WEBMIN (XSS BUG)

REMOTE ARBITARY FILE DISCLOSURE

By

Umair Manzoor (Umz)
umz32.dll@gmail.com

ABSTRACT:

WEBMIN is a system configuration tool for Unix-like systems. It has a web-based interface for configuring all the internals of the operating system. WEBMIN is basically use for Remote Administration of Servers. Remote Administration refers to any method of controlling a computer from a remote location. It consist of a simple web server, and a number of CGI programs which directly update system files. All the modules and web server is coded in PERL 5.Almost all of the development is done by Jamie Cameron along with other individuals who provide patches and additional modules.

Why there is a need of remote administration? Companies like ISPs and Telecoms need this facility to deal with their servers in case of faults. Fault can be define as "An incorrect step, process, or data definition in a computer program" as we all know that software can posses more faults then hardware because they are much vulnerable to user abuse. Most of the time software failures (Denial of Service) are caused by faults which then leads to system failure in some cases. Some time advantage can be taken of these faults to gain access to the system or may occur in leakage of sensitive information. Similar type of fault (bug) was detected in WEBMIN which allows an arbitrary file disclosure of the server that may cause an unauthorized access to the server. The bug is specifically named as Cross-site-scripting bug (XSS) which I exploited can threat to gain access to the system or possibly get sensitive information like DNS entries/account details etc. In this paper I will discuss how can we use these types of bugs for our sole purposes, possible threats to these types of bugs and how to beat them. I will also detail the procedure of exploit I coded and how to test these types of vulnerabilities before using these softwares.

There are many possible solutions to this problem but as a matter of fact no system is secured if it has number of services running on it.

NETWORK[1]:

This term refers to the computers which can be used for person to person communication and can share information, softwares, peripheral devices, and/or processing power. These networks may be fixed (cabled, permanent) or temporary (as via modems).

History of Networks:

In 1940 George_Stibitz used a machine to send a set of instructions from New Hampshire to his COMPLEX NUMBER COMPUTER at New York and received the results by same means of communication. As it was in the interest of Advance Research Project Agency (ARPA) they developed "Intergalactic Network" to link the output and results of teletype machines with the Computer Systems.

Types of Networks:

In the current modern society there are variety of Computer Networks are available based on their protocol stack. Scope of networking is still same and is of communication between other computers but the means are different from wired networks to wireless The major computer networks today are Ethernet in wired and Wifi, buletooth and Ad-hoc in wireless. There are some upcoming great technologies which are not commonly deployed are WiMax (in computers) and 3G (in cellular networks).

A Concept of Client & Server Machine:

A Server is a machine that provides the services and resources to the other machines on the network while client are those which utilize those services provided by the server.

Now a days networks have become so vast that whole city is interconnected via Metropolitan Area Network (MAN) and about the whole world is connected via Internet. As the network grows the number of servers which provide administration to the networks also increase and it is also not possible to place all the servers on a single corner of the world. Considering a server placed at another city has got a minor software problem and need to be fixed before the next working day, the administration team from other city cannot go there to fix it. So they have two options either to hire another team which would be available tomorrow or the same team can have the access (remotely) to the server so they can resolve the issue. Which one is the better option? Of course the second one.

Remote Access:

The ability to access the computer or network from a remote site. Remote access requires communications hardware, software, and actual physical links, although this can be as simple as common carrier (telephone) lines or as complex as Telnet login to another computer across the Internet. The Remote access to the server should be secure so only authorized person and legitimate users can access that computer not everyone else.

Applications[2]:

There are many applications available in market for remote administration. Some of the names are as follows:

- Remote Desktop Connection (Windows based)
- VNC (Windows Based)[3]
- WEBMIN (Windows and Linux based)[4]

All of the above applications requires proper authentication to access the system. There are some ways to secure these types of services not only authentication is required but the whole communication should be on private tunnel so attacks like SESSION HIJACKING cannot harm the servers.

In this paper I will only discuss the security flaws lies in the WEBMIN only several other applications also suffers from other various bugs.

Network Security:

A basic understanding of computer networks is requisite in order to understand the principles of network security.

Protection of networks and their services from unauthorized modification, destruction, or disclosure, and provision of assurance that the network performs its critical functions correctly and there are no harmful side-effects. Security can be seen as a subfield of security engineering, which looks at broader security issues in addition to computer security.

It's very important to understand that in security, one simply cannot say ``what's the best firewall?'' There are two extremes: absolute security and absolute access. The closest we can get to an absolutely secure machine is one unplugged from the network, power supply, locked in a safe, and thrown at the bottom of the ocean. Unfortunately, it isn't terribly useful in this state. A machine with absolute access is extremely convenient to use: it's simply there, and will do whatever you tell it, without questions, authorization, passwords, or any other mechanism. Unfortunately, this isn't terribly practicable, either: the Internet is a bad neighborhood now, and it isn't long before some bonehead will tell the computer to do something like self-destruct, after which, it isn't terribly useful to you.

There are many types of Security Threats all of them can not be explained in this paper. Some of the related threats are as follows:

- Denial of Service (DOS)
- Unauthorized Access
- Remote Command Execution
- File Disclosures

Denial of Service :

Denial of service (DoS) attack is an incident in which a user or organization is deprived of the services of a resource they would normally expect to have. Typically, the loss of service is the inability of a particular network service, such as e-mail, to be available or the temporary loss of all network connectivity and services. In the worst cases, for example, a Web site accessed by millions of people can occasionally be forced to temporarily cease operation. A denial of service attack can also destroy programming and files in a computer system. Although usually intentional and malicious, a denial of service attack can sometimes happen accidentally. A denial of service attack is a type of security breach to a computer system that does not usually result in the theft of information or other security loss. However, these attacks can cost the target person or company a great deal of time and money.

It can be either simple DOS or DDOS (Distributed Denial of services). In DDOS simply the attack is launched from different locations targeting the single network so it may get down soon or for more destruction services. It can be considered as a weapon of mass destruction in Computer Networks.

Unauthorized Access:

"Unauthorized access" is a very high-level term that can refer to a number of different sorts of attacks. The goal of these attacks is to access some resource that your machine should not provide the attacker. For example, a host might be a web server, and should provide anyone with requested web pages. However, that host should not provide command shell access without being sure that the person making such a request is someone who should get it, such as a local administrator.

Remote Command Execution:

In this type of attack the particular (FAULT / bug) in an application is exploited to execute the commands on the server without accessing it. Today many applications suffers from bugs which has a threat of arbitary remote command execution.

File Disclosure:

Similar like the above, these types of attacks are launched due to bugs in an application to disclose files on the server resulting the leakage of sensitive information like passwords, credit cards and other information.

These attacks looks simple but are very powerful and can be use to gain access to the systems or privilege escalation purposes.

In this paper the Term BUGS and FAULTS are use. In Softwares faults results in bugs (software error).

Software BUG:

In computer technology, a bug is a coding error in a computer program. A problem that causes a program to produce invalid output or to crash (lock up). The problem is either insufficient logic or erroneous logic. For example, a program can crash if there are not enough validity checks performed on the input or on the calculations themselves, and the computer attempts to divide by zero. Bad instruction logic misdirects the computer to a place in the program where an instruction does not exist, and it crashes. A program with bad logic may produce bad output without crashing, which is the reason extensive testing is required. For example, if the program is supposed to add an amount, but subtracts it instead, bad output results, although the computer keeps running.

After a product is released or during public beta testing, bugs are still apt to be discovered. When this occurs, users have to either find a way to avoid using the "buggy" code or get a patch from the originators of the code.

There are many types of bugs in software but we will discuss about cross-site scripting bugs (XSS), WEBMIN was prone to this bug.

Introduction to WEBMIN:

WEBMIN is a Remote Administration tool[2] for both LINUX/UNIX and Windows based systems. WEBMIN is a web-based interface for system administration. Using any browser that supports tables and forms (and Java for the File Manager module), you can setup user accounts, Apache, DNS, file sharing and so on. It consist of a simple web server, and a number of CGI programs which directly update system files. All the modules and web server is coded in PERL 5.

WEBMIN is unique in the UNIX world, as it provides Graphical Interface to every service for configuration and maintenance in the system. I can be easily access from any corner of the world by just having a web browser pointing towards it. It does not modify files unnecessarily or in incompatible ways. Using WEBMIN for configuration does not preclude configuration with command line tools. WEBMIN is also accessible, via text mode browsers only (nearly all of browsers are supported like Links/Lynx etc). It is an excellent tool for both types of users Expert and novice. For novice users it provide visual interface to system administration and configuration with all option available and easy to understand.

Expert users can have advantage as there is no need to remember difficult syntaxes and commands. It provides whole system configuration in very less amount of time as compare to command line tools thus providing access to all options in visual fashion.

Discussing other features are beyond the scope of this paper.

Cross-site Scripting Bug (XSS):

XSS is an abbreviation for Cross Site Scripting. This refers to a type of computer security vulnerability where malicious users can add carefully-constructed comments to web pages with the intention of fooling web browsers. While most websites have filters to determine when a post containing XSS code is made, it is near impossible to filter all the different types of attacks possible. Thus, almost any website that allows users to post comments is susceptible to Cross Site Scripting. Cross Site Scripting is generally believed to be one of the most common application layer hacking techniques.

While Cross Site Scripting exploits are often done for the naive fun of creating Java Script popups on popular web pages such as news articles, XSS attacks can also be vectors in more serious attacks such as phishing. Commonly, cookie information is stolen via XSS. This allows attackers access to users accounts on websites under attack. Additionally, attackers can change web pages to link to malicious web sites that appear to be legitimate, add advertising to web pages, change user settings, and more. Every week between 3 and 5 new Cross Site Scripting exploits are discovered, each using a different method and affecting the victim in a different way. All major websites including Yahoo[5], Paypal[6], Ebay[7], CNN[8], Microsoft[9], and even FBI[10] have been subject to successful XSS attacks. As Cross Site Scripting is not dependent upon unencrypted connections, secure websites (those that display a lock icon in the browser) are as vulnerable to XSS as any other website.

In general, cross-site scripting refers to that hacking technique that leverages vulnerabilities in the code of a web application to allow an attacker to send malicious content from an end-user and collect some type of data from the victim.

Code Insertion:

The success of this type of attack hinges upon the functionality of the client browser. In HTML, to distinguish displayable text from the interpreted markup language, some characters are treated specially. One of the most common special characters used to define elements within the markup language is the ?<? character, and is typically used to indicate the beginning of an HTML tag. These tags can either affect the formatting of the page or induce a program that the client browser executes (e.g. the <SCRIPT> tag introduces a JavaScript program).

As most web browsers have the ability to interpret scripts embedded within HTML content enabled by default, should an attacker successfully inject script content, it will likely be executed within context of the delivery (e.g. website, HTML help, etc.) by the end user. Such scripts may be written in any number of scripting languages, provided that the client host can interpret the code. Scripting tags that are most often used to embed malicious content include <SCRIPT>, <OBJECT>, <APPLET> and <EMBED>.

Consider the <FORM> tag ? by inserting the appropriate HTML tag information, an attacker could trick visitors to the site into revealing sensitive information by modifying the behavior of the existing form for instance. Other HTML tags may be inserted to alter the appearance and behavior of a page (e.g. alteration of an organizations online annual accounts or presidents statement?).

It is important to understand the HTML tags that are most commonly used to carry out code insertion tags. The following table details the most important attributes of these tags. However, it is important to note that alternative in-line scripting elements may be used and interpreted by the current generation of web browsers, such as javascript:alert('executing script')[11].

XSS in WEBMIN:

On 13th July the bug XSS cross-site scripting (code injection) bug was discovered in WEBMIN version 1.29x over various security forums. A vulnerability reported in WEBMIN and Usermin can be exploited by malicious people to disclose potentially sensitive information[12].

This vulnerability is caused due to an improper handling and improper sanitization of crafted URL. This can be exploited to read the contents of any files on the server via a specially crafted URL, without requiring a valid login.

Upon more research it was reveal that WEBMIN does not properly handle a URL with a null ("%00") character, which allows remote attackers to conduct cross-site scripting (XSS), read CGI program source code, other sensitive information like Credit Card, Paypal[6] accounts and password (shadows) files in Unix/Linux environment. This security threat lead to the system compromise under existence of best firewalls and system administration.

These files were disclose to an attacker like the other information is available to end-users over a web, thus becoming hard to detect because attacker is just acting as a user browsing for a web service.

XSS BUG:

In order to configure the services on the server user must authenticate to WEBMIN console (web based). Whenever the wrong user id and password is provided the user is returned to the login page again asking for an authentication. When we check the source of an authentication page we can clearly see that all the images and contents are fetched from a directory named "unauthenticated" and are display on page. From there we can understand that with out begin a legitimate user that is without an authentication we have a access to the particular folder named "unauthenticated" from where contents are being display.

The snapshot of HTML code is below

```
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; Charset=iso-8859-1">
<link rel='icon' href='/images/webmin_icon.png' type='image/png'>
<title></title>
</head>
<body bgcolor=#6696bc link=#000000 vlink=#000000 text=#000000 leftmargin="0"
topmargin="0" marginwidth="0" marginheight="0"  onLoad='document.forms[0].pass.value
= ""; document.forms[0].user.focus()'><table width="100%" border="0" cellspacing="0"
cellpadding="0" background="/unauthenticated/nav/bottom_shadow.jpg">
   <tr>
   <td width="100%" nowrap><img src="/unauthenticated/nav/bottom_shadow.jpg"
    width="43" height="9"></td>
   </tr>
   </table><br><hr>

<center>

<form action=/session_login.cgi method=post>
<input type=hidden name=page value='/'>
<table border width=40%>
<tr bgcolor=#7f7f7f> <td><b>Login to Webmin</b></td> </tr>
<tr bgcolor=#b7b7b7> <td align=center><table cellpadding=3>
<tr> <td colspan=2 align=center>You must enter a username and password to login to
the Webmin server on <tt></tt>.</td> </tr>

<tr> <td><b>Username</b></td>
<td><input name=user size=20 value=''></td> </tr>
<tr> <td><b>Password</b></td>
<td><input name=pass size=20 type=password></td> </tr>
<tr> <td colspan=2 align=center><input type=submit value='Login'>
<input type=reset value='Clear'><br>
<input type=checkbox name=save value=1> Remember login permanently?
</td> </tr>

</table></td></tr></table><p>
<hr>
</form></center>

<table border=0 width=100% align=center cellspacing=0 cellpadding=0><tr><td>
</td></tr></table>
<br>
</body></html>
```

When I try to access the folder I didn't get any of permission error confirming that I have a legitimate right over this folder (to view the contents residing the in the particular folder). There was a sub folder named "nav" from which the images were captured to display over the page. When I put the complete URL in the browser I can successfully see that image in my browser so that mean I can access any file with in that folder without any permission. Thus the URL looks like http://host.com/unauthenticated/nav/bottom_shadow.jpg.

By removing the file name from the URL an error occurred "FILE NOT FOUND" which means Either I don't have right for the directory browsing or the filename which in this case is NULL (BLANK) is not available in the folder.

This above activity clearly shows that the application is prone to a XSS bug, which can disclose files to the user.

CRAFTED URL:

Now if I can browse other directories then I can see any of the file of my choice. For browsing other directories I have to convert simple URL above to a crafter URL.

Every file name contains NULL character in the end. It is a property of STRING data type. These NULL characters are generally appended by applications requesting files such as in our case by browser. Like If I am requesting for a URL "http://host.com/unauthenticated/nav/bottom_shadow.jpg" then in actual the browser is requesting "http://host.com/unauthenticated/nav/bottom_shadow.jpg<NULL>" to the server. NULL character represent the end of file name.

The traditional XSS bug is use to browse back to the main directories i.e. "../" will come back from a single sub-folder. For example in that case when the URL "http://host.com/unauthenticated/nav/../" is provided to the browser it is equivalent to the URL "http://host.com/unauthenticated/".

It is actually not a bug and use to browse back for files either putting a complete file name while creating an HTML pages but most of the XSS attacks are caused due to it.

Now we can build up a crafter URL like "http://host.com/unauthenticated/../../../../../../etc/passwd". If we browse this URL we are returned back to the main page asking for an authentication, but the URL http://host.com/unauthenticated/nav/ produce an error of "FILE NOT FOUND" means that in whenever this URL is provided it is trying to access some of the file named provided after the folder name in URL so what if we give it the null string making the server understand that the filename has end and provide “../” to browse back to the directory. The URL now looks like "http://host.com/unauthenticated/nav/..%00/..%00/..%00/..%00/..%00/".

In the above URL %00 represents the null character or EOL (End of Line). and it is intended to provide a null filename so we can avoid of getting file not found error and browse back to the main directory. This crafter string is provided many time approx (26 - 50) to come back to the root file system so we can give the complete path of the needed file for example "http://host.com/unauthenticated/nav/..%00/etc/passwd". When this crafted URL is requested the WEBMIN server would return the /etc/passwd file containing the

user password information on Linux/Unix systems. Similarly we can request for the /etc/shadow file so we can crack the passwords of possible users or root. Other files can also be requested like slocate.db which is database of all files present in the system.

This could lead to a system compromise if root password is cracked and some other information leakage like if Credit card information or other sensitive data without any authentication. This vulnerability is know as ***ARBITARY FILE DISCLOSRE***.

My proof of concept to this Exploit is available on various security websites.[13-15]

Solutions:

Users:

The only solution for the users is to disable all types of scripting from their browsers. By doing this they would suffer from missing much of the functionalities offered by new websites . Alternatively, users must be selective as to the sites they trust, and the sources of URL links. Again, the disabling of scripting languages will not prevent attackers influencing the appearance of content provided by trusted sites by embedding other HTML tags in the URL link.

Unfortunately many integrated applications increase the threat of scripting code being executed on the users system, particularly through the use of embedded objects such as Flash! .swf files. To prevent these types of attacks, users must either uninstall the interpreters or ensure protection systems are capable of stopping the execution of such content. It is envisaged that popular anti-virus and personal intrusion detection systems will eventually be capable of this.

Developers:

The key to preventing applications being vulnerable to code injection and XSS type attacks is by ensuring that dynamically generated page content does not contain undesired HTML tags. There should be a function which can properly the sanitize the URL and then process for further requests. Developers can use following techniques

Limiting the Server Response:

It is be possible to limit the amount of data that will be returned to client browsers. For example If a site providing welcome greeting to the user such as "HELLO UMAIR" by providing the URL "http://host.com/welcome.php?name=UMAIR" it can be vulnerable to XSS security threat so the developer can compromise with dynamically welcoming by just hard coding it as WELCOME USER.

Limiting the Response and its Length:

Developers can also limit the response from the users by filtering tags and dangerous strings like "<>, %00 and </>" etc from the URL and thus limiting the response length so attacker can not easily use the long crafted URL like in above case.

Patch:

System Administrators should patch the application installed on the server so they can avoid these types of security threats.

FIREWALL:

A well configured firewall can also protect and Remote administration should be through VPNs other then over Internet so any legitimate users can access it and use it whenever needed. IPS (intrusion Prevention Systems) can also be use and logs should be monitored on regular basis in order to keep the track of services offered and monitoring of improper actions.

Less Privileges:

All the services on the server should be execute with the less privileges accounts so that in case of system compromise the attacker wont be able to access all information and hence cannot change the configuration. Like in the above case if the WEBMIN is running as root obviously it will return the password shadows and all conf files, but in case if its running as nobody (a user that cannot access shell and use for running services like apache and bind) the attacker would not be able to compromise the system and file disclosure is limited to very few files.

References:

[1] Wikipedia Defination [http://en.wikipedia.org/wiki/Computer_networks]

[2] Other Remote Admnistration Tools [http://www.download.com/3120-20_4-0.html?tg=dl-20&qt=Remote%20administration&tag=srch]

[3] REAL VNC HOMEPAGE [http://www.realvnc.com]

[4] WEBMIN HOMEPAGE [http://www.webmin.com/download.html]

[5] Yahoo [http://www.yahoo.com]

[6] PAYPAL [http://www.paypal.com]

[7] EBAY [http://www.ebay.com]

[8] CNN [http://www.cnnc.com]

[9] Microsoft [http://www.microsoft.com]

[10] FBI [http://fbi.com]

[11] Mentioned in Some technical paper on XSS bugs

[12] Secunia Security [http://secunia.com/advisories/20892/]

[13] Exploit available at [http://www.milw0rm.com/exploits/2017]

[14] Mirror of Exploit
[http://www.securitydot.net/xpl/exploits/vulnerabilities/articles/1164/exploit.html]

[15] Video Tutorial [http://www.milw0rm.com/video/