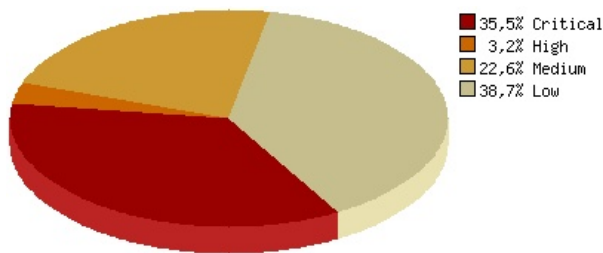


Summary

Site	www.example.com
Start date	2011-08-27 11:01:38
Scan duration	00:51:01
IP	192.0.43.10
Server	Apache
Powered-By	Unknown
No. of Low severity vulnerabilities	12
No. of Medium severity vulnerabilities	7
No. of High severity vulnerabilities	1
No. of Critical severity vulnerabilities	11
No. of URLs scanned	9123
No. of Tests done	25491
Connection errors	7
Total data received	28651Kb



Disclaimer

Please note that while we take best efforts to provide you with the best possible results, some information may be incorrect or lacking due to server configuration, internet connection or other causes and some results may not appear, or appear as errors while they are not. Therefore, if you find any mistake, or any information you think needs to be corrected, please notify us immediately at support@zerodayscan.com and we will inspect the case.

Detected Apache mod_security module

Description:

While performing security scan of your webserver, scanner detected the mod_security Apache module. This module blocks some of the vulnerability assessment attacks performed by ZeroDayScan. As a result, some of the vulnerabilities might not be detected. To achieve better results with security scanning, it is recommended to disable mod_security during web site security assessment.

There are some ways to obfuscate web security attacks for example to obfuscate SQL injection attacks so the mod_security will not detect them. We try to limit a number of injection attacks to a minimal, as a result we do not obfuscate web security assessment attacks. We want the scanner to run as fast as possible, so, during web security assessment, it is recommended to disable mod_security.

Solution:

To achieve better results with security scanning, it is recommended to disable mod_security during web site security assessment.

Hosts Sharing this ip: 192.0.43.10

The following list shows a list of domains that use the same IP address as a target domain. Only the target domain was scanned.

- <http://www.cislgroup.com/>
- <http://www.cislgruppen.com/>
- <http://www.crowntrophymck.com/>
- <http://www.etchapod.com/>
- <http://www.example.com/>
- <http://www.example.net/>
- <http://www.example.org/>
- <http://www.frameduck.com/>
- <http://www.harknesstables.com/>
- <http://www.jas-benelux.com/>
- <http://www.johnwarburton.net/>
- <http://www.ourspacenz.com/>
- <http://www.piximity.com/>
- <http://www.textile.org.ua/>

OS Commanding

Description:

OS Commanding vulnerability allows the malicious users to execute OS commands using vulnerability in the web script. This vulnerability happens when data submitted by the site visitors is not correctly verified. Such vulnerability is one of the most dangerous possible vulnerabilities. Malicious users can install a backdoor, or a root kit on your webserver. Once in control, they can use your servers to do whatever they wish like sending spam and/or attacking other destinations from your servers. If this happens and you end up on a blacklist, you may find that your legitimate incoming and outgoing traffic is rejected. As a result, you'll lose sales, as well as valuable channels of communication with your customers, and end up involved in lengthy, frustrating negotiations to get off the black lists.

Solution:

Never trust user submitted data. Go over your code and make sure that all user submitted data is sanitized and checked for validity before using it.

#	URL	Severity
1	POST Request: http://www.example.com/vulnerabilities/exec/ [ip=%7Ccat%20/etc/passwd&submit=submit]	critical
Post Injected Parameter: ip= cat /etc/passwd		Pattern found: root:

SQL Injection

Description:

SQL Injection vulnerability can be exploited by the attacker to retrieve and change database contents, pass the login page and execute shell commands on the remote server. This vulnerability happens when data submitted by the site's visitors is not correctly verified and is glued together with a legitimate SQL query. Hackers use this technique to inject additional commands to the SQL server.

Basically such vulnerability is among the most dangerous of possible vulnerabilities. Using SQL injection, hackers can gain access to your web server and install a back door, or a root kit. Once in control, they can use your servers to do whatever they wish like sending spam and/or attacking other destinations from your servers. If this happens and you end up on a blacklist, you may find that your legitimate incoming and outgoing traffic is rejected. You'll lose sales, as well as valuable channels of communication with your customers, and end up involved in lengthy, frustrating negotiations to get off the blacklists.

Solution:

Fix application code to make sure it does not contain code that can be exploited using SQL injection attacks. Additional recommendation is to use a database firewall such as GreenSQL - <http://www.greensql.net/> .

Never trust data submitted by end user. Use a whitelist approach, just allow expected characters. For example, if the user submits the name of the file, make sure it contains legit chars only and not chars that for example can be used to change the predefined directory.

#	URL	Severity
10	http://www.example.com/vulnerabilities/xss_r/?name=%3C%3Fphp%20%24c%3D%27php%27%3B%24c.%3D%27info%27%3B%24c%28%29%3B%3F%3E	critical
Query Injected Parameter: name=<?php \$c='php';\$c.='info';\$c();?>		Pattern found: phpinfo
11	POST Request: http://www.example.com/vulnerabilities/xss_s/[txtName=aaaa&mtxMessage=%3C%3Fphp%20%24c%3D%27php%27%3B%24c.%3D%27info%27%3B%24c%28%29%3B%3F%3E&btnSign=Sign+Guestbook]	critical
Post Injected Parameter: mtxMessage=<?php		Pattern found: phpinfo
12	POST Request: http://www.example.com/security.php[security=%3C%3Fphp%20%24c%3D%27php%27%3B%24c.%3D%27info%27%3B%24c%28%29%3B%3F%3E&seclv_submit=Submit]	critical
Post Injected Parameter: security=<?php \$c='php';\$c.='info';\$c();?>		Pattern found: phpinfo
13	http://www.example.com/vulnerabilities/sqli_blind/?id=%3C%3Fphp%20%24c%3D%27php%27%3B%24c.%3D%27info%27%3B%24c%28%29%3B%3F%3E&Submit=Submit	critical
Query Injected Parameter: id=<?php \$c='php';\$c.='info';\$c();?>		Pattern found: phpinfo
14	http://www.example.com/vulnerabilities/csrf/?password_new=%3C%3Fphp%20%24c%3D%27php%27%3B%24c.%3D%27info%27%3B%24c%28%29%3B%3F%3E&password_conf=Q1s2x3w4d&Change=Change	critical
Query Injected Parameter: password_new=<?php		Pattern found: phpinfo
15	POST Request: http://www.example.com/setup.php[create_db=%3C%3Fphp%20%24c%3D%27php%27%3B%24c.%3D%27info%27%3B%24c%28%29%3B%3F%3E]	critical
Post Injected Parameter: create_db=<?php \$c='php';\$c.='info';\$c();?>		Pattern found: phpinfo

Directory Listing

Description:

Directory listing check looks for directories where directory content is printed. Directories reported can be viewed by any visitor. It is vulnerable in the sense that these directories can contain configuration, private and backup files which can be used by the attackers to take your server under control.

Solution:

Simply put an empty index.html file in the reported directories. If you use Apache webserver you can add the following directive to Apache server configuration or to .htaccess file: "Options -Indexes".

#	URL	Severity
16	http://www.example.com/dvwa/images/	low
Found: Index of /dvwa/images		
17	http://www.example.com/dvwa/	low
Found: Index of /dvwa		
18	http://www.example.com/dvwa/includes/	low
Found: Index of /dvwa/includes		
19	http://www.example.com/dvwa/js/	low
Found: Index of /dvwa/js		
20	http://www.example.com/dvwa/css/	low
Found: Index of /dvwa/css		
21	http://www.example.com/dirlist2/	low
Found: Index of /dirlist2		
22	http://www.example.com/vulnerabilities/	low
Found: Index of /vulnerabilities		

#	URL	Severity
23	http://www.example.com/dirlist/	low
Found: Index of /dirlist		
24	http://www.example.com/config/	low
Found: Index of /config		

Archive File

Description:

Archive file check looks for archive files. For example xxx.zip or yyy.rar files. We use some heuristics methods to find such files. These archive files can contain application code, or other sensitive information like database connection string, user names and passwords, IP addresses and other information that can be used by the attackers.

Solution:

Remove archive files from the production server.

#	URL	Severity
25	http://www.example.com/dirlist2.tar.gz	high

URL Redirection Vulnerability

Description:

URL redirection vulnerability happens when a script redirects the user to a third party website submitted as one of the script's parameters. It is considered the security vulnerability because the user can be redirected to a malicious website submitted such way. This vulnerability is exploited as part of the phishing attacks. Fraudsters send bunch of emails to the users with a specially crafted link that includes a legitimate website together with a link to malicious website. When a user clicks on that link, he is automatically redirected to legitimate website and then to the malicious website. That malicious website can pretend to be an original legitimate service. In many cases user's computer got infected with a banker Trojan and/or virus.

Solution:

Lock down your scripts to redirect only to legitimate resource. For example allow redirection only to pages located in your website.

#	URL	Severity
26	http://www.example.com/4-redirect.php?url=http%3A//www.zerodayscan.com/scan/pinfo.txt%3F%2500?%00	medium
Query Injected Parameter:		
27	http://www.example.com/4-redirect.php?url=http%3A//www.zerodayscan.com/scan/pinfo.txt%2500	medium
Query Injected Parameter:		
28	http://www.example.com/4-redirect.php?url=http%3A//www.zerodayscan.com/scan/pinfo.txt	medium
Query Injected Parameter:		

Email Disclosure

Description:

Strictly talking email disclosure is not considered as security vulnerability. Zero Day Scanner looks for emails while performing security scan. Potentially such emails can be collected by the spammers. Spammers use special programs that scan Internet websites and extract email address. Additional attack vector is password bruteforce attempt. Name of the company's employees and their email address is very helpful for password bruteforce attack because email address is used as a login name and not all people are using complicated passwords.

Solution:

A best solution is not to display email addresses at all. Add a link to contact page instead. An additional workaround is to use image with the email address instead of the text.

#	URL	Severity
29	dvwa@dvwa.co.uk: http://www.example.com/instructions.php	low
30	dvwa@dvwa.co.uk: http://www.example.com/README	low

#	URL	Severity
31	dvwa@dvwa.co.uk: http://www.example.com/README.txt	low

Open Ports

Description:

Open ports check detects open ports on your web server. It is recommended to keep this list as small as possible. Each port presents an additional attack vector on your server and it is recommended to keep this list short.

Solution:

Go over a list of ports and remove not necessary or not required ports.

- 22/tcp open ssh
- 25/tcp open smtp
- 53/tcp open domain
- 80/tcp open http
- 587/tcp open submission
- 3306/tcp open mysql

Whois

```
* Whois Server Version 2.0
* Domain names in the .com and .net domains can now be registered
* with many different competing registrars. Go to http://www.internic.net
* for detailed information.
*   Server Name: EXAMPLE.COM.RAFAELYALUFF.COM
*   IP Address: 173.203.204.123
*   Registrar: DOMAIN.COM, LLC
*   Whois Server: whois.domain.com
*   Referral URL: http://www.domain.com
*   Server Name: EXAMPLE.COM.AU
*   Registrar: ENETICA PTY LTD
*   Whois Server: whois.enetica.com.au
*   Referral URL: http://www.enetica.com.au
*   Domain Name: EXAMPLE.COM
*   Registrar: RESERVED-INTERNET ASSIGNED NUMBERS AUTHORITY
*   Whois Server: whois.iana.org
*   Referral URL: http://res-dom.iana.org
*   Name Server: A.IANA-SERVERS.NET
*   Name Server: B.IANA-SERVERS.NET
*   Status: clientDeleteProhibited
*   Status: clientTransferProhibited
*   Status: clientUpdateProhibited
*   Updated Date: 14-aug-2011
*   Creation Date: 14-aug-1995
*   Expiration Date: 13-aug-2012
* >>> Last update of whois database: Sat, 27 Aug 2011 15:00:50 UTC <<<
* NOTICE: The expiration date displayed in this record is the date the
* registrar's sponsorship of the domain name registration in the registry is
* currently set to expire. This date does not necessarily reflect the expiration
* date of the domain name registrant's agreement with the sponsoring
* registrar. Users may consult the sponsoring registrar's Whois database to
* view the registrar's reported date of expiration for this registration.
* The Registry database contains ONLY .COM, .NET, .EDU domains and
* Registrars.% IANA WHOIS server
* % for more information on IANA, visit http://www.iana.org
* % This query returned 1 object
* domain:          EXAMPLE.COM
* organisation:    Internet Assigned Numbers Authority
* created:         1992-01-01
* source:         IANA
```