



1. VULNERABILITY TO PENETRATION ASSESSMENT

Report Subject: example.com

Scan Date: April 12, 2010 between 10:00 and 12:00 PDT

This VPA report has been prepared by **Jaal LLC**, an Internet security startup located in Riverside, CA, for example.com. This report provides details about security concerns that have been identified via a VPA scan of the example.com web pages.

**STRICTLY CONFIDENTIAL MATERIAL
NOT FOR PUBLIC DISTRIBUTION**

It is strongly advised to read the disclaimer provided at the end of this document before proceeding with any actions based on the findings of this report.

Thank you,
The stopthehacker.com Team
<http://www.stopthehacker.com/>



2. OVERVIEW

The example.com web pages were scanned on the 12th of April, 2010, 10:00 to 12:00 PDT approximately. The standard policies of scanning, as described in Robots.txt, were respected and no attempts were made to use brute force to gain access to restricted (login) pages.

Note that each vulnerability in OSVDB is considered an invitation to a security breach by the security community.

Several safety and conformance issues have been identified and are presented in this report. For ease of presentation, we group vulnerabilities into: (a) server level analysis, and (b) software level analysis. When appropriate, we associate each vulnerability that we found with the commonly-used identifier found in the Open Source Vulnerability Database [OSVDB] and also color code them.

Critical: 5

Critical vulnerabilities of the highest importance. These need to be fixed immediately.

Important: 1

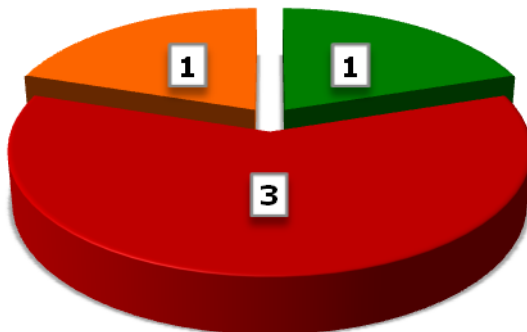
Important vulnerabilities which are not critical, but can be used by a malicious hacker to cause significant harm.

Informational: 1

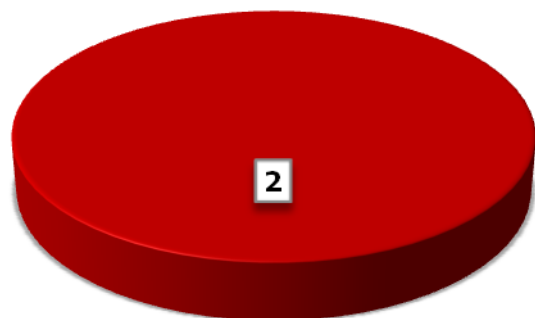
Information which is helpful for system administrators and webmasters to verify the correct functioning of their server.

3. VULNERABILITES DETECTED

Server Vulnerabilities



Application Vulnerabilities





4. SERVER LEVEL ANALYSIS

4.1. Critical Vulnerabilities

4.1.1. Server: Apache/2.2.9 (Unix)

- Server Tokens: Apache/2.2.9 (Unix) mod_ssl/2.2.9 OpenSSL/0.9.8e-fips-rhel5 mod_auth_passthrough/2.1 mod_bwlimited/1.4 FrontPage/5.0.2.2635 PHP/5.2.6
- **Note:** You can set the directive 'ServerTokens Prod' to limit the information emanating from the server in its response headers.
- Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE
- **HTTP method ('Allow' Header):** 'TRACE' is typically only used for debugging and should be disabled. Leaving this option enabled allows an attacker to map out internal infrastructure. An attacker can glean information such as presence of load-balancing servers, redirection-servers and other equipment by exploiting this option. [**OSVDB-877,Trace-Hack**]

Details:

To disable these methods, add the following lines for each virtual host in your configuration file:

```
RewriteEngine on  
RewriteCond %{REQUEST_METHOD}^(TRACE|TRACK)  
RewriteRule .* - [F]
```

Alternatively, note that Apache versions 1.3.34, 2.0.55, and 2.2 support disabling the TRACE method natively via the 'TraceEnable' directive.

Recommendation: 'TRACE' option should be disabled.

4.1.2. Outdated server configuration

Attackers may be able to compromise the system based on older versions of software which are installed on the server [**OSVDB-396, Outdated**]

Outdated software installations:

- **mod_ssl/2.2.9 appears to be outdated (current is at least 2.8.31)**
- **OpenSSL/0.9.8e-fips-rhel5 appears to be outdated (current is at least 0.9.8g)**
- **FrontPage/5.0.2.2635 appears to be outdated (current is at least 5.0.4.3)**
- **PHP/5.2.6 appears to be outdated (current is at least 5.2.6RC4)**

```
mod_ssl / 2.2.9 OpenSSL / 0.9.8e -fips-rhel5 mod_auth_passthrough / 2.1 mod_bwlimited/1.4  
FrontPage/5.0.2.2635
```

PHP/5.2.6 - mod_ssl 2.8.7 and lower are vulnerable to a remote buffer overflow which may allow a remote



shell.

Recommendation: These software packages need to be upgraded immediately.

4.1.3. Apache Frontpage module vulnerability

The remote host is using the Apache mod_frontpage module. The mod_frontpage module older than 1.6.1 is vulnerable to a buffer overflow, which may allow an attacker to gain root access. [CVE : CAN-2002-0427]

Recommendation: If not critical, disable the module immediately.

4.2. Important Vulnerabilities

4.2.1. Potential user enumeration vulnerability

The remote web server is set up in a manner that it is possible to enumerate remote users on the web server. This can help an attacker to launch user specific and/or password cracking attacks. [Usrenum-vuln, CVE : CAN-2001-1013]

Details:

Enumeration of users is possible by requesting ~username (server responds with 'Forbidden' for users, 'not found' for non-existent users).

The information leak occurs on Apache based web servers whenever the UserDir module is enabled.

Recommendation: Disable this feature by changing 'UserDir public_html' (or whatever) to 'UserDir disabled'.

Or, use a RedirectMatch rewrite rule under Apache -- this works even if there is no such entry in the password file, e.g.:

```
RedirectMatch ^/~(.*)$ http://my-target-webserver.somewhere.org/$1
```

Or, add into httpd.conf:

```
ErrorDocument 404 http://localhost/sample.html
```

```
ErrorDocument 403 http://localhost/sample.html
```

(Note: Please use a FQDN inside the URL).

4.3. Informational Notices

4.3.1. List of Open Ports



- domain (53/tcp)
- general/tcp
- general/udp
- domain (53/udp)
- nbx-ser (2095/tcp)
- imap (143/tcp)
- infowave (2082/tcp)
- ftp (21/tcp)
- http (80/tcp)
- imaps (993/tcp)
- mysql (3306/tcp)
- eli (2087/tcp)
- unknown (26/tcp)
- nbx-dir (2096/tcp)
- pop3 (110/tcp)
- pop3s (995/tcp)
- urd (465/tcp)
- radsec (2083/tcp)
- gnunet (2086/tcp)
- https (443/tcp)
- ssh (22/tcp)
- smtp (25/tcp)



5. SOFTWARE LEVEL ANALYSIS

5.1. Critical Vulnerabilities

5.1.1. Potential XSS vulnerability

The following XSS vulnerabilities were identified on the main website.

Page: <http://example.com/invite.html>

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<xml id="X"><a><script>document.vulnerable=true;</script>;</xml>`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<xml src="javascript:document.vulnerable=true;">`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `[ ][ BC]script>document.vulnerable=true;[ ][ BC]/script>`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<div datafld="b" dataformatas="html" datasrc="#X"></div>`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: ``

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: ` onmouseover="document.vulnerable=true;">`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<!-- -- --><script>document.vulnerable=true;</script><!-- -- -->`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<![CDATA[<!--]]<script>document.vulnerable=true;!--></script>`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<<script>document.vulnerable=true;</script>`



The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<style><!--</style><script>document.vulnerable=true;!--></script>`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<object classid="clsid:..." codebase="javascript:document.vulnerable=true;">`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<style type="text/javascript">document.vulnerable=true;</style>`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<div style="width: expression(document.vulnerable=true);">`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<div style="binding: url([link to code]);">`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<div style="behaviour: url([link to code]);">`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<div style="background-image: url(javascript:document.vulnerable=true);">`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<body onload="document.vulnerable=true;">`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<a href="about:<script>document.vulnerable=true;</script>">`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: ``

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: ``



The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: <link rel="stylesheet" href="javascript:document.vulnerable=true;">

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value:

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: &{document.vulnerable=true;};

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: &<script>document.vulnerable=true;</script>

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: <bgsound src="javascript:document.vulnerable=true;">

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: <input type="image" dynsrc="javascript:document.vulnerable=true;">

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value:

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value:

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: <div onmouseover="document.vulnerable=true;">

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value:

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: <HEAD><META HTTP-EQUIV="CONTENT-TYPE" CONTENT="text/html; charset=UTF-7"> </HEAD>+ADw-SCRIPT+AD4-document.vulnerable=true;+ADw-/SCRIPT+AD4-



The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<? echo('<SCR');echo('IPT>document.vulnerable=true</SCRIPT>'); ?>`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<HTML><BODY><?xml:namespace prefix="t" ns="urn:schemas-microsoft-com:time"><?import namespace="t" implementation="#default#time2"><t:set attributeName="innerHTML" to="XSS<SCRIPT DEFER>document.vulnerable=true</SCRIPT>"></BODY></HTML>`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<XML ID="xss"><I><IMG SRC="javas<!-- -->cript:document.vulnerable=true"></I></XML>`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<XML ID=I><X><C><![CDATA[<IMG SRC="javas]]<![CDATA[cript:document.vulnerable=true;"]]</C></X></xml>`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<OBJECT classid=clsid:ae24fdae-03c6-11d1-8b76-0080c744f389><param name=url value=javascript:document.vulnerable=true></OBJECT>`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<BASE HREF="javascript:document.vulnerable=true;/">`

Page: <http://example.com/contact.html>

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<xml id="X"><a><script>document.vulnerable=true;</script>;</xml>`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `[\xC0][\xBC]script>document.vulnerable=true;[\xC0][\xBC]/script>`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<div datafld="b" dataformatas="html" datasrc="#X"></div>`



The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: <xml src="javascript:document.vulnerable=true;">

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: onmouseover="document.vulnerable=true;">

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value:

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: <!-- -- --><script>document.vulnerable=true;</script><!-- -- -->

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: <![CDATA[<!--]]><script>document.vulnerable=true;//--></script>

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: <<script>document.vulnerable=true;</script>

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: <style><!--</style><script>document.vulnerable=true;//--></script>

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: <object classid="clsid:..." codebase="javascript:document.vulnerable=true;">

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: <style type="text/javascript">document.vulnerable=true;</style>

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: <div style="width: expression(document.vulnerable=true);">

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: <div style="binding: url([link to code]);">



The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<div style="background-image: url(javascript:document.vulnerable=true);">`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<div style="behaviour: url([link to code]);">`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<body onload="document.vulnerable=true;">`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<a href="about:<script>document.vulnerable=true;</script>">`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: ``

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: ``

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<link rel="stylesheet" href="javascript:document.vulnerable=true;">`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: ``

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `&{document.vulnerable=true;};`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `&<script>document.vulnerable=true;</script>`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<bgsound src="javascript:document.vulnerable=true;">`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to



this XSS string.

Tested value: `<input type="image" dynsrc="javascript:document.vulnerable=true;">`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: ``

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: ``

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: `<div onmouseover="document.vulnerable=true;">`

The unencoded attack string was found in the html of the document. Some browsers may be vulnerable to this XSS string.

Tested value: ``

Recommendation: Implement anti XSS mechanisms immediately.

5.1.2. Potential SQL injection vulnerability

The following SQL injection vulnerabilities were identified on the main website.

Page: <http://example.com/invite.html>

Variable: required

Submitted Form State:

- recipient: user@example.com
- subject: example.com - Invitation
- redirect: http://www.example.com/thanks_invitation.html
- Event_purpose:
- Event_date:
- Number_people:
- realname:
- Company_name:
- Address:
- City:
- State:
- Zip_code:



- Phone_number:
- email:
- Other_comments:
- unnamed field: Submit Speaking Request

Results:

Server Status Code: 302 Found

Tested value: %31%27%20%4F%52%20%27%31%27%3D%27%31;

Server Status Code: 302 Found

Tested value: %49%39%32%79%82%32%39%49%39%61%39%49

Page: <http://example.com/contact.html>

Variable: required

Submitted Form State:

- recipient: user@example.com
- subject: example.com - Contact
- redirect: http://www.example.com/thanks_contact.html
- realname:
- Company_name:
- Address:
- City:
- State:
- Zip_code:
- Phone_number:
- email:
- Other_comments:
- unnamed field: Submit

Results:

Server Status Code: 302 Found

Tested value: %49%39%32%79%82%32%39%49%39%61%39%49

Server Status Code: 302 Found

Tested value: %31%27%20%4F%52%20%27%31%27%3D%27%31;

Recommendation: Implement anti SQL injection mechanisms immediately.



REFERENCES

- [OSVDB] The Open Source Vulnerability Database, <http://osvdb.org>
[Trace-Hack] http://www.cgisecurity.com/whitehat-mirror/WhitePaper_screen.pdf
[Outdated] <http://cve.mitre.org/cgi-bin/cvename.cgi?name=CAN-2002-0082>.
[Usrenum-vuln] <http://www.securiteam.com/unixfocus/5WP0C1F5FI.html>
[CVE] Common Vulnerabilities and Exposures, <http://cve.mitre.org>



6. DISCLAIMER

The content in this file, including domain names, trademarks, monetization partners, and other information, is provided by Jaal and its third party content providers for your personal information only, and is not intended to be relied upon for any action or decision. This information is gathered through automated means and is subject to change. Content in this file is not appropriate for the purposes of making a determination to pursue legal claims regarding a particular domain or making any determination on whether a particular domain is confusingly similar to a registered trademark. Nor does Jaal provide any form of advice amounting to legal advice, or make any recommendations regarding particular domains and whether they may violate any rule or regulation. Such a determination is complex and should be made with the advice of competent legal counsel.

Neither Jaal nor its third party content providers shall be liable for any errors, inaccuracies or delays in content, or for any actions taken in reliance thereon.

JAAL EXPRESSLY DISCLAIMS ALL WARRANTIES, EXPRESSED OR IMPLIED, AS TO THE ACCURACY OF ANY THE CONTENT PROVIDED, OR AS TO THE FITNESS OF THE INFORMATION FOR ANY PURPOSE.