control header has not been set properly or is missing, allowing the browser and proxies to cache content.

Mitigation:
Whenever possible ensure the cache-control HTTP header is set with no-cache, no-store, must-revalidate.

Affected URLs:
1. https://www.iothreat.com/
HTTP Method: GET
Parameter: Cache-Control
Evidence:

2. https://www.iothreat.com/?Email-2=foo-bar%40example.com
HTTP Method: GET
Parameter: Cache-Control
Evidence:

3. https://www.iothreat.com/about
HTTP Method: GET

Parameter: Cache-Control
Evidence:

4. https://www.iothreat.com/blog
HTTP Method: GET
Parameter: Cache-Control
Evidence:

5. https://www.iothreat.com/contact
HTTP Method: GET
Parameter: Cache-Control
Evidence:

6. https://www.iothreat.com/pricing
HTTP Method: GET
Parameter: Cache-Control
Evidence:

7. https://www.iothreat.com/robots.txt
HTTP Method: GET
Parameter: Cache-Control
Evidence:

8. https://www.iothreat.com/sitemap.xml
HTTP Method: GET
Parameter: Cache-Control
Evidence:

9. https://www.iothreat.com/utility-pages/privacy-policy
HTTP Method: GET
Parameter: Cache-Control
Evidence:

10. https://www.iothreat.com/utility-pages/sign-in
HTTP Method: GET
Parameter: Cache-Control
Evidence:

11. https://www.iothreat.com/utility-pages/terms-conditions
HTTP Method: GET
Parameter: Cache-Control
Evidence:

5. Private IP Disclosure

Severity:
Low

Description:
A private IP (such as 10.x.x.x, 172.x.x.x, 192.168.x.x) or an Amazon EC2 private hostname (for example, ip-10-0-56-78) has been found in the HTTP response body. This information might be helpful for further attacks targeting internal systems.

Mitigation:
Remove the private IP address from the HTTP response body. For comments, use JSP/ASP/PHP comment instead of HTML/JavaScript comment which can be seen by client browsers.

Affected URLs:
1. https://www.iothreat.com/blog-category/vulnerabilities
HTTP Method: GET
Parameter:
Evidence: ip-10-0-56-78

2. https://www.iothreat.com/blog-category/vulnerabilities?email-2=foo-bar%40example.com
HTTP Method: GET
Parameter:

Evidence: ip-10-0-56-78

3. https://www.iothreat.com/blog-category/vulnerabilities?email=foo-bar%40example.com
HTTP Method: GET
Parameter:
Evidence: ip-10-0-56-78

4. https://www.iothreat.com/blog-category/vulnerabilities?email=foo-bar%40example.com&email-2=foo-bar%40example.com
HTTP Method: GET
Parameter:
Evidence: ip-10-0-56-78

5. https://www.iothreat.com/blog/private-ip-disclosure
HTTP Method: GET
Parameter:
Evidence: ip-10-0-56-78

6. https://www.iothreat.com/blog/private-ip-disclosure?email-2=foo-bar%40example.com
HTTP Method: GET
Parameter:
Evidence: ip-10-0-56-78

7. https://www.iothreat.com/blog/private-ip-disclosure?email=foo-bar%40example.com
HTTP Method: GET
Parameter:
Evidence: ip-10-0-56-78

8. https://www.iothreat.com/blog/private-ip-disclosure?email=foo-bar%40example.com&email-2=foo-bar%40example.com

HTTP Method: GET
Parameter:
Evidence: ip-10-0-56-78

6. Strict-Transport-Security Header Not Set

Severity:
Low

Description:
HTTP Strict Transport Security (HSTS) is a web security policy mechanism whereby a web server declares that complying user agents (such as a web browser) are to interact with it using only secure HTTPS connections (i.e. HTTP layered over TLS/SSL). HSTS is an IETF standards track protocol and is specified in RFC 6797.

Mitigation:
Ensure that your web server, application server, load balancer, etc. is configured to enforce Strict-Transport-Security.

Affected URLs:
1. https://www.iothreat.com/
HTTP Method: GET
Parameter:
Evidence:

2. https://www.iothreat.com/?Email-2=foo-bar%40example.com
HTTP Method: GET
Parameter:
Evidence:

3. https://www.iothreat.com/about
HTTP Method: GET
Parameter:
Evidence:

4. https://www.iothreat.com/blog
HTTP Method: GET
Parameter:
Evidence:

5. https://www.iothreat.com/contact
HTTP Method: GET
Parameter:
Evidence:

6. https://www.iothreat.com/pricing
HTTP Method: GET
Parameter:
Evidence:

7. https://www.iothreat.com/robots.txt
HTTP Method: GET
Parameter:
Evidence:

8. https://www.iothreat.com/sitemap.xml
HTTP Method: GET
Parameter:
Evidence:

9. https://www.iothreat.com/utility-pages/privacy-policy
HTTP Method: GET
Parameter:
Evidence:

10. https://www.iothreat.com/utility-pages/sign-in

HTTP Method: GET
Parameter:
Evidence:

11. https://www.iothreat.com/utility-pages/terms-conditions
HTTP Method: GET
Parameter:
Evidence:

7. Timestamp Disclosure - Unix

Severity:
Low

Description:
A timestamp was disclosed by the application/web server - Unix

Mitigation:
Manually confirm that the timestamp data is not sensitive, and that the data cannot be aggregated to disclose exploitable patterns.

Affected URLs:
1. https://www.iothreat.com/blog/content-type-header-missing?email-2=foo-bar%40example.com
HTTP Method: GET
Parameter:
Evidence: 25293479

2. https://www.iothreat.com/blog/cookie-poisoning?email=foo-bar%40example.com&email-2=foo-bar%40example.com
HTTP Method: GET
Parameter:
Evidence: 46928719

3.
https://www.iothreat.com/blog/information-disclosure-debug-error-messages?email-2=foo-bar%40example.com
HTTP Method: GET
Parameter:
Evidence: 67030049

4.
https://www.iothreat.com/blog/remote-file-inclusion?email=foo-bar%40example.com&email-2=foo-bar%40example.com
HTTP Method: GET
Parameter:
Evidence: 70486563

5. https://www.iothreat.com/blog/source-code-disclosure-git?email=foo-bar%40example.com
HTTP Method: GET
Parameter:
Evidence: 90422217

6. https://www.iothreat.com/blog/user-controllable-charset?email-2=foo-bar%40example.com
HTTP Method: GET
Parameter:
Evidence: 74667063

7. https://www.iothreat.com/blog/viewstate-without-mac-signature-sure
HTTP Method: GET
Parameter:
Evidence: 50486154

8. https://www.iothreat.com/blog?8421f9e3_page=3&email=foo-bar%40example.com
HTTP Method: GET
Parameter:

Evidence: 03292826

9. https://www.iothreat.com/careers?email-2=foo-bar%40example.com
HTTP Method: GET
Parameter:
Evidence: 20619514

10. https://www.iothreat.com/category/monthly?email=foo-bar%40example.com&email-2=foo-bar%40example.com
HTTP Method: GET
Parameter:
Evidence: 98152223

11. https://www.iothreat.com/pricing?email=foo-bar%40example.com&email-2=foo-bar%40example.com
HTTP Method: GET
Parameter:
Evidence: 72839294

8. X-Content-Type-Options Header Missing

Severity:
Low

Description:
The Anti-MIME-Sniffing header X-Content-Type-Options was not set to 'nosniff'. This allows older versions of Internet Explorer and Chrome to perform MIME-sniffing on the response body, potentially causing the response body to be interpreted and displayed as a content type other than the declared content type. Current (early 2014) and legacy versions of Firefox will use the declared content type (if one is set), rather than performing MIME-sniffing.

Mitigation:

Ensure that the application/web server sets the Content-Type header appropriately, and that it sets the X-Content-Type-Options header to 'nosniff' for all web pages.If possible, ensure that the end user uses a standards-compliant and modern web browser that does not perform MIME-sniffing at all, or that can be directed by the web application/web server to not perform MIME-sniffing.

Affected URLs:

1. https://www.iothreat.com/
HTTP Method: GET
Parameter: X-Content-Type-Options
Evidence:

2. https://www.iothreat.com/?Email-2=foo-bar%40example.com
HTTP Method: GET
Parameter: X-Content-Type-Options
Evidence:

3. https://www.iothreat.com/about
HTTP Method: GET
Parameter: X-Content-Type-Options
Evidence:

4. https://www.iothreat.com/blog
HTTP Method: GET
Parameter: X-Content-Type-Options
Evidence:

5. https://www.iothreat.com/contact
HTTP Method: GET
Parameter: X-Content-Type-Options
Evidence:

6. https://www.iothreat.com/pricing
HTTP Method: GET

Parameter: X-Content-Type-Options
Evidence:

7. https://www.iothreat.com/robots.txt
HTTP Method: GET
Parameter: X-Content-Type-Options
Evidence:

8. https://www.iothreat.com/sitemap.xml
HTTP Method: GET
Parameter: X-Content-Type-Options
Evidence:

9. https://www.iothreat.com/utility-pages/privacy-policy
HTTP Method: GET
Parameter: X-Content-Type-Options
Evidence:

10. https://www.iothreat.com/utility-pages/sign-in
HTTP Method: GET
Parameter: X-Content-Type-Options
Evidence:

11. https://www.iothreat.com/utility-pages/terms-conditions
HTTP Method: GET
Parameter: X-Content-Type-Options
Evidence:

Compromised Accounts

We have not detected any compromised accounts at this time.

As a security best practice, we recommend implementing a Password Manager to generate and maintain all login credentials, and also enabling Two-Step Verification across all online services to

protect your accounts from Cybercriminals.

Open Ports and Services

We have detected 2 open ports. Review the list of detected open ports, and consider closing any ports other than 80 (HTTP) and 443 (SSL).

IP Address:
75.2.70.75

Hostname:
aacb0a264e514dd48.awsglobalaccelerator.com

This is the list of discovered open ports and services:

1. Port: 80
Service: HTTP
State: open
Protocol: tcp
Product: OpenResty web app server

2. Port: 443
Service: SSL
State: open
Protocol: tcp
Product: OpenResty web app server

Malware Detections

Website

We have not detected any malicious activity at this time.
Use an up-to-date malware scanner to check iothreat.com weekly.

Registrar
GoDaddy.com, LLC

Registration Date

December 16, 2014

Expiration Date

December 16, 2026

Name Servers

ns-1377.awsdns-44.org

ns-1902.awsdns-45.co.uk

ns-221.awsdns-27.com

ns-857.awsdns-43.net

Hosting Server

We have not detected any malicious activity at this time.

Use an up-to-date malware scanner to check 75.2.70.75 weekly.

IP Address

75.2.70.75

ISP

AMAZON-02

Country

US

Email Security

Monitor your email security configurations monthly to protect your customers from Phishing scams.

MX

No issues - well done!

Description:

A mail exchanger record (MX record) specifies the mail servers responsible for accepting email messages on behalf of a domain name.

Details:

alt1.aspmx.l.google.com

alt2.aspmx.l.google.com

alt3.aspmx.l.google.com

alt4.aspmx.l.google.com

aspmx.l.google.com

SPF

No issues - well done!

Description:

Sender Policy Framework (SPF) lets the domain owner authorize IP addresses that are allowed to send email for the domain. Receiving servers can verify that messages appearing to come from a specific domain are sent from servers allowed by the domain owner.

Details:

v=spf1 include:_spf.google.com -all

DMARC

No issues - well done!

Description:

Domain-based Message Authentication, Reporting, and Conformance (DMARC) tells receiving mail servers what to do when they get a message that appears to be from your organization, but does not pass authentication checks, or does not meet the authentication requirements in your DMARC policy record. Messages that are not authenticated might be impersonating your organization, or might be sent from unauthorized servers.

Details:

v=DMARC1; p=reject

SSL Certificate

We have detected one issue with your SSL configuration.

Fix the discovered SSL issue to improve the security of your data.

SSL Issue:
HSTS is not offered

Grade:
A

Issuer:
R3 (Let's Encrypt from US)

Expiration Date:
February 3, 2022

Subdomain Discovery

We have detected 5 subdomains.
Scan the 5 discovered subdomains for malicious activity, SSL issues, open ports, and vulnerabilities.

Subdomains:
1. api.iothreat.com
2. checkup.iothreat.com
3. scan.iothreat.com
4. support.iothreat.com
5. www.iothreat.com

Methodology

The DAST methodology was used to test your web application. Dynamic Application Security Testing (DAST) is a black-box security testing methodology in which an application is tested from the outside. A tester using DAST examines an application when it is running and tries to hack it just like an attacker would.

Contact Email:
security@iothreat.com