Web Application Audit Report

Demo Account

http://testhtml5.vulnweb.com/#/popular

Confidential

Scan Date: 2017-01-16

Scope

Application Audit for URL http://testhtml5.vulnweb.com/#/popular

Limitations

1. The entire test was carried out with no prior knowledge of the systems and applications.
2. All test were carried out without any known credentials to systems and applications.
3. IndusGuard does not allowed to carry out any DoS attacks or to run any exploits which can affect systems availability.

Confidentiality

This document contains sensitive and/or confidential information, do not distribute, email, fax or transfer via any electronic mechanism without proper authorization. Information contained with in this document should be handled with appropriate caution. While reasonable attempts have been made to confirm the accuracy of the data contained herein, IndusGuard, assumes no liability for the completeness, use of, or conclusions drawn from such data.

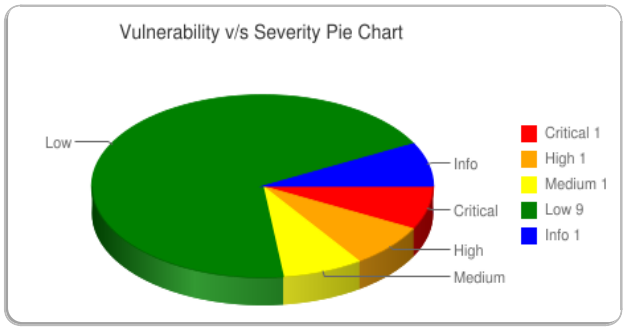
Disclaimer

This, or any other, Security Audit cannot and does not guarantee security. IndusGuard makes no warranty or claim of any kind, whatsoever, about the accuracy or usefulness of any information provided herein. By using this information you agree that IndusGuard shall be held harmless in any event. IndusGuard makes this information available solely under its Terms of Service Agreement published at soc.indusguard.com.

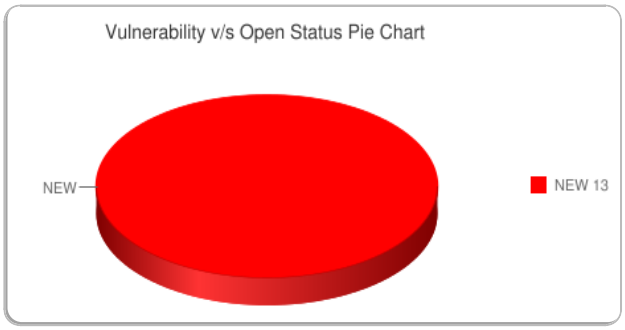
Executive Summary

Total number of vulnerability(s) identified are 13

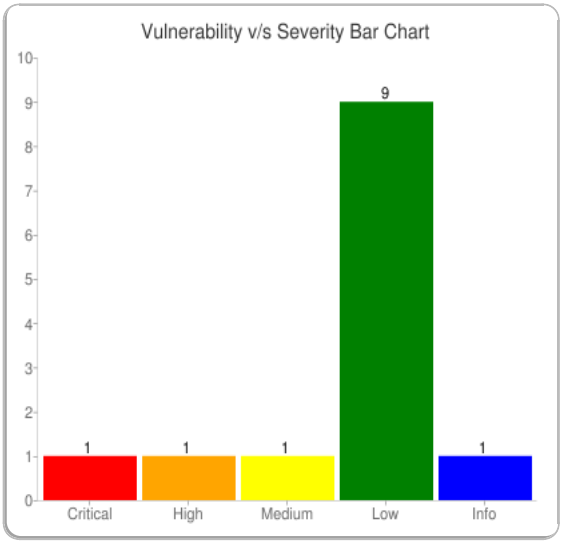
Vulnerability v/s Severity Pie Chart



Vulnerability v/s Open Status Pie Chart



Vulnerability v/s Severity Bar Chart



Vulnerability Summary

Severity	Total
Critical	1
High	1
Medium	1
Low	9
Info	1

Scan Date: 2017-01-16

Vulnerability Details

Title	Total
HTTP Basic Authentication Enabled	1
Cross Site Scripting Vulnerability	1
Form action submits sensitive data in the clear	1
Autocomplete enabled for sensitive HTML form fields	1
Predictable resource location	8
Web server version disclosure	1

Vulnerabilities

Alert ID: 481472 Found on: 2017-01-16 Severity: Critical

HTTP Basic Authentication Enabled

Open Status: NEW **First Found:** 2017-01-16

URL: http://testhtml5.vulnweb.com/admin
Method: get
Cvss Score: 6.5
Cvss Vector: (AV:N/AC:L/Au:S/C:P/I:P/A:P)
PCI Compliance: Fail

Description:

The HTTP Basic Authentication scheme is not considered to be a secure method of user authentication (unless used in conjunction with some external secure system such as TLS/SSL), as the user name and password are passed over the network as cleartext.

Solution:

- Use Basic Authentication over TLS/SSL (HTTPS)

  Enable HTTPS on the Web server. The TLS/SSL protocol will protect cleartext Basic Authentication credentials.

- Use Digest Authentication

  Replace Basic Authentication with the alternative Digest Authentication scheme. By modern cryptographic standards Digest Authentication is weak. But for a large range of purposes it is valuable as a replacement for Basic Authentication. It remedies some, but not all, weaknesses of Basic Authentication. See RFC 2617, section 4. Security Considerations for more information.

Request Header:

GET /admin HTTP/1.1 Referer: http://testhtml5.vulnweb.com User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.64 Safari/537.31 Accept: */* Host: testhtml5.vulnweb.com Connection: Close

Response Header:

HTTP/1.1 401 Unauthorized Connection: close Content-Length: 90 Content-Type: text/html; charset=utf-8 Date: Mon, 16 Jan 2017 10:21:29 GMT Server: nginx/1.4.1 WWW-Authenticate: Basic realm="Login Required"

Result:

Line No:6 Server: nginx/1.4.1
Line No:7 WWW-Authenticate: Basic realm="Login Required"

Line No:8

References:

- http://tools.ietf.org/html/rfc2617

Alert ID: 481470 Found on: 2017-01-16 Severity: High

Cross Site Scripting Vulnerability

Open Status: NEW **First Found:** 2017-01-16

URL: http://testhtml5.vulnweb.com/login
Method: POST
Vector: "><sCrIpT>alert(391700)</sCrIpT>
Cvss Score: 6.4
Cvss Vector: (AV:N/AC:L/Au:N/C:P/I:P/A:N)
PCI Compliance: Fail

Description:

The Web application is vulnerable to cross-site scripting (XSS), which allows attackers to take advantage of Web server scripts to inject JavaScript or HTML code that is executed on the client-side browser. This vulnerability is often caused by server-side scripts written in languages such as PHP, ASP, .NET, Perl or Java, which do not adequately filter data sent along with page requests or by vulnerable HTTP servers. This malicious code appears to come from your Web application when it runs in the browser of an unsuspecting user.

An attacker can do the following damage with an expoloit script:

- access other sites inside another client's private intranet
- steal another client's cookie(s)
- modify another client's cookie(s)
- steal another client's submitted form data
- modify another client's submitted form data before it reaches the server
- submit a form to your Web application on the user's behalf that modifies passwords or other application data

The two most common methods of attack are:

- Having a user click a URL link sent in an e-mail
- Having a user click a URL link while visiting a Web site

In both scenarios, the URL will generally link to the trusted site, but will contain additional data that is used to trigger the XSS attack.

Note that SSL connectivity does not protect against this issue.

Solution:

Fix Cross Site Scripting Vulnerability

Audit the affected url and other similar dynamic pages or scripts that could be relaying untrusted malicious data from the user input. In general, the following practices should be followed while developing dynamic web content:

- Explicitly set the character set encoding for each page generated by the web server
- Identify special characters
- Encode dynamic output elements
- Filter specific characters in dynamic elements

- Examine cookies

For more information on the above practices, read the following CERT advisory: CERT Advisory CA-2000-02

- For ASP.NET applications, the validateRequest attribute can be added to the page or the web.config. For example:

```
<%@ Page ... validateRequest="true" %>

OR

<system.web>
<pages validateRequest="true" />
</system.web>
```

  In addition, all dynamic content should be HTML encoded using HTTPUtility.HTMLEncode.

- For PHP applications, input data should be validated using functions such as strip_tags and utf8_decode. Dynamic content should be HTML encoded using htmlentities.

- For Perl applications, input data should be validated whenever possible using regular expressions. Dynamic content should be HTML encoded using HTML::Entities::encode or Apache::Util::html_encode (when using mod_perl).

Request Header:

POST /login HTTP/1.1 Referer: http://testhtml5.vulnweb.com User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.64 Safari/537.31 Accept: */* Content-Type: application/x-www-form-urlencoded Host: testhtml5.vulnweb.com Cookie: username="\"><sCrIpT>alert(391700)</sCrIpT>" Connection: Close username="><sCrIpT>alert(391700)</sCrIpT>&password=testpass@1234&submit=Submit

Response Header:

HTTP/1.1 200 OK Connection: close Access-Control-Allow-Origin: * Content-Length: 6011 Content-Type: text/html; charset=utf-8 Date: Mon, 16 Jan 2017 10:21:02 GMT Server: nginx/1.4.1

Response URL:

http://testhtml5.vulnweb.com/

Result:

Line No:53
Line No:54 Welcome "><sCrIpT>alert(391700)</sCrIpT> | Logout
Line No:55

References:

- http://www.us-cert.gov/cas/techalerts/CA-2000-02.html

- http://en.wikipedia.org/wiki/Cross_site_scripting

Alert ID: 481468 Found on: 2017-01-16 Severity: Medium

Form action submits sensitive data in the clear

Open Status: NEW **First Found:** 2017-01-16

URL: http://testhtml5.vulnweb.com
Method: POST
Cvss Score: 4.3
Cvss Vector: (AV:N/AC:M/Au:N/C:P/I:N/A:N)
PCI Compliance: Fail

Description:

A web form contains fields with data that is probably sensitive in nature. This form data is submitted over an unencrypted connection, which could allow hackers to sniff the network and view the data in plaintext.

Solution:

Use the HTTPS (HTTP over SSL) protocol to submit sensitive form data

Enable the HTTPS protocol on the server. Change the "action" URL of the form tag to use the HTTPS protocol ("https://...") instead of just the HTTP protocol ("http://..."). All sensitive data should be sent over HTTPS instead of over HTTP.

Request Header:

GET / HTTP/1.1 Host: testhtml5.vulnweb.com Referer: http://testhtml5.vulnweb.com User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; MDDS; InfoPath.2; .NET4.0C; .NET4.0E; Tablet PC 2.0)

Result:

Sensitive input transmitted without SSL: password

Alert ID: 481467 Found on: 2017-01-16 Severity: Low

Autocomplete enabled for sensitive HTML form fields

Open Status: NEW **First Found:** 2017-01-16

URL: http://testhtml5.vulnweb.com
Method: get
Cvss Score: 4.3
Cvss Vector: (AV:N/AC:M/Au:N/C:P/I:N/A:N)
PCI Compliance: Fail

Description:

The Web form contains passwords or other sensitive text fields for which the browser auto-complete feature is enabled. Auto-complete stores completed form field and passwords locally in the browser, so that these fields are filled automatically when the user visits the site again.

Sensitive data and passwords can be stolen if the user's system is compromised.

Note, however, that form auto-complete is a non-standard, browser-side feature that each browser handles differently. Opera, for example, disregards the feature, requiring the user to enter credentials for each Web site visit.

Solution:

Disable autocomplete for all sensitive fields

For each sensitive field in the HTML, set the "autocomplete" attribute to "off". For example:

```
<input type="password" autocomplete="off" name="pw">
```

If there are many fields, it may be faster to set the "autocomplete" attribute to "off" in the outer <form> tag. For example:

```
<form action="/login.jsp" autocomplete="off" name="pw">
<input type="password" name="pw">
</form>
```

Request Header:

GET / HTTP/1.1 Host: testhtml5.vulnweb.com Referer: http://testhtml5.vulnweb.com User-Agent: Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 6.1; Trident/4.0; SLCC2; .NET CLR 2.0.50727; .NET CLR 3.5.30729; .NET CLR 3.0.30729; Media Center PC 6.0; MDDS; InfoPath.2; .NET4.0C; .NET4.0E; Tablet PC 2.0)

Result:

Autocomplete enabled for sensitive form field: password

Alert ID: 481471 Found on: 2017-01-16 Severity: Low

Predictable resource location

Open Status: NEW **First Found:** 2017-01-16

URL: http://testhtml5.vulnweb.com/cgi-bin/

Method: get
Cvss Score: 6.4
Cvss Vector: (AV:N/AC:L/Au:N/C:P/I:P/A:N)
PCI Compliance: Fail

Description:

Rather than an actual vulnerability, this attack is informational, indicating that access to some resource is not granted. The resource is predictable and although it is not accessible via any URL links in the web application, probing using intelligent brute force methods or commonly used resource names indicates presence of the resource.

Solution:

A custom error page should be displayed to handle all such requests.

Request Header:

GET /cgi-bin HTTP/1.1 Referer: http://testhtml5.vulnweb.com User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.64 Safari/537.31 Accept: */* Host: testhtml5.vulnweb.com Connection: Close

Response Header:

HTTP/1.1 403 Forbidden Connection: close Vary: Accept-Encoding Content-Length: 263 Content-Type: text/html; charset=iso-8859-1 Date: Mon, 16 Jan 2017 10:21:07 GMT Server: nginx/1.4.1

Result:

http://testhtml5.vulnweb.com/cgi-bin/

References:

- http://projects.webappsec.org/w/page/13246953/Predictable%20Resource%20Location

Alert ID: 481473 Found on: 2017-01-16 Severity: Low

Predictable resource location

Open Status: NEW **First Found:** 2017-01-16

URL: http://testhtml5.vulnweb.com/cgi-bin
Method: get
Cvss Score: 6.4
Cvss Vector: (AV:N/AC:L/Au:N/C:P/I:P/A:N)
PCI Compliance: Fail

Description:

Rather than an actual vulnerability, this attack is informational, indicating that access to some resource is not granted. The resource is predictable and although it is not accessible via any URL links in the web application, probing using intelligent brute force methods or commonly used resource names indicates presence of the resource.

Solution:

A custom error page should be displayed to handle all such requests.

Request Header:

GET /cgi-bin HTTP/1.1 Referer: http://testhtml5.vulnweb.com User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.64 Safari/537.31 Accept: */* Host: testhtml5.vulnweb.com Connection: Close

Response Header:

HTTP/1.1 403 Forbidden Connection: close Vary: Accept-Encoding Content-Length: 263 Content-Type: text/html; charset=iso-8859-1 Date: Mon, 16 Jan 2017 10:21:35 GMT Server: nginx/1.4.1

Result:

http://testhtml5.vulnweb.com/cgi-bin

References:

- http://projects.webappsec.org/w/page/13246953/Predictable%20Resource%20Location

Alert ID: 481474 Found on: 2017-01-16 Severity: Low

Predictable resource location

Open Status: NEW **First Found:** 2017-01-16

URL: http://testhtml5.vulnweb.com/static/app/
Method: get
Cvss Score: 6.4
Cvss Vector: (AV:N/AC:L/Au:N/C:P/I:P/A:N)
PCI Compliance: Fail

Description:

Rather than an actual vulnerability, this attack is informational, indicating that access to some resource is not granted. The resource is predictable and although it is not accessible via any URL links in the web application, probing using intelligent brute force methods or commonly used resource names indicates presence of the resource.

Solution:

A custom error page should be displayed to handle all such requests.

Request Header:

GET /static/app HTTP/1.1 Referer: http://testhtml5.vulnweb.com User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.64 Safari/537.31 Accept: */* Host: testhtml5.vulnweb.com Connection: Close

Response Header:

HTTP/1.1 403 Forbidden Connection: close Content-Length: 570 Content-Type: text/html Date: Mon, 16 Jan 2017 10:21:39 GMT Server: nginx/1.4.1

Result:

http://testhtml5.vulnweb.com/static/app/

References:

- http://projects.webappsec.org/w/page/13246953/Predictable%20Resource%20Location

Alert ID: 481475 Found on: 2017-01-16 Severity: Low

Predictable resource location

Open Status: NEW **First Found:** 2017-01-16

URL: http://testhtml5.vulnweb.com/static/css/
Method: get
Cvss Score: 6.4
Cvss Vector: (AV:N/AC:L/Au:N/C:P/I:P/A:N)
PCI Compliance: Fail

Description:

Rather than an actual vulnerability, this attack is informational, indicating that access to some resource is not granted. The resource is predictable and although it is not accessible via any URL links in the web application, probing using intelligent brute force methods or commonly used resource names indicates presence of the resource.

Solution:

A custom error page should be displayed to handle all such requests.

Request Header:

GET /static/css HTTP/1.1 Referer: http://testhtml5.vulnweb.com User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.64 Safari/537.31 Accept: */* Host: testhtml5.vulnweb.com Connection: Close

Response Header:

HTTP/1.1 403 Forbidden Connection: close Content-Length: 570 Content-Type: text/html Date: Mon, 16 Jan 2017 10:21:42 GMT Server: nginx/1.4.1

Result:

http://testhtml5.vulnweb.com/static/css/

References:

- http://projects.webappsec.org/w/page/13246953/Predictable%20Resource%20Location

Alert ID: 481476 Found on: 2017-01-16 Severity: Low

Predictable resource location

Open Status: NEW **First Found:** 2017-01-16

URL: http://testhtml5.vulnweb.com/static/app/libs/
Method: get
Cvss Score: 6.4
Cvss Vector: (AV:N/AC:L/Au:N/C:P/I:P/A:N)
PCI Compliance: Fail

Description:

Rather than an actual vulnerability, this attack is informational, indicating that access to some resource is not granted. The resource is predictable and although it is not accessible via any URL links in the web application, probing using intelligent brute force methods or commonly used resource names indicates presence of the resource.

Solution:

A custom error page should be displayed to handle all such requests.

Request Header:

GET /static/app/libs HTTP/1.1 Referer: http://testhtml5.vulnweb.com User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.64 Safari/537.31 Accept: */* Host: testhtml5.vulnweb.com Connection: Close

Response Header:

HTTP/1.1 403 Forbidden Connection: close Content-Length: 570 Content-Type: text/html Date: Mon, 16 Jan 2017 10:21:45 GMT Server: nginx/1.4.1

Result:

http://testhtml5.vulnweb.com/static/app/libs/

References:

- http://projects.webappsec.org/w/page/13246953/Predictable%20Resource%20Location

Alert ID: 481477 Found on: 2017-01-16 Severity: Low

Predictable resource location

Open Status: NEW **First Found:** 2017-01-16

URL: http://testhtml5.vulnweb.com/static/app/controllers/
Method: get
Cvss Score: 6.4
Cvss Vector: (AV:N/AC:L/Au:N/C:P/I:P/A:N)
PCI Compliance: Fail

Description:

Rather than an actual vulnerability, this attack is informational, indicating that access to some resource is not granted. The resource is predictable and although it is not accessible via any URL links in the web application, probing using intelligent brute force methods or commonly used resource names indicates presence of the resource.

Solution:

A custom error page should be displayed to handle all such requests.

Request Header:

GET /static/app/controllers HTTP/1.1 Referer: http://testhtml5.vulnweb.com User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.64 Safari/537.31 Accept: */* Host: testhtml5.vulnweb.com Connection: Close

Response Header:

HTTP/1.1 403 Forbidden Connection: close Content-Length: 570 Content-Type: text/html Date: Mon, 16 Jan 2017 10:21:45 GMT Server: nginx/1.4.1

Result:

http://testhtml5.vulnweb.com/static/app/controllers/

References:

- http://projects.webappsec.org/w/page/13246953/Predictable%20Resource%20Location

Alert ID: 481478 Found on: 2017-01-16 Severity: Low

Predictable resource location

Open Status: NEW **First Found:** 2017-01-16

URL: http://testhtml5.vulnweb.com/static/app/services/
Method: get
Cvss Score: 6.4
Cvss Vector: (AV:N/AC:L/Au:N/C:P/I:P/A:N)
PCI Compliance: Fail

Description:

Rather than an actual vulnerability, this attack is informational, indicating that access to some resource is not granted. The resource is predictable and although it is not accessible via any URL links in the web application, probing using intelligent brute force methods or commonly used resource names indicates presence of the resource.

Solution:

A custom error page should be displayed to handle all such requests.

Request Header:

GET /static/app/services HTTP/1.1 Referer: http://testhtml5.vulnweb.com User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.64 Safari/537.31 Accept: */* Host: testhtml5.vulnweb.com Connection: Close

Response Header:

HTTP/1.1 403 Forbidden Connection: close Content-Length: 570 Content-Type: text/html Date: Mon, 16 Jan 2017 10:21:50 GMT Server: nginx/1.4.1

Result:

http://testhtml5.vulnweb.com/static/app/services/

References:

- http://projects.webappsec.org/w/page/13246953/Predictable%20Resource%20Location

Alert ID: 481479 Found on: 2017-01-16 Severity: Low

Predictable resource location

Open Status: NEW **First Found:** 2017-01-16

URL: http://testhtml5.vulnweb.com/static/img/
Method: get
Cvss Score: 6.4
Cvss Vector: (AV:N/AC:L/Au:N/C:P/I:P/A:N)

PCI Compliance: Fail

Description:

Rather than an actual vulnerability, this attack is informational, indicating that access to some resource is not granted. The resource is predictable and although it is not accessible via any URL links in the web application, probing using intelligent brute force methods or commonly used resource names indicates presence of the resource.

Solution:

A custom error page should be displayed to handle all such requests.

Request Header:

GET /static/img HTTP/1.1 Referer: http://testhtml5.vulnweb.com User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.64 Safari/537.31 Accept: */* Host: testhtml5.vulnweb.com Connection: Close

Response Header:

HTTP/1.1 403 Forbidden Connection: close Content-Length: 570 Content-Type: text/html Date: Mon, 16 Jan 2017 10:21:52 GMT Server: nginx/1.4.1

Result:

http://testhtml5.vulnweb.com/static/img/

References:

- http://projects.webappsec.org/w/page/13246953/Predictable%20Resource%20Location

Alert ID: 481469 Found on: 2017-01-16 Severity: Info

Web server version disclosure

Open Status: NEW **First Found:** 2017-01-16

URL: http://testhtml5.vulnweb.com/Crossdomain.xml
Method: get
Cvss Base: 1.0
Cvss Score: 1.0
PCI Compliance: Pass

Description:

HTTP web server information is disclosed in HTTP headers. This information may revel software name, version etc. It may help an attacker to look for specific web server version related vulnerabilities.

Solution:

Versions and types information should be omitted where possible.

Request Header:

GET /Crossdomain.xml HTTP/1.1 Referer: http://testhtml5.vulnweb.com User-Agent: Mozilla/5.0 (Windows NT 6.1) AppleWebKit/537.31 (KHTML, like Gecko) Chrome/26.0.1410.64 Safari/537.31 Accept: */* Host: testhtml5.vulnweb.com Connection: Close

Response Header:

HTTP/1.1 404 NotFound Connection: close Content-Length: 238 Content-Type: text/html Date: Mon, 16 Jan 2017 10:20:56 GMT Server: nginx/1.4.1

Result:

nginx/1.4.1

References:

- http://osvdb.org/91